OW-VISCapTor: Abstractors for Open-World Video Instance Segmentation and Captioning

Anwesa Choudhuri, Girish Chowdhary and Alexander G. Schwing
University of Illinois at Urbana-Champaign
{anwesac2,girishc,aschwing}@illinois.edu

Abstract

We propose the new task ‘open-world video instance segmentation *and* captioning’. It requires to detect, segment, track *and* describe with rich captions never before seen objects. This challenging task can be addressed by developing “abstractors” which connect a vision model and a language foundation model. Concretely, we connect a multi-scale visual feature extractor and a large language model (LLM) by developing an object abstractor and an object-to-text abstractor. The object abstractor, consisting of a prompt encoder and transformer blocks, introduces spatially-diverse open-world object queries to discover never before seen objects in videos. An inter-query contrastive loss further encourages the diversity of object queries. The object-to-text abstractor is augmented with masked cross-attention and acts as a bridge between the object queries and a frozen LLM to generate rich and descriptive object-centric captions for each detected object. Our generalized approach surpasses the baseline that jointly addresses the tasks of open-world video instance segmentation and dense video object captioning by 13% on never before seen objects, and by 10% on object-centric captions.

1 Introduction

We propose the generalized task of open-world video instance segmentation *and* captioning (OW-VISCap). This task combines open-world video instance segmentation (OW-VIS) [1–5] and the generation of rich object-centric captions for objects [6]. Specifically, OW-VISCap involves detecting, segmenting, and tracking previously seen or unseen objects in a video, while simultaneously generating free-form captions for each of the detected/segmented objects. The open-world aspect makes this task widely applicable. However, it also makes this task challenging because the objects are often never seen during training, are occasionally partly or entirely occluded, the appearance and position of these objects change over time, the objects may leave the scene only to re-appear at a later time, and because generating free-form object-centric captions requires a powerful object representation as well as a strong natural language understanding. Addressing these challenges to obtain an accurate method for this task that works online is crucial in fields like autonomous systems, and augmented as well as virtual reality, among others.

OW-VISCap generalizes open-world video instance segmentation (OW-VIS) which requires to deal with never-before-seen objects [2–5, 1, 7]. For example, in Fig. 1, the trailer truck (top row) highlighted in yellow, and the lawn mower (bottom row) highlighted in green, are never seen before during training. Current OW-VIS methods suffer from the following two main issues.

Firstly, all methods on video instance segmentation, closed- or open-world, assign a one-word label to the segmented objects. However, we argue that free-form captions are more descriptive than discrete class labels [8, 9], and naturally facilitate zero-shot fine-grained captioning of objects, especially in an open-world setting. Notably, vision-language models have not been explored to unify open-world video instance segmentation and spatio-temporally dense fine-grained object captioning.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

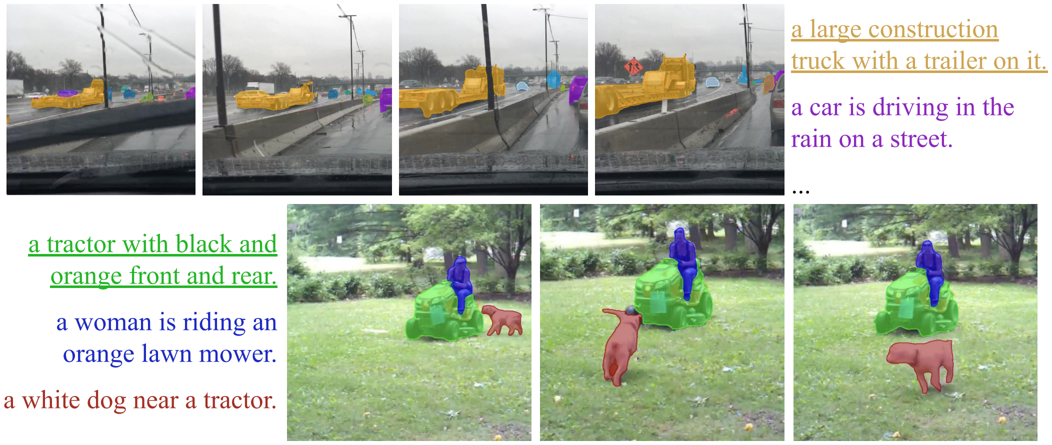

a large construction truck with a trailer on it.
a car is driving in the rain on a street.
...
a tractor with black and orange front and rear.
a woman is riding an orange lawn mower.
a white dog near a tractor.

Figure 1: Our method, OW-VISCapTor, can simultaneously detect, segment, track, and caption objects in the given video frames. The first example (top row) shows a road scene with a previously unseen trailer truck and cars that are seen during training. The second example (bottom row) shows a person on a lawn mower and a dog on the grass. The lawn mower isn't part of the training set. We generate meaningful object-centric captions even for objects never seen during training. The captions for unseen objects are underlined.

Secondly, while classic OW-VIS methods [2–5, 1, 7] rely on region-based object proposals [2–5], more recent OW-VIS methods [1, 7] develop an "abstractor" to generate object queries. Abstractors [10–14] are used to extract and project important information from one space to another depending on the task, e.g., abstractors connect pixel and language spaces in vision and language models [11, 12], pixel and object spaces in object detection/segmentation networks [13, 14, 1, 7], etc. However, OW-VIS abstractors suffer from spatial information loss because they primarily focus on a few spatial regions, leading to a loss of finer spatial details [10]. For closed-world object detection/segmentation networks, this can be compensated through extensive supervision [15–18], but it is challenging to address this issue for never-before-seen objects, i.e., when targeting an open-world setting. One way of overcoming this issue is to use a prompt as additional input from the user, ground truth or another network. The prompts can be in the form of points, bounding boxes, or text. However, these methods only work when the additional inputs are available, making them less practical in the real world.

We study these issues in the newly proposed OW-VISCap task and propose the new baseline **OW-VISCapTor**, consisting of two **O**pen-**W**orld **V**ideo **I**nstance **S**egmentation and **Cap**tioning abstrac**Tor**s. Note that OW-VISCap requires a more holistic object understanding: object representations need to be expressive and capture information not only for detecting and segmenting previously seen or unseen objects, but also capture information for generating meaningful object-centric captions. This is an important step towards generalized scene understanding. OW-VISCapTor demonstrates this holistic object understanding by simultaneously detecting, segmenting and generating object-centric captions for objects in a video. Fig. 1 shows two examples in which our method successfully detects, segments, tracks and captions both closed- and open-world objects.

OW-VISCapTor addresses the first issue via an object-to-text abstractor which uses masked attention to project the object queries into text queries that can be interpreted by a frozen LLM to generate rich object-centric captions. OW-VISCapTor addresses the second issue via an object abstractor that projects image features into object queries. Spatial information is retained by finetuning a pretrained prompt encoder which is part of the object abstractor and forms open-world object queries from points distributed evenly across video frames.

Our approach improves upon a baseline that simply combines prior work: open-world video instance segmentation (OW-VIS) [2–5, 1, 7] and dense video captioning (Dense VOC) [6]. Also note, there are no existing datasets directly collected for the generalized OW-VISCap task. Yet, OW-VIS and Dense VOC cover all aspects of OW-VISCap: open-world object discovery, video instance segmentation and dense object-centric captioning. Compared to the baseline, we improve results by 13% on the unseen categories for OW-VIS (BURST [3] data), and by 10% on the captioning accuracy

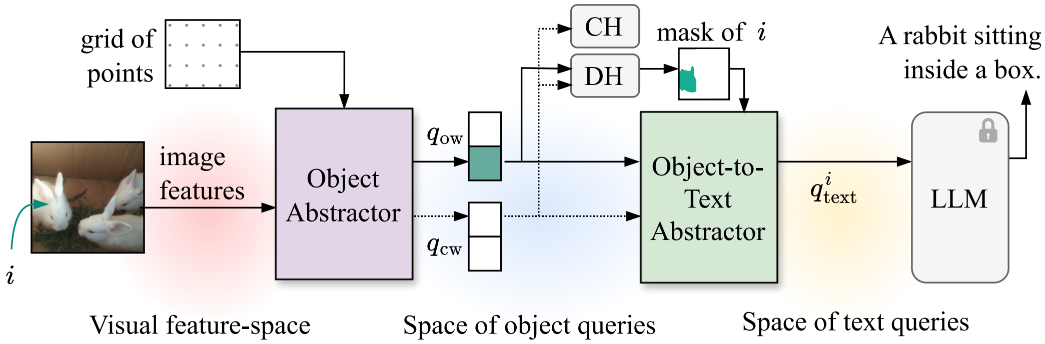


Figure 2: Overview of OW-VISCapTor (Sec. 3.1): an object abstractor (Sec. 3.2) connects the image feature space to the object query space, and an object-to-text abstractor (Sec. 3.3) connects the object query space to the text query space. DH and CH stand for detection head and classification head.

for Dense VOC (VidSTG [19] data). To further demonstrate the generalizability and efficacy of OW-VISCapTor, we compare our approach with the specialized state-of-the-art (SOTA) on OW-VIS and Dense VOC. Note that the OW-VIS SOTA can't be used for Dense VOC and vice-versa. Our generalized approach improves upon individual SOTA methods by $\sim 6\%$ on the previously unseen (uncommon) categories in the BURST [3] data, and by $\sim 7\%$ on the captioning accuracy for detected objects on the VidSTG [19] data. We also perform similar to the specialized state-of-the-art on the closed-world video instance segmentation task on the OVIS [20] data, demonstrating generalizability.

2 Related Work

2.1 Abstractors as Versatile Projectors

Abstractors in MLLMs. Abstractors have been immensely successful in Multimodal Large Language Models (MLLMs), connecting frozen vision encoders and a frozen LLM. BLIP-2 [11] successfully adapted LLMs to visual tasks, showing notable zero-shot generalization and in-context learning capabilities. More recently, abstractors are used to enhance MLLMs through visual instruction tuning [12, 21, 10]. However, to our best knowledge, abstractors have not been explored for fine-grained captioning of objects in videos in the open-world, which is of interest here.

Abstractors for object discovery. Object abstractors can generate powerful object queries, useful for closed-world object detection and segmentation in both images [14, 22, 13] and videos [15–18]. More recently, abstractors have been used for open-world object discovery [1, 7]. However, they suffer from a spatial information loss because they primarily focus on a few spatial regions, leading to a loss of finer spatial details [10]. This can be compensated through extensive supervision in the closed-world [15–18], but is challenging to address in the open-world for unseen objects. Prompt-based methods [23–25] can overcome this information loss, but use of prompts is often not realistic. In this work, we operate in a promptless open-world setting.

2.2 Generalized Video Understanding Tasks

Recently, there has been progress in unifying different video related tasks. TubeFormer [26], Unicorn [27], and CAROQ [18] unify different video segmentation tasks in the closed world. DVOC-DS [6] unifies the tasks of detecting (but not segmenting) closed-world objects in videos and captioning those closed-world objects. In this work, we explore the task of detecting and segmenting both closed- and open-world objects in videos, and captioning these objects.

Video understanding often starts from a strong generalized image understanding. Some methods [22, 13, 28] unify different image segmentation methods, and provide a baseline for many different video understanding tasks [15, 18, 17, 16]. X-Decoder [24] unifies different image segmentation tasks along with the task of referring image segmentation. SAM [23] introduces a vision foundation model, that primarily performs prompt-based open-world image segmentation, and can be used for many

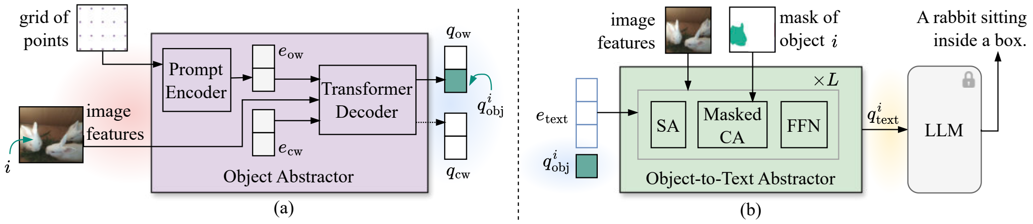


Figure 3: The proposed abstractors. (a) The **object abstractor** generates spatially rich open-world object queries q_{ow} from open-world embeddings e_{ow}, and closed-world object queries q_{cw} from closed-world embeddings e_{cw}. The open-world embeddings e_{ow} are generated by encoding a grid of points via a prompt encoder. The closed-world embeddings are learnt. (b) The **object-to-text abstractor** generates the object-centric text queries (e.g., q^i_{text} for the i^{th} object) that the frozen LLM uses for object-centric captioning. There are L transformer blocks in the object-to-text abstractor, each one consisting of self-attention (SA), masked cross-attention (Masked CA), and a feed forward network (FFN).

downstream tasks. Different from these works, we develop a generalized method for videos that tackles segmentation and object-centric captioning for both open- and closed-world objects.

2.3 Specialized Video Understanding Tasks

We briefly review OW-VIS, Dense VOC and VIS, and detail these tasks in Appendix A.

Open-world video instance segmentation. OW-VIS methods can be categorized into prompt-less methods [5, 4, 1, 7, 3] that either operate on classic region-based object proposals or suffer from spatial information loss; or promp-based methods that use prompts in the form of masks [29–37], words [25], points [23], etc. We operate in a prompt-less setting, but spatially enrich an object-abstractor to generate open-world object queries.

Dense video object captioning. DVOC-DS [6] performs object-centric captioning of closed-world objects, but cannot caption multiple action segments or process long videos like many other video models [38–40] . Unlike DVOC-DS [6], we operate in the open-world, leverage masked attention for dense video object captioning and can address the aforementioned drawbacks.

Closed-world video instance segmentation. Methods for closed-world VIS either rely on classical region-based object proposals [41, 20, 42–50], or on abstractor-based object-queries [51, 16–18, 15, 52, 53]. Differently, in this work, we explore abstractors to generate both closed- and open-world query-based proposals.

3 Abstractors for Open-World Video Instance Segmentation and Captioning

We propose to jointly address the tasks of 1) open-world video instance segmentation and 2) object-centric captioning: given a video, we jointly detect, segment, track and caption object instances in a video. Importantly, object instance categories may not be part of the training set (e.g., the trailer truck in Fig. 1 (top row)), placing our goal in an open-world setting. To achieve this goal, we develop an approach which first breaks a given video into short clips, each consisting of T frames. Each clip is processed using our OW-VISCapTor. We discuss merging of the results of each clip in Appendix C.

We provide an overview of OW-VISCapTor in Sec. 3.1. We then discuss our contributions: (a) the object abstractor that generates spatially-rich open-world object queries (Sec. 3.2), along with our proposed inter-query contrastive loss, and (b) the object-to-text abstractor that uses masked cross-attention for object-centric captioning (Sec. 3.3). We discuss the final training objective in Sec. 3.4.

3.1 Overview

Fig. 2 provides an overview of our OW-VISCapTor method. OW-VISCapTor consists of two abstractors: an object abstractor (Sec. 3.2 and Fig. 3 (a)) and an object-to-text abstractor (Sec. 3.3 and

Fig. 3 (b)). The object abstractor connects the visual space of image features ($\mathbb{R}^{HWT\times C}$) to the space of object queries ($\mathbb{R}^{N_{\text{obj}}\times C}$). Here, $N_{\text{obj}} = N_{\text{obj,ow}} + N_{\text{obj,cw}}$ is the total number of object queries, which includes the total number of open-world and closed-world queries, $N_{\text{obj,ow}}$ and $N_{\text{obj,cw}}$. C refers to the channel dimension for each object query and image feature. H, W and T refer to the height, width and clip-length. The object-to-text abstractor connects the space of object queries ($\mathbb{R}^{N_{\text{obj}}\times C}$) and text queries ($\mathbb{R}^{(N_{\text{text}}+1)\times C}$), to generate fine-grained object-centric captions. Here, N_{text} is the total number of text queries which are concatenated with individual object queries (hence the '$+1$') to generate text queries for each object.

To deal with the open-world setting, i.e., to ensure that we can detect, segment, track and caption never before seen object instances, the object abstractor generates spatially rich open-world object queries, $q_{\text{ow}} \in \mathbb{R}^{N_{\text{obj,ow}}\times C}$. These open-world object queries are in addition to closed-world object queries $q_{\text{cw}} \in \mathbb{R}^{N_{\text{obj,cw}}\times C}$, commonly used in prior work [15, 13, 18, 16, 17]. The open-world object queries are enriched spatially by a prompt encoder (shown in Fig. 3 (a)) that encodes a grid of points into prompt representations. Note, by using open-world object queries, we discover new and diverse open-world objects without needing additional prompts.

We use $q_{\text{obj}} = [q_{\text{ow}}, q_{\text{cw}}]$ to denote all the object queries obtained from the object abstractor. The i^{th} object query $q^i_{\text{obj}} \in \mathbb{R}^{1\times C}$ is concatenated with learnt text embeddings $e_{\text{text}} \in \mathbb{R}^{N_{\text{text}}\times C}$. The concatenated object query and text embeddings are continuously modulated in the object-to-text abstractor by combining them with image features so as to generate meaningful text queries $q^i_{\text{text}} \in \mathbb{R}^{(N_{\text{text}}+1)\times C}$ for the i^{th} object. This is shown in Fig. 3 (b). Note, q^i_{text} is used by the frozen LLM to generate object-centric captions. The segmentation mask for the i^{th} object generated in the detection head is used to mask the attention in the object-to-text abstractor, as described in Sec. 3.3. We find this design to enable the LLM to generate more object-centric captions.

Both open- and closed-world object queries are processed by our object-to-text abstractor and LLM which yields an object-centric caption, and a detection head (DH in Fig. 2) which yields either a segmentation mask or a bounding-box. The closed-world object queries are further processed by a classification head (CH in Fig. 2) which yields a category label.

3.2 Object Abstractor

The object abstractor is detailed in Fig. 3 (a). It consists of a prompt encoder, a transformer decoder and closed-world trainable embeddings e_{cw}. The closed-world embeddings e_{cw} are modulated in the transformer decoder to generate the closed-world object queries q_{cw}. We discuss the generation of open-world object queries next.

Spatially-rich open-world object queries. To help discover new objects, the object abstractor introduces open-world object queries q_{ow} in addition to the commonly used closed world object queries. Our open-world object queries are generated from open-world embeddings e_{ow}, which are continuously modulated in the transformer decoder by combining them with image features via masked attention, following [13, 15].

Prompt encoder. The open-world embeddings e_{ow} are generated in the prompt encoder. An illustration is provided in Fig. 3 (a). We encode a grid of equally spaced points along the height and width of the frames of the clip, using the prompt encoder employed in SAM [23]. The use of equally spaced points encourages the open-world object queries to focus on different regions of the video frames, making them spatially rich and encouraging object discovery throughout the frames. This also encourages the open-world object queries to be diverse from one another.

Inter-query contrastive loss. We introduce an inter-query contrastive loss $\mathcal{L}_{\text{cont}}$ to ensure that the object queries are different from each other. For closed-world objects, this loss helps in removing highly overlapping false positives. For open-world objects, it helps in the discovery of new objects. Formally,

$$\mathcal{L}_{\text{cont}} = -\sum_{i,j} L_1(q^i_{\text{obj}}, q^j_{\text{obj}}), \tag{1}$$

where q^i_{obj} and q^j_{obj} are the i^{th} and the j^{th} objects and $i \neq j$. L_1 refers to the L1 distance. Via this loss, we *maximize* the L1 distance between the object queries, i.e., we encourage that the object queries differ from each other.

3.3 Object-To-Text Abstractor

The object-to-text abstractor (Fig. 3 (b)) connects the space of object queries and the space of text queries. It consists of L transformer blocks, each block consisting of self-attention (SA), masked cross-attention (masked CA) and a feed forward network (FFN) as shown in Fig. 3 (b) to generate object-centric text queries. Next, we discuss masked cross-attention.

Masked cross-attention for object-centric captioning. Masked cross-attention involves attending within the foreground region of the predicted mask for each object query. Concretely, to generate object-centric text queries q^i_{text} for each object i, we restrict the cross attention in the object-to-text abstractor by using the segmentation mask of the object generated by the detection head. Intuitively, this enables the model to focus on local object-centric features, which are sufficient to update the text queries. Importantly, note that context information from the video frames can be gathered through the self-attention layers. The proposed design hence doesn't take away any information. It rather provides the same information in clearly separated layers.

Formally, for the i^{th} object we compute the query features $X_l^{\text{cap},i} \in \mathbb{R}^{(N_{\text{text}}+1)\times C}$ obtained from the l^{th} object-to-text transformer layer via

$$X_l^{\text{cap},i} = \text{softmax}(\mathcal{M}^i + Q_l^i K_l^T)V_l + X_{l-1}^{\text{cap},i}. \tag{2}$$

Here, $\mathcal{M}^i \in \{0, -\infty\}^{1\times HWT}$ is the attention mask in the object-to-text abstractor such that at feature location (x, y), $\mathcal{M}^i(x, y) = 0$, if $M^i(x, y) = 1$ and $\mathcal{M}^i(x, y) = -\infty$, if $M^i(x, y) = 0$. M^i is the binary mask obtained from the detection head. Moreover, $K_l, V_l \in \mathbb{R}^{HWT\times C}$ are the linearly transformed image features. To initialize, we let $X_0^{\text{cap},i} = [q^i_{\text{obj}}, e_{\text{text}}]$, where $q^i_{\text{obj}} \in \mathbb{R}^{1\times C}$ is the i^{th} object query obtained from the object abstractor and $e_{\text{text}} \in \mathbb{R}^{N_{\text{text}}\times C}$ are the learnt text embeddings shown in Fig. 2 and introduced in Sec. 3.1.

3.4 Training

Our training objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{cap}} + \mathcal{L}_{\text{cw}} + \mathcal{L}_{\text{ow}}.$$

Here, $\mathcal{L}_{\text{cont}}$ is the inter-query contrastive loss discussed in Sec. 3.2. $\mathcal{L}_{\text{cap}}$ is the standard captioning loss introduced by DVOC-DS [6]. $\mathcal{L}_{\text{cw}}$ is the closed-world loss following prior work [13]. To compute $\mathcal{L}_{\text{cw}}$, the ground truth objects are first matched with the predicted closed-world objects; the optimal matching is used to compute the final closed-world loss $\mathcal{L}_{\text{cw}}$. This loss consists of a detection/segmentation loss and a cross-entropy loss for predicting the closed-world object categories. $\mathcal{L}_{\text{ow}}$ is the open-world loss, which consists of only a detection/segmentation loss, unlike $\mathcal{L}_{\text{cw}}$. The open-world loss is detailed in Appendix B. Note, that the training data consists of only closed-world objects. We match the closed-world ground truth objects twice, once with the predicted open-world objects to compute $\mathcal{L}_{\text{ow}}$, and once with the predicted closed-world objects to compute $\mathcal{L}_{\text{cw}}$. We train the object abstractor, the object-to-text abstractor, and the image-feature extractor that generates the image-features as illustrated in Fig. 2. The parameters of the LLM are frozen.

4 Experiments

We evaluate OW-VISCapTor on the diverse tasks of open-world video instance segmentation (OW-VIS) and dense video object captioning (Dense VOC). Note that there is no dedicated dataset for our proposed task of open-world video instance segmentation *and* captioning (OW-VISCap). Hence we use the two aforementioned tasks and evaluate the three different aspects of our approach: open-world capability, video instance segmentation and video object captioning. In the following subsections, we first discuss the datasets and evaluation metrics used in our evaluation (Sec. 4.1). We then compare our performance to a baseline that jointly addresses OW-VIS and Dense VOC, and also to specialized methods that address these two tasks individually (Sec. 4.2). We demonstrate how our contributions result in better performance through an ablation study (Sec. 4.3). Finally, we show qualitative results (Sec. 4.4). In Appendix D, we also show results on closed-world VIS.

Table 1: Results on OW-VIS (left) and Dense VOC (right). Onl. refers to online frame-by-frame processing. The columns highlighted in blue (OWTA for Unseen categories in OW-VIS and CapA for Dense VOC) highlight the 'open-world' and 'captioning' capabilities of different methods. The best scores are highlighted in bold font, and the second-best scores are underlined.

		OW-VIS (OWTA)			Dense VOC				
Method	Mode	Unseen	Overall	Seen	CapA	CHOTA	DetA	AssA	$\mathrm{AP_M}$
OWTB [5]	onl.	38.8	55.8	59.8	-	-	-	-	-
Mask2Former [13]+STCN [35]	onl.	25.0	64.6	71.0	-	-	-	-	-
Mask2Former [13]+DEVA [7]	onl.	42.3	**69.5**	**74.6**	-	-	-	-	-
EntitySeg [54]+DEVA [7]	onl.	49.6	68.8	72.7	-	-	-	-	-
DVOC-DS (joint training) [6]	off.	-	-	-	36.8	51.6	**65.5**	**56.9**	**69.3**
DVOC-DS (disjoint training) [6]	off.	-	-	-	10.0	28.0	45.9	48.0	39.8
Mask2Former [13]+DEVA [7]+ BLIP2 [11]	onl.	42.3	**69.5**	**74.6**	34.0	48.5	59.6	$\underline{56.4}$	60.1
OW-VISCapTor+CAROQ [18] (online)	*onl.*	*$\underline{50.0}$*	*66.1*	*63.0*	***43.9***	***53.1***	*$\underline{60.1}$*	*54.0*	*62.6*
OW-VISCapTor+DEVA [7] (online)	*onl.*	***55.2***	*$\underline{69.0}$*	*$\underline{73.5}$*	*$\underline{40.1}$*	*$\underline{51.7}$*	*60.0*	*56.3*	*$\underline{63.0}$*

Table 2: Ablation on the BURST [3] validation data. 'w/o p.e.' refers to without prompt encoder; 'w/o $\mathcal{L}_{\mathrm{cont}}$' refers to without contrastive loss.

Method	OWTA		
	Overall	Seen	Unseen
Ours	**55.5**	**58.2**	**43.8**
w/o p.e.	54.2	57.7	41.2
w/o $\mathcal{L}_{\mathrm{cont}}$	53.5	56.1	41.6

Table 3: Ablation on the VidSTG [19] data. 'w/o m.a.' refers to without masked attention. 'bb. cap.' and 'en. bb. cap.' refers to bounding box captioning and enlarged bounding box captioning.

Method	CHOTA	DetA	AssA	CapA
Ours	**51.0**	56.1	54.0	**43.9**
w/o m.a.	39.5	56.1	54.0	20.3
bb. cap.	48.1	56.1	54.0	36.6
en. bb. cap.	49.2	56.1	54.0	39.4

4.1 Datasets and Evaluation Metrics

We evaluate our approach on the OW-VIS and Dense VOC tasks. For OW-VIS, we use the BURST dataset [3]. For Dense VOC, we use the VidSTG dataset [19]. Note, VidSTG [19] has bounding box and tracking identity for all objects, but captions are not exhaustively provided. Following DVOC-DS [6] the captioning loss for missing captions is removed during training and data with missing captions isn't evaluated at test time.

For OW-VIS, we use the standard evaluation metrics of open-world tracking accuracy (OWTA) [3] for all, common (seen) and uncommon (unseen) categories. For Dense VOC, we use the captioned higher order tracking accuracy (CHOTA) [6], which depends on the detection accuracy (DetA), association accuracy (AssA), and captioning accuracy (CapA). We also report the frame-based METEOR score ($\mathrm{AP_M}$).

4.2 Main Results

We compare OW-VISCapTor with a generalized baseline, as well as specialized SOTA methods on the tasks of OW-VIS and Dense VOC. The results are summarized in Tab. 1. Even when compared to specialized SOTA on individual tasks, our method is able to achieve the best results (highlighted in bold) or the second-best results (underlined). Tab. 1 (left) shows results for the OW-VIS task on the BURST dataset [3]. We report the open-world tracking accuracy for all, common (seen) and uncommon (unseen) categories. Tab. 1(right) shows results on the Dense VOC task.

Comparison with generalized baseline. We create a generalized baseline (Mask2Former [13] + DEVA [7] + BLIP2 [11]) that is able to address the tasks of OW-VIS and Dense-VOC jointly. The generalized baseline consists of Mask2Former [13] trained on the VidSTG [19] dataset for object detection and integrated with DEVA [7] for tracking. The per-frame predictions are first enlarged by 10% to provide more overall image context. The enlarged bounding boxes are then used to crop the images for captioning using BLIP-2 [11]. Note that Mask2Former [13]+DEVA [7] is a specialized previous SOTA on the OW-VIS task.

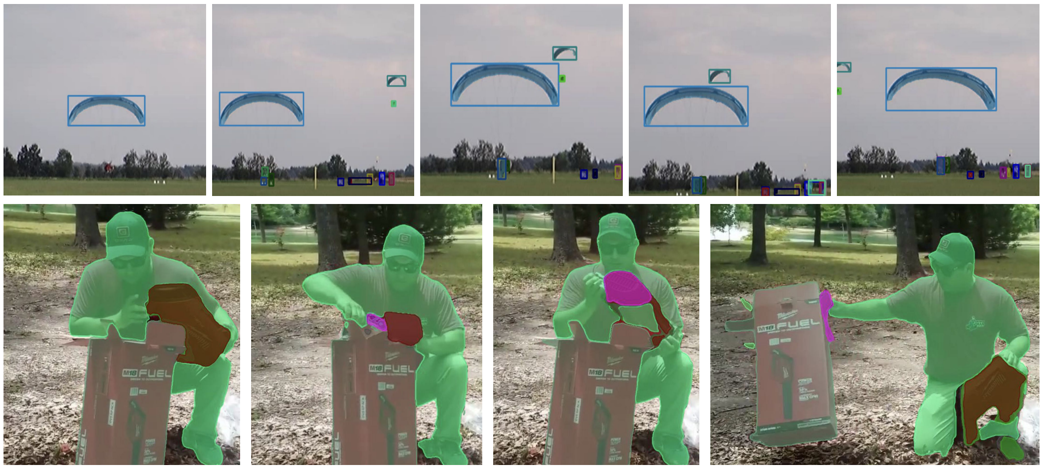


Figure 4: Example from the BURST validation data. The masks are superimposed on the objects. The top row shows examples of parachutes in the air and people on the grass. The parachutes belong to the uncommon object category, i.e., parachutes were never seen during training. Our approach detects and retains the identities of the blue and the green parachutes as the green parachute crosses the blue one. The bottom row shows a person unboxing a leaf blower. The carton of the leaf blower (gray mask), the leaf blower (maroon mask), and the plastic wrapper (pink mask) are never seen during training. We can consistently detect, segment, and track them along with the person (common object category during training).

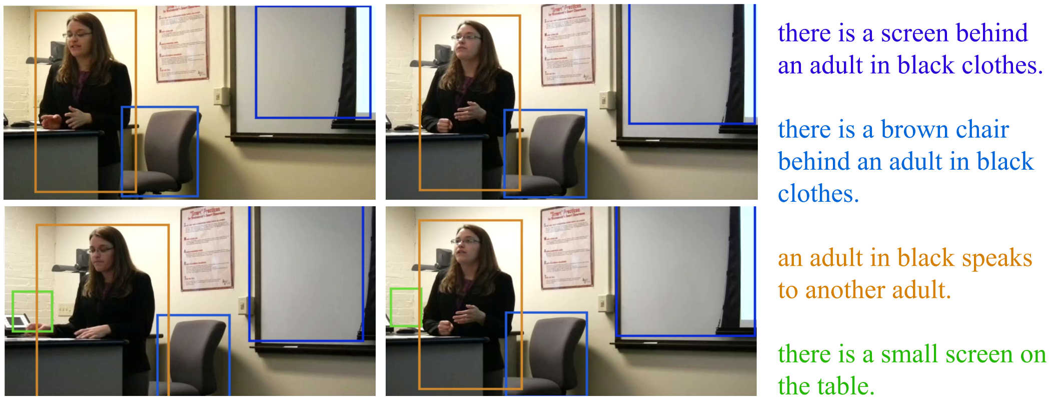

there is a screen behind an adult in black clothes.
there is a brown chair behind an adult in black clothes.
an adult in black speaks to another adult.
there is a small screen on the table.

Figure 5: An example from the VidSTG data. Our approach is able to detect and track objects in the scene consistently and to generate meaningful object-centric captions for each of the detected objects.

OW-VISCapTor outperforms Mask2Former [13]+DEVA [7]+BLIP2 [11] consistently across both datasets on unseen categories and object-captioning, demonstrating that our object queries are expressive enough to simultaneously detect, segment, track and caption seen or unseen objects. Mask2Former [13]+DEVA [7]+BLIP2 [11], in-spite of being SOTA on the previously seen object categories, struggles on the Dense VOC task. This is due to the lack of overall image-based context for captioning.

Comparison with specialized SOTA. OW-VISCapTor, integrated with DEVA [7] for temporal association of objects, achieves the state-of-the-art on the uncommon object categories (Tab. 1 (left)), improving upon the next best method (EntitySeg [54]+DEVA [7]) by ~ 6 points on the BURST [3] validation data. For the common categories, our method ranks 2^{nd} in the BURST validation data. We use a SwinL [55] backbone, and a clip-length of $T = 1$ in this setting.

Our method, when integrated with CAROQ [18] or DEVA [7] for temporal association, outperforms DVOS-DS [6] on the captioning accuracy (CapA), demonstrating that our object-to-text abstractor with masked cross-attention (Sec. 3.3) is effective in generating object-centric captions. We improve upon DS-VOC on the overall CHOTA metric, even though we slightly underperform on DetA and AssA. Note that DVOS-DS is an offline method: the entire object trajectories are used for generating the captions. Hence DVOS-DS cannot process videos with more than 200 frames. This is in contrast to our online method, where we sequentially process short video clips (of length $T = 2$ for CAROQ [18] and of length $T = 1$ for DEVA [7]). DVOS-DS uses a ViT [56] backbone, whereas we use SwinL [55], which leads to a difference in DetA scores. We provide additional details on the merging of video clips in Appendix C.

Note that, even though the OW-VISCap task focuses on generalizability, OW-VISCapTor outperforms specialized methods on the **open-world** metrics and the **captioning** metrics, demonstrating the effectiveness of our contributions: the object abstractor that processes our novel open-world object queries, and the novel object-to-text abstractor that generates rich object-centric captions.

4.3 Ablation Studies

Spatially-rich open-world object queries. Tab. 2 (first and second row) shows that the spatially-rich open-world object queries q_{ow}, described in Sec. 3.2, help in discovering new objects. In Tab. 2, 'w/o p.e.' refers to the setting without the prompt encoder encoding spatial prompts into the open-world embeddings e_{ow}. The open-world embeddings e_{ow} are trained like the closed-world embeddings e_{cw}. We observe that the performance drops by 2.6 points for uncommon categories compared to 'Ours', even though the number of object queries are exactly the same in both settings. This highlights that the object abstractor suffers from spatial information loss if not augmented with a prompt encoder to encode spatial points.

Contrastive loss. Tab. 2 (first and last row) shows that the contrastive loss $\mathcal{L}_{\text{cont}}$, described in Sec. 3.2, helps in detecting both the common (seen) and uncommon (unseen) categories of objects. The performance drops by ~ 2 points for both the common and uncommon categories for the setting 'w/o $\mathcal{L}_{\text{cont}}$', i.e., when the contrastive loss is not used. The contrastive loss helps in removing highly overlapping false positives in the closed-world setting and in discovering new objects in the open-world setting.

Masked attention in object-to-text abstractor. Tab. 3 shows that masked attention in the object-to-text abstractor, described in Sec. 3.3, helps in object-centric captioning. The second row 'w/o m.a.' of Tab. 3 refers to the setting without masked attention, i.e., the entire image-feature is used to calculate the cross-attention in the object-to-text abstractor. The object-centric context is only accumulated by concatenating the i^{th} object query with the learnt text embeddings, as discussed in Sec. 3.3 and shown in Fig. 2. We observe that the captioning accuracy CapA drops by 23 points, indicating that concatenating the object query with the text embeddings is not sufficient for an object-centric focus. The third row in Tab. 3, 'bb. cap.' (bounding box captioning), pursues the opposite setting. Here, the images are cropped based on the object bounding box predictions in the detection head. The cropped images are directly used for captioning, ensuring that both the self- and cross-attention blocks in the object-to-text transformer operate on object-centric features. Note, that we don't use masked attention in this setting. We observe a drop in CapA of 5 points. Although cropping helps in retaining the object-centric information, the overall context from the entire image is missing. The fourth row in Tab. 3, 'en. bb. cap.' (enlarged bounding box captioning), shows a similar setting as the third row, but the bounding boxes are first enlarged by 10% to provide more overall image context. The enlarged bounding boxes are then used to crop the images for captioning. We observe a drop in CapA of 3 points, indicating that enlarging the bounding boxes helps but is not sufficient to provide overall context. This is also highlighted in the third-last row and the last row in Tab. 1.

4.4 Qualitative Results

Fig. 1 shows results on the BURST [3] validation data. OW-VISCap is able to simultaneously detect, segment, track and caption objects. The objects belong to both the open- and closed-world. Note that the BURST [3] data doesn't provide object-centric captions for training, hence our object-to-text abstractor was not trained on BURST [3] but only on VidSTG. We find this object-to-text abstractor to be effective in generating meaningful object-centric captions even for objects never seen during training. Fig. 4 shows two examples from the BURST validation data. We can consistently detect, segment, and track previously seen and unseen objects. Fig. 5 shows an example from the

VidSTG [19] data. Our method can detect, track and caption objects. Additionally, we discuss some failure modes of our method in Appendix G.

5 Conclusion

We introduce OW-VISCapTor: two abstractors to *jointly detect, segment, track, and caption previously seen or unseen objects in videos*. The developed object abstrator generates spatially-rich open-world object queries which encourage discovery of previously unseen objects without the need of additional user-input. Instead of assigning a fixed label to detected objects, we introduce an object-to-text abstractor that uses masked cross-attention to generate rich object-centric captions for each object. **Societal Impact**. Our method can be used to segment and describe never before seen objects. This capability could be beneficial in assistive technologies for the blind, as well as in AR/VR. Although we don't see any direct ethical concerns, research in this direction makes video-processing technology increasingly accessible. This could encourage and increase malicious use and potentially create issues regarding unethical surveillance and privacy threats. This calls for stricter security measures both at a personal level (like password protecting video data), and societal level (like regulating open-source dataset and model releases).

6 Acknowledgements

This work is supported in party by the Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021-32799/project accession no. 1024178 from the USDA National Institute of Food and Agriculture: NSF/USDA National AI Institute: AIFARMS. We also thank the Illinois Center for Digital Agriculture for seed funding for this project. Work is also supported in part by NSF under grants 2008387, 2045586, 2106825, MRI 1725729.

References

[1] Omkar Thawakar, Sanath Narayan, Hisham Cholakkal, Rao Muhammad Anwer, Salman Khan, Jorma Laaksonen, Mubarak Shah, and Fahad Shahbaz Khan. Video instance segmentation in an open-world. *arXiv preprint arXiv:2304.01200*, 2023.

[2] Yang Liu, Idil Esen Zulfikar, Jonathon Luiten, Achal Dave, Deva Ramanan, Bastian Leibe, Aljoša Ošep, and Laura Leal-Taixé. Opening up open world tracking. In *CVPR*, 2022.

[3] Ali Athar, Jonathon Luiten, Paul Voigtlaender, Tarasha Khurana, Achal Dave, Bastian Leibe, and Deva Ramanan. Burst: A benchmark for unifying object recognition, segmentation and tracking in video. In *WACV*, 2023.

[4] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021.

[5] Yang Liu, Idil Esen Zulfikar, Jonathon Luiten, Achal Dave, Deva Ramanan, Bastian Leibe, Aljoša Ošep, and Laura Leal-Taixé. Opening up open world tracking. In *CVPR*, 2022.

[6] Xingyi Zhou, Anurag Arnab, Chen Sun, and Cordelia Schmid. Dense video object captioning from disjoint supervision. *arXiv preprint arXiv:2306.11729*, 2023.

[7] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *ICCV*, 2023.

[8] Yanxin Long, Youpeng Wen, Jianhua Han, Hang Xu, Pengzhen Ren, Wei Zhang, Shen Zhao, and Xiaodan Liang. Capdet: Unifying dense captioning and open-world detection pretraining. In *CVPR*, 2023.

[9] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. Grit: A generative region-to-text transformer for object understanding. *arXiv preprint arXiv:2212.00280*, 2022.

[10] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. *arXiv preprint arXiv:2312.06742*, 2023.

[11] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[12] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023.

[13] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[14] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[15] 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.

[16] 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.

[17] De-An Huang, Zhiding Yu, and Anima Anandkumar. Minvis: A minimal video instance segmentation framework without video-based training. In *NeurIPS*, 2022.

[18] Anwesa Choudhuri, Girish Chowdhary, and Alexander G. Schwing. Context-aware relative object queries to unify video instance and panoptic segmentation. In *CVPR*, 2023.

[19] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *CVPR*, 2020.

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

[21] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[22] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021.

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[24] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *CVPR*, 2023.

[25] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *CVPR*, 2022.

[26] Dahun Kim, Jun Xie, Huiyu Wang, Siyuan Qiao, Qihang Yu, Hong-Seok Kim, Hartwig Adam, In So Kweon, and Liang-Chieh Chen. Tubeformer-deeplab: Video mask transformer. In *ICCV*, 2022.

[27] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*, 2022.

[28] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *CVPR*, 2023.

[29] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

[30] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019.

[31] Lu Zhang, Zhe Lin, Jianming Zhang, Huchuan Lu, and You He. Fast video object segmentation via dynamic targeting network. In *ICCV*, 2019.

[32] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018.

[33] Xuhua Huang, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Fast video object segmentation with temporal aggregation network and dynamic template matching. In *CVPR*, 2020.

[34] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[35] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *NeurIPS*, 2021.

[36] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019.

[37] Xuhua Huang, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Fast video object segmentation with temporal aggregation network and dynamic template matching. In *CVPR*, 2020.

[38] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.

[39] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, 2022.

[40] 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.

[41] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019.

[42] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *ICCV*, 2020.

[43] Ali Athar, Sabarinath Mahadevan, Aljosa Osep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020.

[44] 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.

[45] Anwesa Choudhuri, Girish Chowdhary, and Alexander G. Schwing. Assignment-space-based multi-object tracking and segmentation. In *ICCV*, 2021.

[46] Guillem Braso and Laura Leal-Taixe. Learning a neural solver for multiple object tracking. In *CVPR*, June 2020.

[47] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *ICML*, 2020.

[48] Jonathon Luiten, Tobias Fischer, and Bastian Leibe. Track to reconstruct and reconstruct to track. *RAL*, 2020.

[49] Paul Voigtlaender, Michael Krause, Aljoša Ošep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-object tracking and segmentation. In *CVPR*, 2019.

[50] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Huan Huang, Shilei Wen, Errui Ding, and Liusheng Huang. Segment as points for efficient online multi-object tracking and segmentation. In *ECCV*, 2020.

[51] 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.

[52] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *CVPR*, 2022.

[53] Junfeng Wu, Yi Jiang, Wenqing Zhang, Xiang Bai, and Song Bai. Seqformer: a frustratingly simple model for video instance segmentation. *arXiv preprint arXiv:2112.08275*, 2021.

[54] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Philip Torr, Zhe Lin, and Jiaya Jia. Open world entity segmentation. *PAMI*, 2022.

[55] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[56] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[57] James Munkres. Algorithms for the assignment and transportation problems. *J-SIAM*, 1957.

[58] Adrià Caelles, Tim Meinhardt, Guillem Brasó, and Laura Leal-Taixé. Devis: Making deformable transformers work for video instance segmentation. *arXiv preprint arXiv:2207.11103*, 2022.

[59] Lei Ke, Henghui Ding, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Video mask transfiner for high-quality video instance segmentation. In *ECCV*, 2022.

[60] Rajat Koner, Tanveer Hannan, Suprosanna Shit, Sahand Sharifzadeh, Matthias Schubert, Thomas Seidl, and Volker Tresp. Instanceformer: An online video instance segmentation framework. *arXiv preprint arXiv:2208.10547*, 2022.

[61] Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. Vita: Video instance segmentation via object token association. *arXiv preprint arXiv:2206.04403*, 2022.

[62] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020.

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

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

Supplementary Material — OW-VISCapTor: Abstractors for Open-World Video Instance Segmentation and Captioning

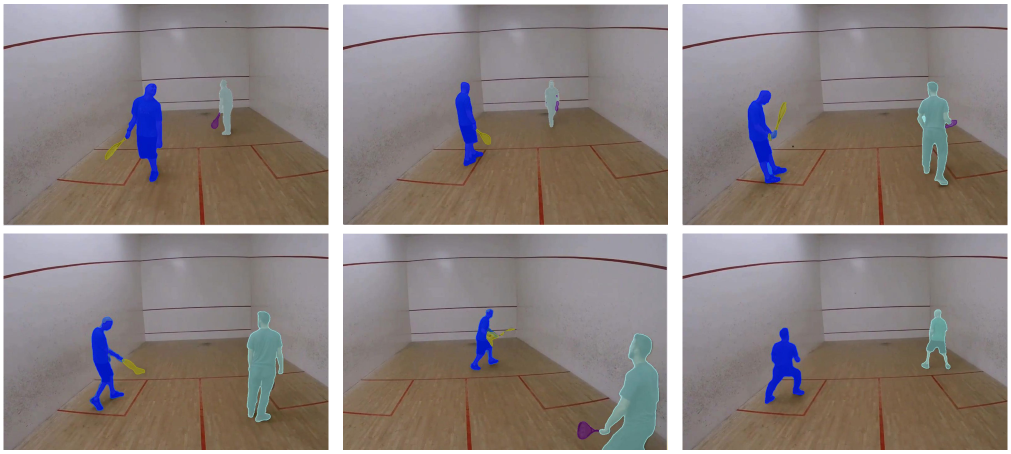


Figure 6: Results on the BURST dataset. We successfully segment and track closed-world objects (e.g., persons), and open-world objects (e.g., rackets) throughout the video.

This is the supplementary material of the **OW-VISCapTor** paper, where we develop **O**pen-**W**orld **V**ideo **I**nstance **S**egmentation and **Cap**tioning abstrac**Tor**s. We test our approach on the task of open-world video instance segmentation (OW-VIS) and dense video object captioning (Dense VOC) in Sec. 4 on the BURST [3] and VidSTG [19] datasets respectively. Fig. 6 shows additional results on the BURST [3] dataset. We successfully segment and track the closed-world objects: persons, and the open-world objects: rackets, throughout the video. Fig. 7 shows results on the VidSTG [19] dataset. The detected objects are tracked throughout the video and our method generates meaningful captions for each object.

In this supplementary material, we first discuss additional related works in Sec. A. In Sec. B, we discuss how to train the open-world object queries introduced in Sec. 3 of the main paper and then detail the merging of video-clips in Sec. C. We provide additional quantitative results (Sec. D) and qualitative analysis (Sec. E) to support the contributions we made in Sec. 3 of the main paper. Finally, we discuss the implementation details (Sec. F) and limitations (Sec. G) of our approach.

A Additional Related Work

A.1 Specialized Video Understanding Tasks

We over-viewed some specialized video understanding tasks in Sec. 2 of the main paper. In this section, we provide additional details on these methods.

Open-world video instance segmentation. OW-VIS methods can be categorized into prompt-less and prompt-based methods. Prompt-less methods [5, 4, 1, 7, 3] discover new objects based on an objectness score. Recent abstractor-based methods [1, 7] have shown to outperform classic methods using region-based object proposals [5, 3]. However, these methods suffer from spatial information loss [10]. Differently, we use an abstractor to generate spatially rich open-world object queries to address the OW-VIS task. Prompt-based methods rely on prompts, i.e., prior knowledge, to segment objects in videos. Prompts can be in the form of masks [29–37], words [25], points [23], etc. Prompts provide a way to encode additional information to offset the spatial information loss observed with abstractors. However, in a true open-world setting, such prior knowledge may not be available. We operate in such a setting.

Dense video object captioning. Dense video object captioning involves detecting, tracking, and captioning trajectories of closed-world objects in a video. For this task, DVOC-DS [6] extends the image-based GRiT [9] and trains it with a mixture of disjoint tasks. However, DVOC-DS [6] cannot caption multiple action segments within a single object trajectory because the method produces a

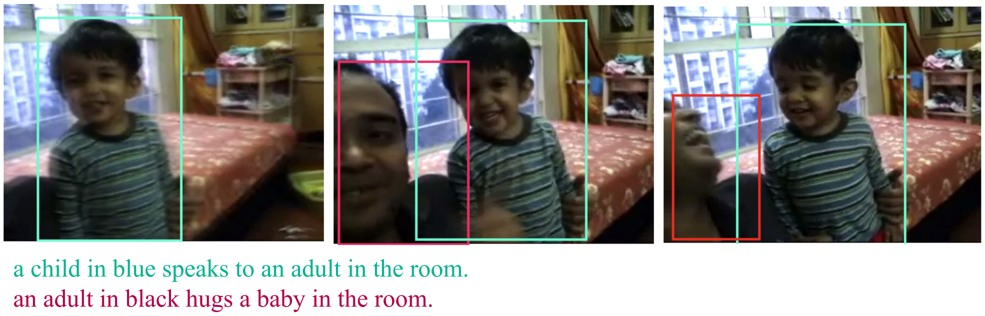

a child in blue speaks to an adult in the room.
an adult in black hugs a baby in the room.

Figure 7: Results on the VidSTG dataset. Our approach detects, tracks and generates meaningful object-centric captions for each object throughout the video.

single caption for the entire object trajectory. In addition, similar to many other video models [38–40], DVOC-DS [6] struggles with very long videos and only processes up to 200 frames. The method is further constrained to handle only known object categories and it is unclear how the method extends to an open-world setting. Unlike DVOC-DS [6], we use abstractors to generate spatially rich open-world object queries and leverage masked attention for dense video object captioning. We also process video frames sequentially using a short temporal context and hence can process long videos, as well as handle multiple action segments within a single object trajectory.

Closed-world video instance segmentation. Closed-world video instance segmentation involves simultaneously segmenting and tracking objects from a fixed category set in a video. Some works [41, 20, 42–50] rely on classical region-based proposals. Recent works [51, 16–18, 15, 52, 53] rely on abstractor-based object queries and perform significantly better at discovering closed-world objects. Differently, in this work, we explore abstractors to generate query-based proposals for the closed- and open-world setting.

A.2 Contrastive Loss for Object Queries

Contrastive losses have been used to help in video instance segmentation. OWVISFormer [1] uses a contrastive loss in the open-world setting: it ensures that assigned foreground objects are similar to each other while being different from the background objects. IDOL [16] works in the closed-world setting and uses an inter-frame contrastive loss to ensure object queries belonging to the same object across frames are similar, and object queries of different instances across frames differ. In contrast, in this work, for both the closed- and open-world setting, we use a contrastive loss to ensure that no two object queries in the foreground are similar to each other, even in the same frame.

B Open-World Loss

To encourage the discovery of new objects, we introduce an open-world loss $\mathcal{L}_{\text{ow}}$ in Sec. 3.4 of the main paper. It differs in one key aspect from a classic closed-world loss [15]: we don't have a cross-entropy loss.

We first match the ground truth objects with the open-world predictions by minimizing a matching cost using the Hungarian algorithm [57]. The optimal matching is then used to calculate the final open-world loss. Let us use $\hat{\sigma}_{\text{ow}}$ to represent the optimal matching between the ground truth objects and the open-world predictions.

Formally, the open-world loss is given by

$$\mathcal{L}_{\text{ow}} = \sum_{i} \mathcal{L}_{\text{det}}(i, \hat{\sigma}^{i}_{\text{ow}}), \tag{3}$$

where i is the object index. $\mathcal{L}_{\text{det}}(i, \hat{\sigma}^{i}_{\text{ow}})$ is the detection loss (mask loss or bounding box loss) between the ground truth object with index i and a prediction with index $\hat{\sigma}^{i}_{\text{ow}}$.

Differently, in the classic closed-world setting, the object categories are available, which enables us to train a classification head (CH in Fig. 2) using a cross-entropy loss.

Table 4: Open-world tracking accuracy (OWTA) on the BURST test dataset for uncommon (unseen), overall and common (seen) categories of objects. The best scores are highlighted in bold font, and the second-best scores are underlined.

Method	Test (OWTA)		
	Unseen	Overall	Seen
OWTB [5]	38.3	56.0	59.9
Mask2Former [13]+STCN [35]	23.9	57.5	62.9
Mask2Former [13]+DEVA [7]	44.1	**70.1**	**75.0**
EntitySeg [54]+DEVA [7]	53.0	69.5	72.9
OW-VISCapTor (ours) + DEVA [7]	**57.2**	69.2	72.4

Table 5: Results on the OVIS [20] validation data. All methods use the ResNet-50 backbone. The best performing methods are highlighted in bold font. The second best methods are underlined.

Method	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack [41]	10.8	25.3	8.5	7.9	14.9
DeVIS [58]	23.7	47.6	20.8	12.0	28.9
MinVIS [17]	25.0	45.5	24.0	13.9	29.7
VMT [59]	16.9	36.4	13.7	10.4	22.7
InstanceFormer [60]	20.0	40.7	18.1	12.0	27.1
VITA [61]	19.6	41.2	17.4	11.7	26.0
CAROQ [18]	**25.8**	47.9	**25.4**	14.2	**33.9**
OW-VISCapTor (w/o $\mathcal{L}_{\text{cont}}$)	23.2	45.2	21.7	13.5	30.1
OW-VISCapTor (ours)	25.4	**48.8**	22.8	**14.3**	32.8

C Merging of Clips

To achieve open-world video instance segmentation and captioning, OW-VISCapTor first breaks a given video into short clips, each consisting of T frames. Each clip is processed using our OW-VISCapTor. We now discuss how the predictions of individual clips are merged.

We use DEVA [7], a recent state-of-the-art for temporal association, to connect video clips temporally, as highlighted in Tab. 1 of the main paper. DEVA [7] develops a class-agnostic temporal propagation approach to track detected or segmented objects. However, DEVA [7] permits clip-length of $T = 1$. To incorporate more temporal context, we also adopt CAROQ [18], where the object queries are propagated again and again for subsequent video clips, thereby carrying temporal information forward, and implicitly tracking the objects. CAROQ [18] can operate with a clip-length of $T \geq 1$. The effect of clip-length while using CAROQ [18] is ablated for the Dense VOC task in Tab. 6.

D Additional Quantitative Results

Results on BURST test data. Tab. 4 shows the results of our method on the BURST test data. Our method performs the best on unseen categories, which is consistent with the results on the BURST [3] validation data shown in Sec. 4.2.

Results on video instance segmentation (OVIS data). Tab. 5 shows our results for the closed-world video instance segmentation task on the OVIS [20] dataset. In the closed-world setting, we disable the open-world object queries. We notice that the contrastive loss $\mathcal{L}_{\text{cont}}$ (discussed in Sec. 3.2) improves the closed-world results. We use a clip-length of $T = 2$ in this setting and CAROQ [18] to combine results from video clips.

Effect of clip-length T. Tab. 6 shows how length T of the video-clips affects performance, when CAROQ [18] is used to merge the video-clips. $T = 1$ defaults to frame-by-frame inference. We clearly observe that frame-by-frame inference is sub-optimal as compared to a larger temporal context. However, the performance improvement is marginal with $T > 2$. This suggests that the object queries already retain long-term temporal information and additional context features are no longer significant, as also highlighted in CAROQ [18].

Table 6: Ablation to show how the clip-length T affects the performance on the DenseVOC task.

T	CHOTA	CapA	DetA	AssA
1	49.2	40.5	58.8	50.0
2	**53.0**	**43.9**	**60.1**	54.0
4	52.3	43.6	59.6	**55.1**

Reproducibility. Sec. 4.2 discusses the main results on the BURST [3] and VidSTG [19] data. Results are calculated by repeating the respective experiments 3 times with different seeds. The variances in performance for OW-VISCapTor+DEVA and OW-VISCapTor+CAROQ are 0.4 on the unseen categories for the BURST [3] dataset and 0.5 on the captioning accuracy for the VidSTG dataset [19] respectively. This shows that the proposed method produces consistent results despite different seeds.

E Additional Qualitative Analysis

In this section, we provide additional qualitative analysis of the different components discussed in Sec. 3 of the main paper.

Open-world embeddings as spatially-rich object proposals. The object abstractor in our approach introduces spatially rich open-world embeddings e_{ow} (Sec. 3.2 of the main paper). These embeddings are modulated in the transformer decoder (Fig. 3 (a)) to generate open-world object queries. We obtain the open-world embeddings by encoding a grid of equally spaced points across the feature dimensions through a prompt encoder. This encourages object discovery throughout the video frame. The open-world embeddings act as initial abstract object proposals.

In Fig. 8, we show that the open-world embeddings are strong object proposals, even before they are modulated by the object transformer by being combined with video frame features. The person, the spoon, the plate, and some food on the plate are discovered by the open-world embeddings. Their segmentation masks are obtained by the dot product between the open-world embeddings and the video frame features. Further, we see a strong spatial correlation between the grid of points and the segmentation masks generated by the corresponding open-world embeddings. This suggests that encoding a grid of points using the prompt encoder makes our object abstractor spatially rich.

Masked cross-attention for object-centric captioning. In Tab. 3, we quantitatively show that masked cross-attention in the object-to-text abstractor (Sec. 3.3) helps in generating accurate object-centric captions for individual objects. Fig. 9 shows the high quality of the object-centric captions generated. The black colored caption (‘a family sitting on a couch with a child’) is obtained when no mask is provided in the object-to-text transformer. The entire image features are seen during the cross-attention operation in each layer of the object-to-text transformer. The caption fails to capture the object-centric details.

Other colored captions are generated with masked attention for the individual objects. The colored captions on the left highlight the effectiveness of masked attention in generating object-centric captions. For example, the three persons (highlighted in cyan, grayish blue, and green) have distinct captions, each one describing the individual identities of the corresponding person. The school bag (light blue) is also described correctly. We want to note that sometimes the method fails to generate meaningful object-centric captions for small objects (captions on the right). We discuss this more in Appendix G.

Contrastive loss to suppress overlapping predictions. In Tab. 2 of the main paper and in Tab. 5, we demonstrate the effectiveness of using the contrastive loss $\mathcal{L}_{\text{cont}}$ (discussed in Sec. 3.2) for object detection. This loss encourages that object queries differ from each other, among others by suppressing highly overlapping predictions. We highlight this in Fig. 10. The left image shows a frame from the OVIS [20] dataset. The top-right and bottom-right images show a few predictions from our network trained without (top) and trained with (bottom) the contrastive loss. The repetitive predictions for the top-right image are highlighted with red and cyan boxes. The contrastive loss helps in removing these repetitions.

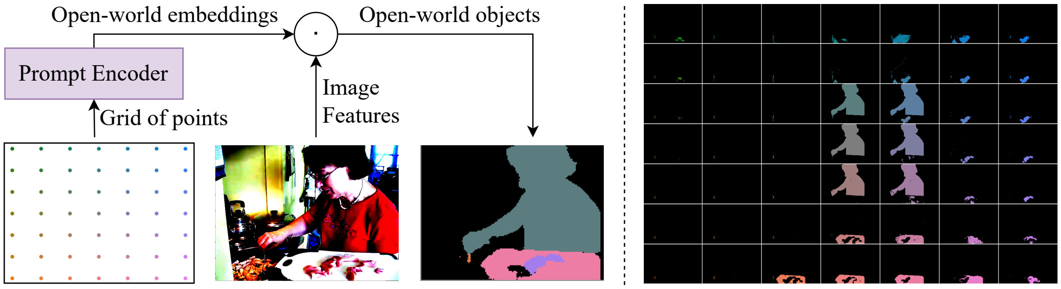


Figure 8: The prompt encoder in our proposed object abstractor creates spatially-rich open-world embeddings that act as strong object proposals. A strong spatial correlation exists between the grid of points and the segmentation masks generated by the corresponding open-world embeddings. This is highlighted by the color of the points in the grid and the color of the segmentation masks.

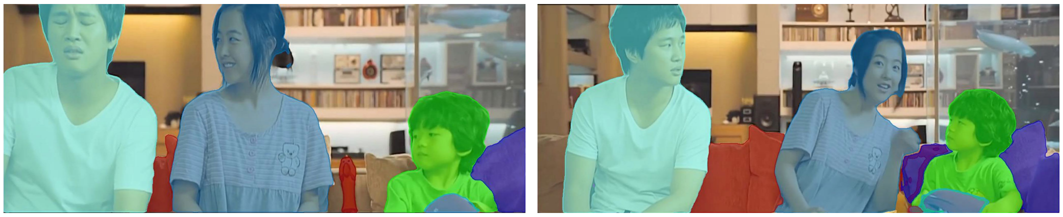

a family sitting on a couch with a child.
back of the school bag.
the mother is sitting on the couch.
a child is sitting on a couch.
a man is sitting on a couch with wife and child.
a family sits on the couch.
the the the the the the the

Figure 9: Masked cross-attention in the object-to-text abstractor helps generate object-centric captions (colored captions on the left). Providing no mask during cross attention fails to capture object-centric details (black caption). However, masked attention sometimes fails to generate meaningful object-centric captions for small objects (colored captions on the right).

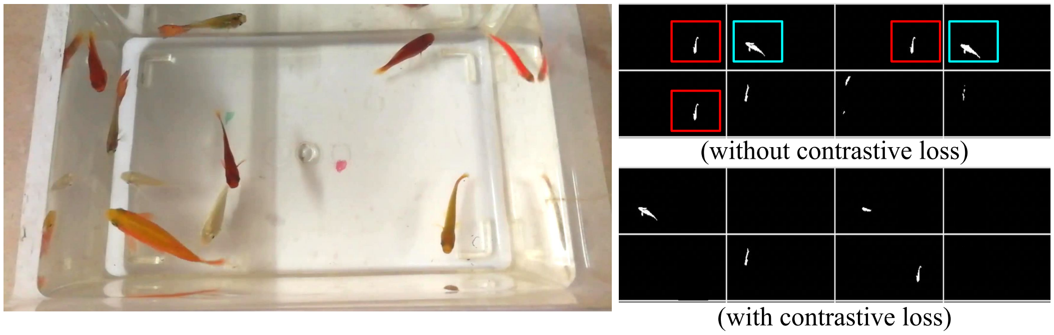
F Implementation Details

In this section, we provide the implementation details of OW-VISCapTor. We first describe the architecture of different components discussed in Sec. 3. We then discuss our choice of hyper-parameters for all experiments discussed in Sec. 4. We also discuss the resources and the licenses of the code bases and datasets used in the paper.

F.1 Architecture

Object abstractor. Our object abstractor consists of a prompt encoder, a transformer decoder, and closed-world embeddings e_{ow}. Our prompt encoder, discussed in Sec. 3.2, is lightweight and consists of 4 learned embeddings with 256 channels added to their positional encodings [62]. We initialize our prompt encoder from SAM [23] and fine-tune it on the BURST [3] dataset to generate open-world embeddings. The transformer decoder has 3 transformer layers, each layer consisting of a masked cross-attention, self-attention and FFN layer followed by a LayerNorm. The transformer layers and the closed-world embeddings are initialized from Mask2Former trained on COCO instance segmentation, and fine-tuned on the BURST [3] or VidSTG [19] datasets. An auxiliary loss is added to every intermediate transformer decoder layer and to the closed-world embeddings following Mask2Former [13] to improve the detection performance of OW-VISCapTor.

Object-to-text abstractor. Our object-to-text abstractor, detailed in Sec. 3.3, consists of 11 transformer layers and the text embeddings e_{text}, which are initialized from BLIP-2 [11]. Each transformer layer consists of a self-attention, and a FFN layer, followed by a LayerNorm. The masked cross-



(without contrastive loss)

(with contrastive loss)

Figure 10: The repetitive predictions without contrastive loss (top-right) are highlighted with red and cyan boxes. Contrastive loss (bottom-right) helps in suppressing these repetitions.

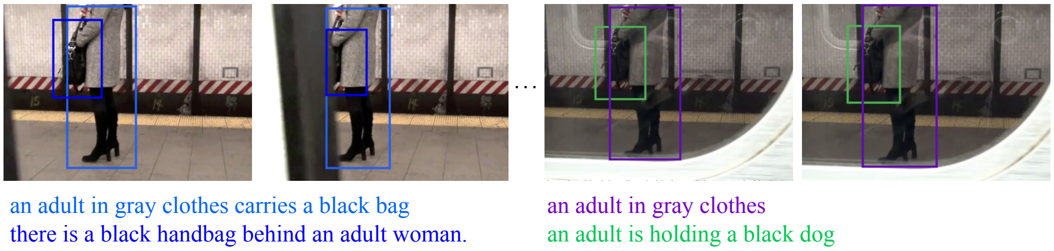

an adult in gray clothes carries a black bag
there is a black handbag behind an adult woman.
an adult in gray clothes
an adult is holding a black dog

Figure 11: A failure mode, where the object identities aren't retained after prolonged occlusion. After a train crosses the screen for a prolonged period ($\sim$ 30 frames), the person initially identified as light blue is later identified as purple. The bag initially identified as dark blue is later confused to be a dog identified as green.

attention is present in alternate layers after the self-attention, following BLIP-2 [11]. The object queries (with channel dim. 256) are first passed through a linear layer to match the channel dimensions of the text embeddings (with channel dim. 768). Each of these modified object queries is then concatenated with the text embeddings and modulated in the object-to-text transformer to generate object-centric text queries. We fine-tune the object-to-text abstractor, the linear layer, and the text embeddings on the VidSTG [19] dataset. The object-centric text queries are used by an LLM for object-centric captioning. We use a frozen OPT-2.7B model as the LLM, which is a decoder-only model having 2.7 billion parameters.

Feature extractor. The feature extractor for each video frame, consisting of a backbone and a pixel-decoder, takes the video frame as input and produces multi-level image features. This design follows the meta-architecture from Mask2Former [13]. The transformer decoder in the object abstractor modulates the open- and closed-world object queries using the multi-level image features. We use the ResNet-50 and the SwinL backbones for our experiments. The Deformable-DETR [63] is used as the pixel decoder, with 6 multi-scale deformable attention layers applied to feature maps to generate multi-scale image features of resolution $1/4$, $1/8$, $1/16$ and $1/32$ with 256 channels. These image features are used as input in the transformer decoder of the object transformer in a round robin fashion following Mask2Former [13]. Note the image features which act as inputs to the object-to-text abstractor are obtained from the vision encoder of BLIP-2 [11].

Detection head and classification head. The detection head consists of 2 linear layers separated by ReLU activation layers. It generates either bounding box predictions (VidSTG [19] dataset) or predictions that yield the segmentation masks (BURST [3] dataset) when a dot product is computed between them and the image-features. The classification head for the closed-world objects consists of a linear layer to generate the logits for each category.

F.2 Hyper-Parameters

We now discuss the hyper-parameters used in this work. For experiments on the BURST [3] dataset, we encode a grid of 7×7 points across the width and height of the image features to obtain the open-world embeddings e_{ow} discussed in Sec. 3.2. Hence the total number of open-world object queries $N_{\text{obj,ow}}$ is 49. We also experimented with a grid of 4×4 and 10×10 but didn't see a significant change in performance. For experiments on the VidSTG [19] dataset, the number of text embeddings e_{text} is 32. In all experiments, the maximum number of closed world objects ($N_{\text{obj,cw}}$) in a given video for a ResNet-50 backbone is 100, and for a Swin-L backbone is 200. We use a feature dimension C (Sec. 3.1) of 256 in all models, unless stated otherwise.

We trained the models with an initial learning rate of 0.0001 and ADAMW [64] optimizer with a weight decay of 0.05. We use a batch size of 8. The backbone, the pixel decoder, the closed-world embeddings and the transformer layers in the object abstractor were first initialized with weights from Mask2Former [13] trained on the COCO image instance segmentation dataset. The object-to-text abstractor and the text embeddings were first initialized with weights from BLIP-2 [11]. We then fine-tune the models on the respective BURST [3], VidSTG [19], and OVIS [20] datasets for $10,000$, $16,000$, and $8,000$ iterations respectively.

F.3 Resources

We used 8 NVIDIA A100 GPUs to run the experiments presented in this paper. Each experiment took roughly 10 GPU hours of training on the A100 GPUs for the BURST experiments, 16 GPU hours for the VidSTG experiments, and 8 GPU hours for the OVIS dataset experiments.

F.4 Licenses

Our code is built on Mask2Former [13] which is majorly licensed under the MIT license, with some portions under the Apache-2.0 License. We also build on SAM [23], which is released under the Apache 2.0 License, and BLIP-2 [11] which is released under the MIT license. The VidSTG [19] and BURST [3] datasets are released under the MIT license. The OVIS [20] dataset is released under the Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) License.

G Limitations

In this section, we show some failure modes of our proposed approach and discuss limitations. Fig. 8 shows that our approach sometimes fails to detect some open-world objects that a human may find to be of interest. For example, the grinder on the left, the window at the top-right, etc., are not detected by the network. The colored captions on the right side of Fig. 9 show that our approach sometimes fails to generate meaningful object-centric captions for small objects. For the purple object (cushion on a sofa), the caption ('the the the ...') is not meaningful since it fails to form a complete sentence or capture the identity of the object it represents. For the red object (other cushions on a sofa), the caption ('a family sits on the couch') is not object-centric since it fails to provide a description specific to the object. Fig. 11 further highlights a failure mode. After a train crosses the scene for a prolonged period (~ 30 frames), object identities may be lost.

These issues can be addressed by stronger strategies for open-world object discovery, stronger caption-generators, and by integrating better object trackers, which we leave for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our claims are consistent with the experimental results in Sec. 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This is mentioned in Appendix G.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sufficient details are provided in the paper and supplementary. Additionally, we include the codes in the supplementary zip file.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Codes are included in the supplementary zip file.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We mention this in Appendix F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: These are mentioned in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: These are mentioned in Appendix F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conform to the Code of Ethics to the best of our knowledge.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This is discussed in Sec. 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Authors are cited and licenses are mentioned and respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Codes with documentation are included in the supplementary zip file.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.