UVIS: Unsupervised Video Instance Segmentation

Shuaiyi Huang[1], Saksham Suri[1], Kamal Gupta[1*],
Sai Saketh Rambhatla[2*], Ser-nam Lim[3], Abhinav Shrivastava[1]
[1]University of Maryland, College Park [2]Meta [3]University of Central Florida

Abstract

Video instance segmentation requires classifying, segmenting, and tracking every object across video frames. Unlike existing approaches that rely on masks, boxes, or category labels, we propose UVIS, a novel Unsupervised Video Instance Segmentation (UVIS) framework that can perform video instance segmentation without any video annotations or dense label-based pretraining. Our key insight comes from leveraging the dense shape prior from the self-supervised vision foundation model DINO and the open-set recognition ability from the image-caption supervised vision-language model CLIP. Our UVIS framework consists of three essential steps: frame-level pseudo-label generation, transformer-based VIS model training, and query-based tracking. To improve the quality of VIS predictions in the unsupervised setup, we introduce a dual-memory design. This design includes a semantic memory bank for generating accurate pseudo-labels and a tracking memory bank for maintaining temporal consistency in object tracks. We evaluate our approach on three standard VIS benchmarks, namely YoutubeVIS-2019, YoutubeVIS-2021, and Occluded VIS. Our UVIS achieves 21.1 AP on YoutubeVIS-2019 without any video annotations or dense pretraining, demonstrating the potential of our unsupervised VIS framework.

1. Introduction

Video Instance Segmentation (VIS) [51] is the task of classifying, segmenting, and tracking individual objects within a video, with a wide range of industry applications such as in robotics, sports, autonomous driving, surveillance, AR/VR, 3D navigation [39, 59], etc. It is a challenging problem due to variations in object appearances, occlusions, and cluttered scenes over time. Reliable models for VIS require dense annotated data which is costly. To circumvent the need for costly dense annotations in videos, existing methods have utilized various strategies such as pretraining on densely-labeled image datasets like COCO and finetuning on fully-labeled [11] or unlabeled [15] videos, or reducing annotations through subsampled frames [19], boxes [24], per-frame category labels [28]. However, these methods still rely on annotations [19, 24, 28] or can only handle categories that overlap with the densely-labeled image dataset [15]. In contrast, the human perception leverages image and video-level priors to effortlessly recognize, segment, and track objects [3]. This leads us to explore whether it is possible to learn an unsupervised video instance segmentation model without any dense pretraining or video annotations, covering all categories in a dataset.

Unsupervised video instance segmentation presents several challenges when only the category label set is provided for the video dataset. The first challenge is accurately predicting object boundaries without dense labeling in videos. The second challenge is conducting object classification when only the category label set is available. To address these challenges, we draw inspiration from recent advancements in large-scale unsupervised vision models, specifically the dense shape prior in self-supervised vision model DINO [7] and the open-set recognition capability in image-caption supervised vision and language model CLIP [36]. By combining these strengths, our unsupervised VIS model can effectively segment and recognize objects within a given vocabulary set without the need for dense pretraining. To the best of our knowledge, we are the first work to explore CLIP and DINO in the field of VIS. This naturally solves the limitation of existing works that can only handle categories that overlap with densely labeled external image datasets [15].

To this end, we introduce an unsupervised video instance segmentation framework (UVIS), which is the first VIS framework, that can learn to segment all categories in videos without any dense annotation based pretraining or video annotations, as shown in Figure 1. Our unsupervised framework for video instance segmentation comprises of three essential steps. First, we generate class-agnostic instance masks for each video frame using a pre-trained self-supervised model [43] and equip the masks with semantic labels by using CLIP [37]. Second, we train a transformer-based video instance segmentation model by

*Work done while at UMD.

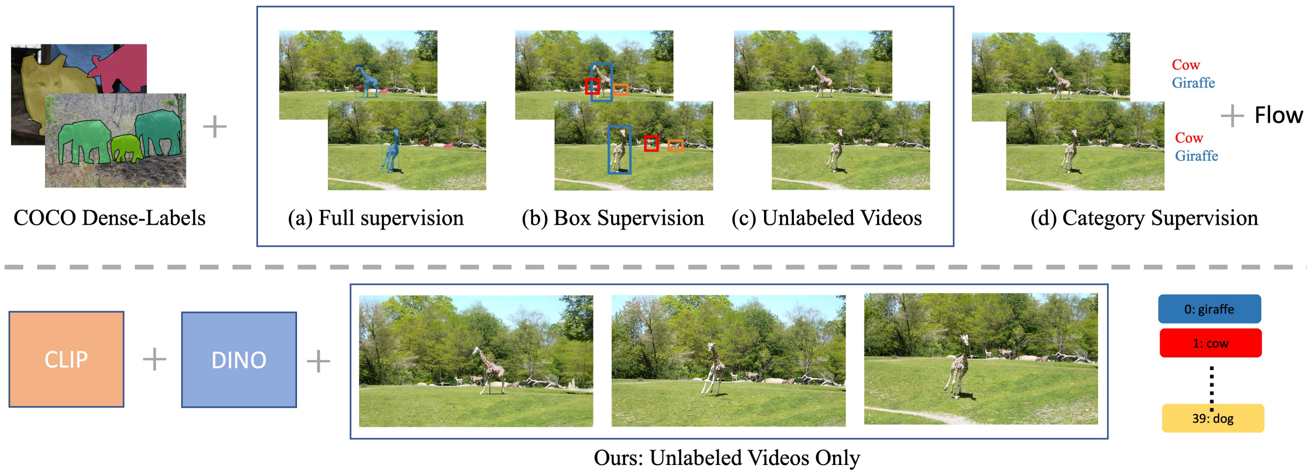


Figure 1. **Setting Overview.** Previous approaches have tried to use COCO dense annotations in addition to VIS dataset full supervision (a), box supervision (b) and no supervision (c). Additionally, previous works have also used flow information along with frame level category labels (d). Our approach UVIS works in the unsupervised setting and does not require any dense labels or per frame labels and instead utilizes foundation models.

using the per-frame pseudo-masks obtained from the first step [19]. Third, during inference, we generate dense and consistent mask tubes by linking per-frame predictions using bipartite matching of query features.

To generate high-quality VIS predictions in the unsupervised setup, we further propose a novel dual-memory design on top of the above proposed UVIS framework for semantically accurate and temporally-consistent predictions. Specifically, to obtain semantically accurate pseudo-labels, we construct a class-specific prototype memory bank during pseudo-label generation. The prototypes serve as representative references, enhancing generalization and handling noisy false positives. In addition, to address the inherent limitations of the online inference pipeline in VIS that utilized only short-term information for tracking, we propose a simple but effective tracking memory that models long-term temporal information. To summarize, our main contributions are as follows:

- We introduce the first VIS framework that eliminates the need for any video annotations or dense label based pretraining, thereby significantly reducing annotation costs. Our framework covers all categories in the dataset, offering a comprehensive solution.
- We propose a novel dual-memory design on top of our unsupervised VIS framework. This design includes a prototype memory filtering component, which enhances the quality of pseudo-labels, and a tracking memory bank, which captures long-term temporal information for accurate tracking.
- We conduct comprehensive experiments in three standard video instance segmentation datasets including YoutubeVIS-2019 [51], YoutubeVIS-2021 [51], and Occluded VIS [35], demonstrating the potential of our unsupervised VIS framework.

2. Related Work

Video object segmentation. Video object segmentation (VOS) [5, 6, 33, 34, 50] is a dense binary classification problem of separating salient foreground objects from the background in videos. The most popular task in VOS is the so-called *Semi-supervised* VOS, where the goal is to segment objects in target frames given ground truth masks in the first frame. To prevent the annotation costs of exhaustively labeling each frame, several weakly and unsupervised VOS method have been proposed. [47] uses video level tags while [44] uses point supervision as weak labels to train a weakly supervised VOS system. [12, 26, 42, 55] propose unsupervised VOS to completely eliminate the need for supervision and is a much harder problem than fully and weakly supervised VOS. In this work, we tackle a much harder problem of unsupervised Video Instance segmentation that not only does background separation but additionally performs instance segmentation that requires classification and tracking without any human supervision.

Supervised video instance segmentation. Video Instance Segmentation (VIS), initially proposed by Yang et al.[51], is an extension of image instance segmentation to videos, where the goal is to classify, segment and track objects across video frames. Early approaches[14, 27, 31, 52, 57] segment and classify objects in each frame independently, and then associated the objects across frames using heuristics such as mask or box IoU. Recently, transformer-based approaches for VIS have gained significant attention [11, 22, 45, 53]. These approaches train VIS models in a video-based manner where they feed a clip as input and generate spatio-temporal mask predictions in one shot. A more recent development is the introduction of MinVIS [19]. This pioneering work demonstrates that a transformer-based VIS model trained solely on images can achieve competitive performance without video-based training or specialized video-based architecture design. They observe that instance tracking naturally emerges in query-based image instance segmentation models with proper architectural constraints. We build our work on top of MinVIS [19] due to its excellent performance in VIS using image-based training. Note that such a pipeline differs fundamentally from existing approaches such as IDOL [49], which rely on post-processing steps like non-maximum suppression (NMS) during inference for tracking. However, MinVIS does not consider long-term temporal information during tracking, we address this inherent limitations by incorporating crucial temporal information in image-based VIS.

Weakly/Semi-supervised video segmentation. Reducing the annotation requirements in VIS has become a focus of recent research efforts [15, 19, 28]. Liu *et al.*[28] utilize per-frame category annotations and correspondences [1, 16, 20, 21] in videos, but exhibiting limited competitiveness compared to supervised approaches. Fu *et al.*[15] utilize instance segmentation annotations from the COCO dataset to learn VIS without video annotations, but are only applicable to overlapping categories between video and image datasets. Huang *el al.* [19] utilized annotations in subsampled frames but still rely on dense annotations. In contrast, our UVIS method handle all categories for a given vocabulary without any per-frame category/box/mask label or COCO pretraining. To the best of our knowledge, this is the first unsupervised VIS framework that achieves impressive results without any human annotations.

VL models based segmentation. Recently, foundational models trained on large amounts of uni-modal or multimodal data using weak or self-supervision have gained significant attention [4]. CLIP [36], a vision-language model using image-text pairs as supervision, has been particularly popular. CLIP has been extended to perform per-pixel detection and segmentation tasks in images[29, 30, 56, 58]. However, the effectiveness of CLIP for videos and instance segmentation tasks has not been thoroughly studied. In this work, we explore the use of CLIP for unsupervised VIS, which has not been adequately explored. DINO [7], a unimodal foundational model trained on unlabeled images using self-supervised learning, demonstrates impressive segmentation capabilities. However, it cannot handle complex tasks like instance segmentation due to the lack of labeled information. Our approach combines the segmentation capabilities of self-supervised models with the zero-shot capabilities of CLIP to perform instance segmentation in videos. While NamedMask [40] is a related approach that performs semantic segmentation on images, our approach specifically focuses on VIS, offering a more comprehensive solution.

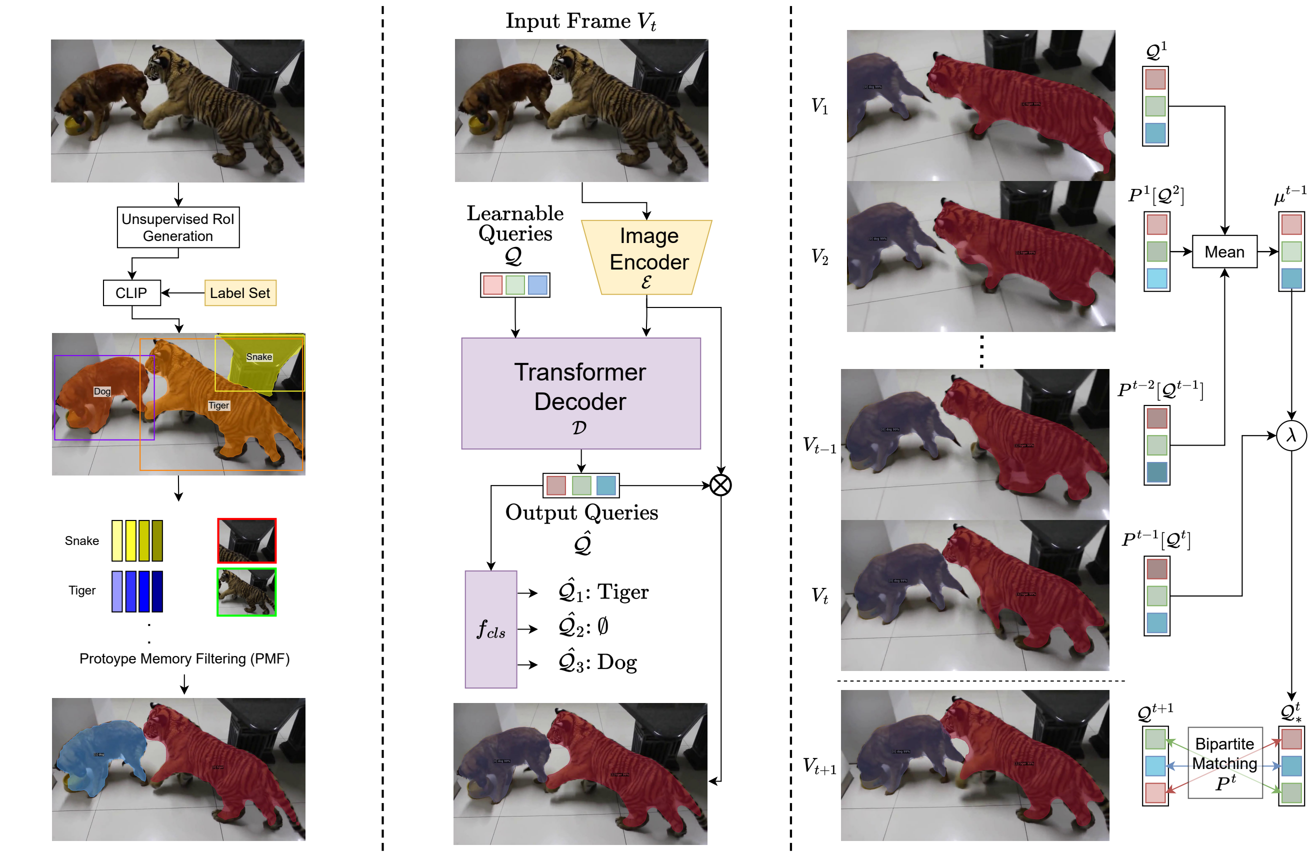
3. Method

Our objective is to learn a video instance segmentation model without groundtruth mask, box, or point annotations. The problem is challenging since we need to maintain temporal consistency while the objects may undergo appearance changes, occlusions, or partial visibility, making it difficult to track and segment them accurately over time. We build upon the recent advances in large-scale models pretrained with Internet-scale data without any dense labels, also often called 'foundation' models. Many of these models are image and text-based and do not extend trivially to videos. Hence, in this section, we propose the framework to utilize them for the video segmentation task. Our framework consists of three steps as shown in Figure 2: (1) we start by generating pseudo-masks (Section 3.1) per video frame and build a prototype memory bank for different classes in the training data. Our proposed prototype memory encodes per-class semantic information and is used to filter the false positives improving the quality of the pseudolabels, as shown in Figure 2a; (2) Secondly, we train a transformer-based video instance segmentation model (Section 3.2) by using the per-frame pseudo-masks generated from the first step as shown in Figure 2b; (3) Finally, during inference, we perform bipartite matching between instances of consecutive frames and propose a tracking memory (Section. 3.3) to build dense and consistent mask tubes across the video as shown in Figure 2c.

Formally, we are given a collection of N videos $\mathcal{V} = \{V_n\}_{n=1}^N$ (with no pixel, box, or instance-level annotations), and a set of categories that we want to segment in these videos as $\mathcal{C} = \{l_c\}_{c=1}^C$, where $C = |\mathcal{C}|$ is the number of categories, and l is the text label. Note that we assume that no per-video label information is provided, *i.e.*, the set of videos are not tagged with labels. Furthermore, a video may have zero or more of object instances corresponding to each label. Each video $V \in \mathcal{V}$ can also have a variable number of frames and we denote the t^{th} frame for this video as V_t.

3.1. Generating pseudo-labels for instance masks

Class agnostic mask generation. In the first step of our approach, we leverage self-supervised image models to generate pseudo-labels for video frames. Self-supervised learning (SSL) models such as [7–10, 18, 54], are typically trained using unlabeled ImageNet [13] train set, and have an innate discriminative and localization abilities. Several methods [2, 38, 41, 46] have been proposed to extract object masks from images using features from the SSL models. These methods typically work by performing a graph partitioning over features corresponding to various images patches and iteratively refining these partitions. We adopt



(a) Pseudo-label generation and filtering. (b) Model training with learnable queries. (c) Learned query based tracking and tracking memory.

Figure 2. **We present our approach UVIS.** On the left we show our pseudo-label generation pipeline which involves generating masks and instance labels using CutLER [43] and CLIP [37] followed by Prototype Memory Filtering (PMF). In the center we show our model training which uses and image encoder and a transformer decoder to learn queries to predict per-frame predictions. On the right we show our proposed tracking memory approach which utilizes per frame queries and a memory based update rule to perform matching between frames to track instances and generate temporally consistent predictions.

CutLER [43]'s self-training strategy to generate mask and box predictions for each image using the pre-trained SSL model DINO [7]. See the supplementary material for the details of the approach. Given a video frame V_t, CutLER predicts a set of boxes $\{b_t^i\}$, masks $\{M_t^i\}$ and their corresponding objectness scores $\{o_t^i\}$ where i corresponds to the i^{th} object instance in the frame.

CLIP based Text-Instance Matching. In order to associate each mask M_t^i to the corresponding label of interest, we utilize CLIP [37], a vision-language model trained with aligned text and image data. CLIP consists of a vision module $f_{\text{vision}}^{\text{CLIP}}$ and a text module $f_{\text{text}}^{\text{CLIP}}$ to compute image and text embeddings respectively. Given an image I, the model assigns a class (from a list of classes) to it by computing the cosine similarity between the image embedding and the embeddings of a list of text prompts, and selecting the closest prompt. The list of text prompts is generated from labels by simple strings such as "a photo of < class >". In practice, a larger set of text prompts per class is used, we provide details in the supplemental. In our case, we generate the CLIP embeddings and the scores for each of the instance regions $\{M_t^i\}$ by using the corresponding box $(\{b_t^i\})$ to get the instance crop $(\{b_t^{i\oplus}\})$. We assign initial class labels to each of the instances using the CLIP model as following.

$$\text{class}(i) = \arg\max_{l\in\mathcal{C}} \left(f_{\text{vision}}^{\text{CLIP}}(b_t^{i\oplus}) \cdot f_{\text{text}}^{\text{CLIP}}(\text{a photo of } \langle l \rangle\ ) \right) \tag{1}$$

where $b_t^{i\oplus}$ is the cropped instance region for frame V_t and $\text{class}(i)$ is the initial class assigned to the i^{th} instance by the CLIP model. We also denote the CLIP class score for this instance as $u_t^i = f_{\text{vision}}^{\text{CLIP}}(b_t^{i\oplus}) \cdot f_{\text{text}}^{\text{CLIP}}(\text{a photo of} < \text{class}(i) >)$.

Prototype Memory Filtering (PMF). These initial classes or pseudo-labels are often noisy and contain a lot of false positives. To address this, we create class specific prototypes using the initial class labels as following. For each class label $l \in \mathcal{C}$ we accumulate all the instance features given by $f_{\text{vision}}^{\text{CLIP}}(b_t^{i\oplus})$. We apply K-Means clustering on the features and compute k^l centroids. We set k^l to be proportional to the number of instances in class l. We denote these clusters as the prototype clusters for class l. We can now compute an out-of-distribution score for each instance i such that $\text{class}(i) = l$. To do this we compute the cosine similarity between the prototype clusters of class l and CLIP features for an instance $f^{\text{CLIP}}_{\text{vision}}(b^{i\oplus}_t)$ which has the same predicted class. We discard all instances for which the maximum similarity with any prototype is less than a threshold τ. Together with objectness score and CLIP score, τ determines the final instances we retain for each prototype in our prototype memory. These protoypes' embeddings collectively reflect various pose, appearances and instances of the objects within the same category.

3.2. Training the segmentation model

Using the pseudo-mask labels from the last step, we next train an instance segmentation model which comprises of a convolutional image encoder $\mathcal{E}$ and a transformer decoder $\mathcal{D}$. Our setup is similar to the one used in MinVIS [19] which uses a supervised setting, as compared to our unsupervised case. We provide more details below and some additional details in the supplemental.

Given a frame V_t and corresponding pseudo-labels l^i_t and M^i_t, the model uses the fully convolutional image encoder to extract multi-scale features $\mathbf{F}_t = \mathcal{E}(V_t)$. Input to the decoder are $q \in \mathcal{Q}$ learnable query embeddings along with the the encoder features ($\mathbf{F}_t$) with $|\mathcal{Q}| = N_q$. The transformer decoder then outputs transformed queries such that $\hat{q} = \mathcal{D}(\mathbf{F}_t, q)$. Each query $\hat{q} \in \hat{\mathcal{Q}}$ is passed to a classification head f_{cls} to obtain classification scores $s = f_{\text{cls}}(\hat{q}), s \in \mathbb{R}^{1\times C}$ where $C = |\mathcal{C}|$.

Along with classification score per query, we also obtain a segmentation masks $M \in \mathbb{R}^{N_q\times H\times W}$ for the query by convolving transformed query embedding $\hat{q}$ with last layer's features in $\mathbf{F}_t$ where H and W are the height and width of the image. In other words, $M = \sigma(\hat{q} * \mathbf{F}^{-1}_t)$ where $\sigma(.)$ is the sigmoid function, $*$ is the convolution operation and -1 represents the last layer's features. During training, the classification head outputs (s) and segmentation head outputs (M) are used to perform bipartite matching between predictions and pseudo-labels that minimize the classification and segmentation losses. Once assigned, the losses are recomputed based on the matching to obtain the total loss $\mathcal{L}_{\text{vis}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{seg}}$ where $\mathcal{L}_{\text{cls}}$, the classification loss, is computed using cross entropy. $\mathcal{L}_{\text{seg}}$, the segmentation loss, is computed using binary cross entropy and dice loss [32].

Once the model is trained, the per-frame learned queries $\hat{q}$ are used to perform tracking and generate temporally consistent instance masks for each instance along with the predicted class labels.

3.3. Tracking using learned Queries and Memory

Query based Tracking. Our transformer-based model learns queries which help us identify and label each instance region. During training, we utilize per frame predictions to propogate the loss and do not use any temporal cues. But while performing inference, we require temporally and spatially consistent predictions. To extend the per-frame predictions during inference, we utilize the similarity between query embeddings between adjacent frames. Using cosine similarity based Hungarian matching between queries $\mathcal{Q}^t$ and $\mathcal{Q}^{t+1}$ of frames V_t and the next frame V_{t+1} we obtain the permutation operator (P^t) of queries ($\mathcal{Q}^t$) which assigns them to $\mathcal{Q}^{t+1}$. Utilizing a large number of queries, automatically tackles occlusion, and birth and death of tracklets by detecting null/empty masks, with enough queries remaining to track the foreground objects of interest. The final class prediction in this case for each tracklet is computed using the averaged logits across time.

Tracking with Memory Bank. While using cosine similarity to propagate instances across frames using Hungarian matching can give us tracklets and corresponding labels, we notice that for the unsupervised case, these are not very accurate. This can be attributed to the noise in pseudo-labels involved in training and the lack of encoding of variations in appearance of the same instance. To further improve the tracking we utilize a tracking memory module. This module performs averaging of the query vectors based on the matching between two frames to use a weighted query feature from all previous frames. This adds temporal memory to each query feature which is able to encode the instance appearance over a time window instead of just focusing on the previous frame. Specifically, given two frames, V_t and V_{t+1} of video V_n, having queries $\mathcal{Q}^t$ and $\mathcal{Q}^{t+1}$, instead of matching Q^{t+1} with $\mathcal{Q}^t$ we define the averaged $\mathcal{Q}^t_*$ instead as follows:

$$\mathcal{Q}^t_* = \lambda * P^{t-1}[\mathcal{Q}^t] + (1-\lambda) * \mu^{t-1} \tag{2}$$

where $\mu^t \in \mathbb{R}^{N_q\times d}$ is the average memory. Here N_q is the number of per frame queries and d is the dimension of each query vector. We define μ^{t-1} as follows:

$$\mu^{t-1} = \frac{1}{t-1}\left(\mathcal{Q}^1 + P^1[\mathcal{Q}^2] + P^2[\mathcal{Q}^3] + ... + P^{t-2}[\mathcal{Q}^{t-1}]\right) \tag{3}$$

4. Experiments

We evaluate our method on three VIS benchmarks: YouTube-VIS 2019 [51] (YTVIS-2019), YouTube-VIS 2021 [51] (YTVIS-2021), and Occluded VIS [35] (OVIS). We describe our experimental setup in Section 4.1, compare UVIS with state-of-the-art fully-supervised approaches in Section 4.2, and provide an ablation study in Section 4.3. For more details, please refer to the supplement.

Table 1. **Mask ($\mathcal{M}$) / Box ($\mathcal{B}$) / category ($\mathcal{C}$) vs. our unsupervised setting on validation set of YouTube-VIS 2019 [51], YouTube-VIS 2021 [51], and OVIS [35].** * indicates training in videos without COCO pretrained model weights as initialization using authors' official code. "I-Sup." and "V-sup." indicate the supervision used in the image dataset and the video dataset, respectively. All results below are based on R50 backbone. Our UVIS achieves decent results in all three datasets without any videos annotations or dense supervision from images.

Method	Video-Dataset	COCO	I-Sup.	V-Sup.	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
IDOL [49]	YTVIS-2019	✓	$\mathcal{M}$	$\mathcal{M}$	49.5	74.0	52.9	47.7	58.7
MinVIS [19]	YTVIS-2019	✓	$\mathcal{M}$	$\mathcal{M}$	47.4	69.0	52.1	45.7	55.7
MaskFreeVIS [24]	YTVIS-2019	✓	$\mathcal{B}$	$\mathcal{B}$	42.5	66.8	45.7	41.2	51.2
MinVIS [19]	YTVIS-2019	✗	-	$\mathcal{M}$	30.3	51.3	30.1	34.7	38.1
WISE [25]	YTVIS-2019	✗	-	$\mathcal{C}$	6.3	17.5	3.5	7.1	7.8
IRN [1]	YTVIS-2019	✗	-	$\mathcal{C}$	7.3	18.0	3.0	9.0	10.7
WeakVIS [28]	YTVIS-2019	✗	-	$\mathcal{C}$	10.5	27.2	6.2	12.3	13.6
DeepSort [48]	YTVIS-2019	✗	-	-	12.5	27.1	10.8	15.3	18.1
UVIS	YTVIS-2019	✗	-	-	21.4	42.3	19.4	22.5	28.2
IDOL [49]	YTVIS-2021	✓	$\mathcal{M}$	$\mathcal{M}$	43.9	68.0	49.6	38.0	50.9
MinVIS [19]	YTVIS-2021	✓	$\mathcal{M}$	$\mathcal{M}$	44.2	66.0	48.1	39.2	51.7
MaskFreeVIS [24]	YTVIS-2021	✓	$\mathcal{B}$	$\mathcal{B}$	36.2	60.8	39.2	34.6	45.6
MinVIS [19]	YTVIS-2021	✗	-	$\mathcal{M}$	32.1	54.0	33.2	30.9	39.1
DeepSort [48]	YTVIS-2021	✗	-	-	10.3	23.0	9.4	11.9	15.5
UVIS	YTVIS-2021	✗	-	-	17.5	35.6	16.3	19.7	26.3
IDOL [49]	OVIS	✓	$\mathcal{M}$	$\mathcal{M}$	30.2	51.3	30.0	15.0	37.5
MinVIS [19]	OVIS	✓	$\mathcal{M}$	$\mathcal{M}$	25.0	45.5	24.0	13.9	29.7
MaskFreeVIS [24]	OVIS	✓	$\mathcal{M}$	$\mathcal{B}$	15.7	35.1	13.1	10.1	20.4
MinVIS [19]	OVIS	✗	-	$\mathcal{M}$	15.0	33.9	12.8	9.8	19.3
DeepSort [48]	OVIS	✗	-	-	1.6	4.0	1.4	1.9	3.9
UVIS	OVIS	✗	-	-	3.5	11.1	2.1	3.6	7.0

4.1. Experimental Setup

Datasets. YouTube-VIS 2019 dataset [51] (YTVIS-2019) is widely used for video instance segmentation task. It comprises 2,883 labeled videos, 131,000 instance masks, and covers 40 different classes. An improved version called **YouTube-VIS 2021** (YTVIS-2021) was also introduced [51], featuring 8,171 unique video instances and 232,000 instance masks. **OVIS** is another challenging dataset, offering heavy occlusion, longer sequences and more number of objects. OVIS consists of 296,000 instance masks and contains an average of 5.8 instances per video across 25 classes.

Experimental Setup. We highlight our experimental setup here. For pseudo-label generation, we utilize CutLER [43] pretrained on ImageNet for class-agnostic masks generation using their Cascade-Mask-RCNN-based pretrained checkpoint. For labeling the proposed regions we use CLIP ViT-bigG-14 [36] from OpenCLIP [23]. We apply a threshold of 0.7 to both objectness score from CutLER (o_t^i) and class score (u_t^i) from CLIP, and set $\tau = 0.7$.

For VIS architecture and optimization, we follow MinVIS [19]'s model architecture, training hyperparameters, and losses. Specifically, for the MinVIS architecture, we utilizes a ResNet-50 [17] (R50) image encoder and a transformer decoder and sets $N_q = 100$. However, we made three major modifications to it. Firstly, instead of relying on ground truth masks, we employed pseudo masks generated by our method (cf. Section 3.1). Secondly, instead of pretraining on COCO with dense labels, we use ImageNet classification for backbone initialization and train the transformer from scratch. Therefore, we increased the number of interactions to 320k as our setup requires more time to converge. Lastly,we incorporated our proposed tracking memory during inference and set λ=0.5.

Baselines. To our best knowledge, ours is one of the first works to introduce the task of unsupervised VIS and does not have any direct baselines to compare with. We propose a new baseline for comparison by utilizing DeepSort [48]. DeepSort [48] does not require any training and produces tracks given per-frame detections and deep features. We feed our per-frame pseudo-labels in the validation split and the associated CLIP CLS token features of each instance crop into DeepSort to generate satio-temporal masks for evaluation.

Metrics. For evaluation, we utilize the metrics of AP (Average Precision) and AR (Average Recall), and evaluate the performance on the validation split in line with the previous work [19, 28, 49].

4.2. Quantitative Comparison

We compare our UVIS with recent full-supervised methods including IDOL [49], MinVIS [19] as shown in Table 1. We also compare our method with recent box-supervised method MaskFreeVIS [24] and category-label supervised

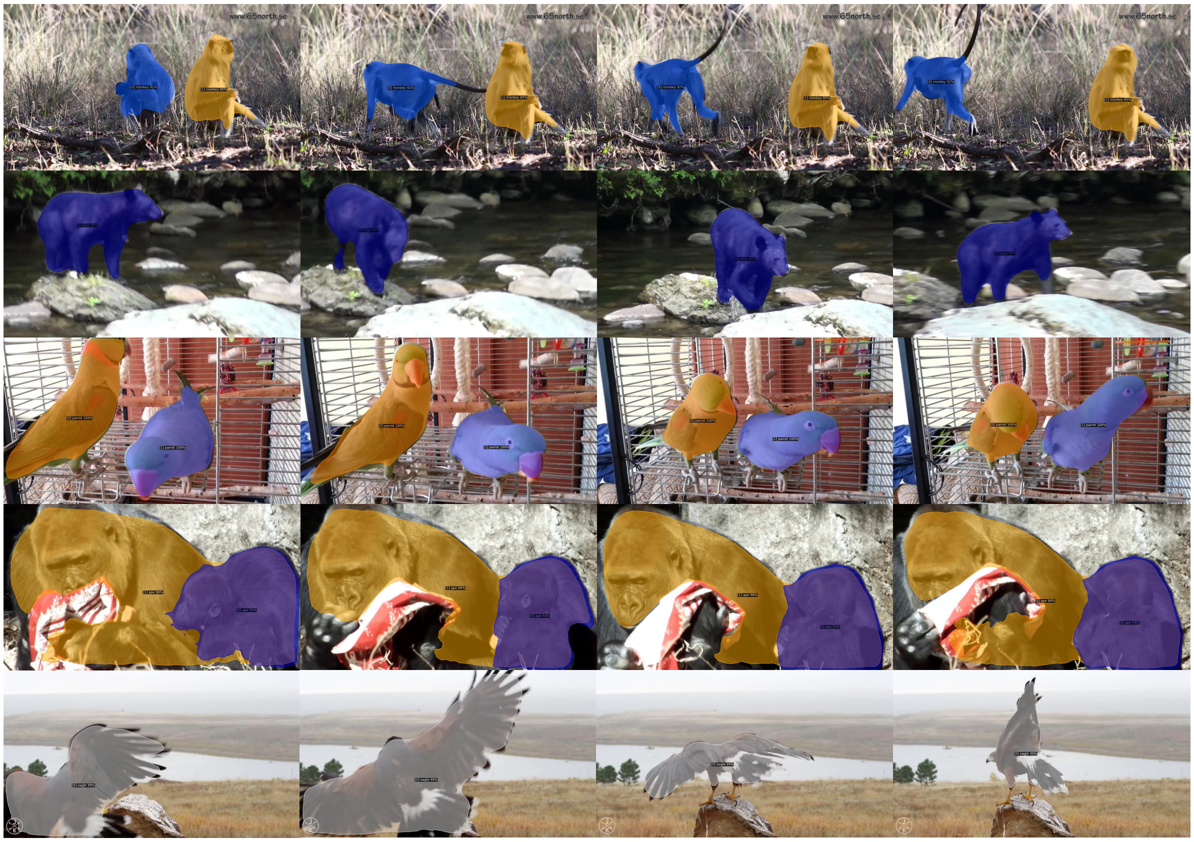


Figure 3. **Visualizations on YoutubeVIS-2019 [51] with our UVIS**. Each row shows temporal instance mask and class predictions. Our method is able to work for examples containing multiple instances of the same class (rows 1, 3, 4) and also when there are instances from different classes (row 5). UVIS shows promising results when instances of the same class might overlap (row 4).

method WeakVIS [28]. Note that MinVIS [19] (w/o COCO pretraining) serves as the fully-supervised counterpart of UVIS.

YouTube-VIS 2019. As shown in Table 1, we achieved an impressive AP of 21.4 without relying on any annotations or COCO pretraining. This result outperforms the previous weakly-supervised method [28], which utilized per-frame category labels on videos and external flow networks, by a significant margin of 10.9 AP. Our self-constructed baseline of DeepSort [48] also performs better than WeakVIS [28] by 2 AP showing it as an effective approach for comparison. We also show qualitative results of our approach in Figure 3.

YouTube-VIS 2021. Our UVIS achieves 17.5 AP on this more challenging dataset. It also beats the DeepSort baseline by 7.2 AP showing the effectiveness of the proposed prototype memory filtering (PMF), training and memory based tracking. These compelling findings highlight the potential of our unsupervised video instance segmentation framework and its ability to deliver competitive results.

Occluded-VIS 2021. In the most challenging setting with the Occluded-VIS 2021 dataset, we achieve a modest result of 3.5 AP despite heavy occlusions and extremely long sequences. This is again a 1.9 AP improvement over the DeepSort baseline.

Table 2. **Ablation of different components of our pipeline on YTVIS-2019 [51] val set.**

Model ID	CLIP	Video Train	Label Denoise			Tracking Memory	AP	Δ
			Mask Score	CLIP Score	PMF			
MinVIS [19] (Upperbound)	-	-	-	-	-	-	30.3	
A1	✓	-	-	-	-	-	12.5	-
A2	✓	✓	-	-	-	-	16.6	4.1
A3	✓	✓	✓	-	-	-	18.4	5.9
A4	✓	✓	✓	✓	-	-	19.8	7.3
A5	✓	✓	✓	✓	✓	-	20.7	8.2
A6	✓	✓	✓	✓	✓	✓	**21.4**	8.9

4.3. Ablation Study

We perform ablation on 1) the effects of each model component; 2) the prototype memory filtering design choice; and 3) the tracking memory component generalizability. All results are based on R50 backbone and conducted under the same configuration for a fair comparison.

Effects of model components. We conducted an ablation study to assess the impact of each component of our model, as presented in Table 2. The Baseline-DeepSort achieves a validation split performance of 12.5 AP using CLIP features for tracking, without any video training. When training a VIS model with pseudo-labels without any filtering, the per-

Table 3. **Ablation of Prototype Memory Filtering (PMF) on YouTube-VIS 2019 [51] val**. Prototype Memory Filtering improves AP by 0.9.

τ	0.0	0.5	0.7	0.9
AP	19.8	20.1	**20.7**	19.9

Table 4. **Ablation of tracking memory on YouTube-VIS 2019 [51] val**. Our proposed tracking memory can be generalized to different datasets and different supervision settings.

Model	Sup.	Dataset	AP	AP (+Tracking Memory)
C0	-	YTVIS-2019	20.7	21.4 (+0.4)
C1	-	Occluded VIS	3.1	3.5 (+0.4)
C2	$\mathcal{I}_\mathcal{M} + \mathcal{V}_\mathcal{M}$	YTVIS-2019	47.3	50.7 (+3.4)
C3	$\mathcal{I}_\mathcal{M} + \mathcal{V}_\mathcal{M}$	Occluded VIS	26.7	27.2 (+0.5)

Figure 4. **Visualizations of failure cases on YoutubeVIS-2019 [51]**. On the left we show CLIP labeling failures where the CLIP model incorrectly classifies to the wrong class. In the center we show prediction inconsistencies where multiple instances are predicted as one. On the right we show temporal inconsistencies in predicted masks.

formance increases to 16.6 AP. By incorporating mask score and CLIP score for filtering, we observe improvements to 18.4 and 19.8 AP, respectively. Our Prototype Memory Filtering (PMF) component further enhances the performance to 20.7 AP, highlighting the importance of employing prototype memory banks for filtering out noisy labels. Finally, with the addition of our Tracking Memory component, the model achieves a 0.7 AP boost, resulting in a final performance of 21.4 AP without any supervision. This performance is only 8.9 AP lower than the upper bound achieved with full mask supervision in videos.

Prototype memory filtering ablation. We analyze the prototype memory filtering, shown in Table 3, by adjusting the threshold for keeping proposals. We observe that a lower threshold is relatively safer and yields improvements (+0.3 AP) compared to not using any prototype memory filtering. As we increase the threshold (τ), the performance further improves (+0.9 AP) due to the removal of noisy labels facilitated by our prototype memory. However, we noticed that an excessively large threshold of 0.9 does not perform as well. This could be attributed to the fact that a higher threshold leads to the significant removal of true positives.

Tracking memory ablation. We ablate our tracking memory module in Table 4. In the unsupervised setup, incorporating the tracking memory resulted in a consistent 0.4 AP boost on both the YTVIS-2019 and Occluded VIS datasets. We observe similar consistent improvement in the supervised setup too, where we use our tracking module over the official fully-supervised MinVIS checkpoint and produce a boost of 3.4 AP on YouTube-VIS 2019 and 0.5 AP on Occluded VIS. This result highlights the importance of temporal information compared to MinVIS, which only utilizes information from consecutive frames for tracking. These experimental results confirm the generalization ability and effectiveness of our tracking memory component, both in unsupervised and supervised settings.

Failure cases. In Figure 4 we highlight some failure examples. We show examples where the CLIP model assigns incorrect class to the region (left). We also show multi-instance failures where the trained model assigns an instance mask covering multiple instances of the same category (center). This usually arises when the two objects occlude each other. Finally, we show temporal inconsistency failures where the model predicts masks that are not temporally consistent and end up not masking the object perfectly.

5. Conclusion

We introduced UVIS, the first unsupervised video instance segmentation approach that eliminates the need for video annotations or dense pretraining, to the best of our knowledge. UVIS consists of three essential steps and incorporates our proposed dual-memory module to improve mask predictions. First, we generate class-agnostic instance masks for each video frame using CutLER and associate them with semantic labels using CLIP. We then employ a class-specific prototype memory bank to filter out noisy labels. Second, we train a transformer-based VIS model using image-based training and pseudo-labels obtained from the previous step. Third, during inference, we connect per-frame predictions to form mask tubes using bipartite matching of query embeddings. We enhance the tracking performance by updating query embeddings using our tracking memory bank, which captures long-term temporal information. We evaluate our approach on three standard benchmarks, namely YTVIS 2019, YTVIS 2021, and OVIS. Our work demonstrates the potential of utilizing foundation models for unsupervised VIS, contributing to the advancement of scalable video applications.

Acknowledgements. This work was partially funded by NSF CAREER Award (2238769) to AS.

References

[1] 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*, pages 2209–2218, 2019. 3, 6

[2] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022. 3

[3] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2): 115, 1987. 1

[4] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, 2021. 3

[5] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv preprint arXiv:1803.00557*, 2018. 2

[6] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv preprint arXiv:1905.00737*, 2019. 2

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 1, 3, 4

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. 3

[11] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021. 1, 2

[12] Suhwan Cho, Minhyeok Lee, Seunghoon Lee, Chaewon Park, Donghyeong Kim, and Sangyoun Lee. Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5140–5149, 2023. 2

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[14] Yang Fu, Linjie Yang, Ding Liu, Thomas S Huang, and Humphrey Shi. Compfeat: Comprehensive feature aggregation for video instance segmentation. *arXiv preprint arXiv:2012.03400*, 2020. 2

[15] Yang Fu, Sifei Liu, Umar Iqbal, Shalini De Mello, Humphrey Shi, and Jan Kautz. Learning to track instances without video annotations. In *CVPR*, 2021. 1, 3

[16] Bo He, Xitong Yang, Hanyu Wang, Zuxuan Wu, Hao Chen, Shuaiyi Huang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Towards scalable neural representation for diverse videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6132–6142, 2023. 3

[17] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 6

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 3

[19] De-An Huang, Zhiding Yu, and Anima Anandkumar. Minvis: A minimal video instance segmentation framework without video-based training. *arXiv preprint arXiv:2208.02245*, 2022. 1, 2, 3, 5, 6, 7

[20] Shuaiyi Huang, Qiuyue Wang, Songyang Zhang, Shipeng Yan, and Xuming He. Dynamic context correspondence network for semantic alignment. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 2010–2019, 2019. 3

[21] Shuaiyi Huang, Luyu Yang, Bo He, Songyang Zhang, Xuming He, and Abhinav Shrivastava. Learning semantic correspondence with sparse annotations. In *European Conference on Computer Vision*, pages 267–284. Springer, 2022. 3

[22] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. Video instance segmentation using inter-frame communication transformers. *NeurIPS*, 2021. 2

[23] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 6

[24] Lei Ke, Martin Danelljan, Henghui Ding, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask-free video instance segmentation. *arXiv preprint arXiv:2303.15904*, 2023. 1, 6

[25] Issam H Laradji, David Vazquez, and Mark Schmidt. Where are the masks: Instance segmentation with image-level supervision. *arXiv preprint arXiv:1907.01430*, 2019. 6

[26] Minhyeok Lee, Suhwan Cho, Seung-Hyun Lee, Chaewon Park, and Sangyoun Lee. Unsupervised video object segmentation via prototype memory network. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5913–5923, 2022. 2

[27] Minghan Li, Shuai Li, Lida Li, and Lei Zhang. Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11215–11224, 2021. 2

[28] Qing Liu, Vignesh Ramanathan, Dhruv Mahajan, Alan Yuille, and Zhenheng Yang. Weakly supervised instance segmentation for videos with temporal mask consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13968–13978, 2021. 1, 3, 6, 7

[29] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096, 2022. 3

[30] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. *ArXiv*, abs/2211.14813, 2022. 3

[31] Kira Maag, Matthias Rottmann, Serin Varghese, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Improving video instance segmentation by light-weight temporal uncertainty estimates. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 2

[32] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 5

[33] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 2

[34] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 2

[35] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *arXiv preprint arXiv:2102.01558*, 2021. 2, 5, 6

[36] 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 *International Conference on Machine Learning*, 2021. 1, 3, 6

[37] 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. 1, 4

[38] Sai Saketh Rambhatla, Ishan Misra, Rama Chellappa, and Abhinav Shrivastava. Most: Multiple object localization with self-supervised transformers for object discovery, 2023. 3

[39] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Ving: Learning open-world navigation with visual goals. *arXiv preprint arXiv:2012.09812*, 2020. 1

[40] Gyungin Shin, Weidi Xie, and Samuel Albanie. Namedmask: Distilling segmenters from complementary foundation models. In *CVPRW*, 2023. 3

[41] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. 2021. 3

[42] Ren Sucheng, Liu Wenxi, Liu Yongtuo, Chen Haoxin, Han Guoqiang, and He Shengfeng. Reciprocal transformations for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[43] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. *arXiv preprint arXiv:2301.11320*, 2023. 1, 4, 6

[44] Yufei Wang, Yongjiang Hu, Alan Wee-Chung Liew, and Junhu Wang. Weakly supervised video object segmentation. In *TENCON 2018 - 2018 IEEE Region 10 Conference*, pages 0315–0320, 2018. 2

[45] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021. 2

[46] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Conference on Computer Vision and Pattern Recognition*, 2022. 3

[47] Lili Wei, Congyan Lang, Liqian Liang, Songhe Feng, Tao Wang, and Shidi Chen. Weakly supervised video object segmentation via dual-attention cross-branch fusion. *ACM Trans. Intell. Syst. Technol.*, 13(3), 2022. 2

[48] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 6, 7

[49] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In defense of online models for video instance segmentation. In *ECCV*, 2022. 3, 6

[50] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 2

[51] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019. 1, 2, 5, 6, 7, 8

[52] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learning for fast online video instance segmentation. In *ICCV*, 2021. 2

[53] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Ying Shan, Bin Feng, and Wenyu Liu. Tracking instances as queries. *arXiv preprint arXiv:2106.11963*, 2021. 2

[54] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 3

[55] Mingmin Zhen, Shiwei Li, Lei Zhou, Jiaxiang Shang, Haoan Feng, Tian Fang, and Long Quan. Learning discriminative feature with crf for unsupervised video object segmentation. In *European Conference on Computer Vision*, 2020. 2

[56] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, 2021. 3

[57] Tianfei Zhou, Jianwu Li, Xueyi Li, and Ling Shao. Target-aware object discovery and association for unsupervised video multi-object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6985–6994, 2021. 2

[58] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[59] Fengda Zhu, Yi Zhu, Xiaodan Liang, and Xiaojun Chang. Deep learning for embodied vision navigation: A survey. *arXiv preprint arXiv:2108.04097*, 2021. 1