

CTVIS: Consistent Training for Online Video Instance Segmentation

Kaining Ying^{1,2*} Qing Zhong^{4*} Weian Mao⁴ Zhenhua Wang^{3†} Hao Chen^{1†}
Lin Yuanbo Wu⁵ Yifan Liu⁴ Chengxiang Fan¹ Yunzhi Zhuge⁴ Chunhua Shen¹

¹ Zhejiang University ² College of Computer Science and Technology, Zhejiang University of Technology

³ College of Information Engineering, Northwest A&F University

⁴ The University of Adelaide, Australia ⁵ Swansea University, UK

<https://github.com/KainingYing/CTVIS>

Abstract

The discrimination of instance embeddings plays a vital role in associating instances across time for online video instance segmentation (VIS). Instance embedding learning is directly supervised by the contrastive loss computed upon the **contrastive items (CIs)**, which are sets of anchor/positive/negative embeddings. Recent online VIS methods leverage CIs sourced from one reference frame only, which we argue is insufficient for learning highly discriminative embeddings. Intuitively, a possible strategy to enhance CIs is replicating the inference phase during training. To this end, we propose a simple yet effective training strategy, called **Consistent Training for Online VIS (CTVIS)**, which devotes to aligning the training and inference pipelines in terms of building CIs. Specifically, CTVIS constructs CIs by referring inference the momentum-averaged embedding and the memory bank storage mechanisms, and adding noise to the relevant embeddings. Such an extension allows a reliable comparison between embeddings of current instances and the stable representations of historical instances, thereby conferring an advantage in modeling VIS challenges such as occlusion, re-identification, and deformation. Empirically, CTVIS outstrips the SOTA VIS models by up to +5.0 points on three VIS benchmarks, including YTVIS19 (55.1% AP), YTVIS21 (50.1% AP) and OVIS (35.5% AP). Furthermore, we find that pseudo-videos transformed from images can train robust models surpassing fully-supervised ones.

1. Introduction

Video instance segmentation is a joint vision task involving classifying, segmenting, and tracking interested instances

*KY (email: kaining.ying.cv@gmail.com) and QZ contributed equally to this work. This work was done when KY, QZ, WM, YZ were visiting Zhejiang University.

†Corresponding authors.

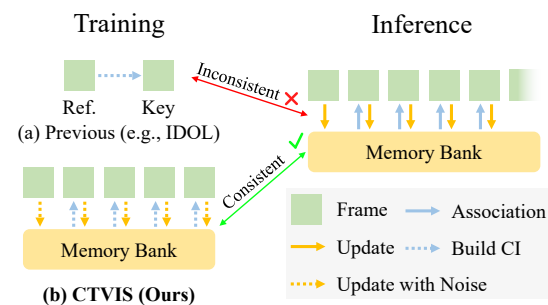


Figure 1. Comparison of inconsistent and consistent training (Ours). (a) Previous methods typically build contrastive items (CIs) and supervise the instance embeddings between key and reference frames. We call this paradigm inconsistent training, where the interaction with the long-term memory bank during training and the lack of modeling for long video in real inference scenarios is overlooked. (b) The purpose of **CTVIS** is to align the training and inference pipelines. Specifically, CTVIS constructs training stage CIs by leveraging the memory bank and incorporates noise during the memory bank updating to simulate real-world scenarios, such as ID switching, that can occur during inference.

across videos [27]. It is critical in many video-based applications, such as video surveillance, video editing, autonomous driving, augmented reality, *etc.* Current mainstream VIS methods [4, 11–13, 15, 24–28] can be categorized into offline and online groups. The former [4, 11, 13, 24, 25] segments and classifies all video frames simultaneously and makes the instance association in a single step. The latter [12, 26–28] takes as input a video in a frame-by-frame fashion, detecting and segmenting objects per frame while associating instances across time. In this paper, we focus on the online branch.

Online methods are typically built upon image-level instance segmentation models [5, 8, 22, 34, 35]. Several works [17, 21, 27] utilize convolution-based instance segmentation models to segment each frame and associate instances by incorporating heuristic clues, such as mask-overlapping ratios and the similarity of appearance. However, these hand-

designed approaches always fail to tackle complicated cases, which typically include severe target occlusion, deformation and re-identification. Recently, encouraged by the thriving of Transformer-based [23] architectures in object detection and segmentation [2, 5, 35], a bunch of query-based online frameworks have been proposed [12, 26], which take advantage of the temporal consistency of query embeddings and associate instances by linking corresponding query embeddings frame by frame. These advances boost the performance of online VIS models, which become de-facto leading VIS performance on most benchmarks (especially on challenging ones such as OVIS [21]).

Though the importance of the discrimination of query embeddings to associate instances has been nominated [12, 26], less research attention has been paid in this vein. MinVIS [12] simply trains a single-frame segmentor, and the quality of its query embedding is hampered by the segmentor originally proposed for image-based instance segmentation. As shown in Figure 1(a), recent methods [14, 26] merely supervise instance embedding generation between two temporally adjacent frames with their contrastive losses computed upon contrastive items. Specifically, for each instance at the key frame, if the same instance appears on the reference frame, the embedding of it is selected as the anchor embedding \mathbf{v} . Meanwhile, its embedding in the reference frame is taken as the positive embedding \mathbf{k}^+ , and the embeddings of other instances in the reference frame are used as the negative embeddings \mathbf{k}^- . In convention the set $\{\mathbf{v}, \mathbf{k}^+, \mathbf{k}^-\}$ is called **contrastive item** (CI). This training paradigm is *inconsistent* with the inference (shown in the right of Figure 1), as it overlooks the interaction with the long-term memory bank to construct contrastive items and lacks modelling for long videos. To bridge this gap, we propose CTVIS (as shown in Figure 1(b)), which intuitively brings in useful tactics from inference, including memory bank, momentum-averaged (MA) embedding and noise training. Specifically, CTVIS samples several frames from a long video to form one training sample. Then we process each sample frame by frame, which can produce abundant CIs. Moreover, we sample momentum-averaged (MA) embeddings from the memory bank to create positive and negative embeddings. Furthermore, we introduce noise training for VIS, incorporating a few noises into the memory bank updating procedure to simulate the tracking failure scenarios during the inference process.

We also consider the availability of large-scale training samples, which are especially expensive to annotate and maintain for VIS. To tackle this, we implement and test several goal-oriented augmentation methods (to align with the distribution of real data) to produce pseudo-videos. Different from the COCO joint training, we only use pseudo-videos to train VIS models.

Without bells and whistles, CTVIS outperforms the state-

of-the-art by large margins on all benchmark datasets, including YTVIS19 [27], YTVIS22 [27], and OVIS [21]. Even trained with pseudo-videos only, CTVIS surpasses fully supervised VIS models [11, 25, 26]. Here we summarize our key contributions as

- We propose a simple yet effective training framework (CTVIS) for online VIS. CTVIS promotes the discriminative ability of the instance embedding by interacting with long-term memory banks to build CIs, and by introducing noise into the memory bank updating procedure.
- We propose to create pseudo-VIS training samples by augmenting still images and their mask annotations. CTVIS models trained with pseudo-data only surpass their fully-supervised opponents already, which suggests that it is a desirable choice, especially when dense temporal mask annotations are limited.
- CTVIS achieves impressive performance on three public datasets. Meanwhile, extensive ablation validates the method’s effectiveness.

2. Related Work

Online VIS Method [12, 14, 16, 26–28, 32, 33] typically builds upon image-level instance segmentation models [2, 5, 8, 30, 35]. MaskTrack R-CNN [27] extends Mask R-CNN [8] by incorporating an additional tracking head, which associates instances across videos using heuristic cues. CrossVIS [28] proposes to guide the segmentation of the current frame by the features extracted from previous frames. With the emergence of query-based instance segmentors [2, 5, 35], matching with query embeddings instead of hand-designed rules boosts the performance of online VIS [12, 26]. Utilizing the temporal consistency of intra-frame instance queries predicted by the image-level segmentor [5, 35], MinVIS [12] tracks instances by Hungarian matching of the corresponding queries frame by frame without video-based training. IDOL [26] supervises the matching between instances that appeared within two adjacent frames during training. During inference, IDOL maintains a memory bank to store instance momentum averaged embeddings detected from previous frames, which are employed to match with newly detected foreground instance embeddings. Concurrent work GenVIS [10] applies a query-propagation framework to bridge the gap between training and inference in online or semi-online manners. Different from previous approaches, CTVIS aims to absorb ideas from the inference stage of online methods and learn more robust and discriminative instance embeddings during training.

Offline VIS Method [4, 11, 13, 24, 25] takes as input the entire video and predicts masks for all frames in a single run. VisTR [24] utilises clip-level instance features as input and predicts clip-level mask sequences in an end-to-end manner. Subsequently, several follow-up works, such

as Mask2Former-VIS [4], and SeqFormer [25], exploit attention [23] to process spatio-temporal features and directly predict instance mask sequences. To mitigate the memory consumption on extremely long videos, VITA [11] proposes to decode video object queries from sparse frame-level object tokens instead of dense spatio-temporal features.

Discriminative Instance-Level Feature Learning. The discrimination of instance embeddings plays a vital role in instance-level association tasks. Most works absorb the ideas from contrastive learning in self-supervised representation learning. IDOL [26] and QDTrack [6] supervise the learning of contrastive instance representations between two adjacent frames. SimCLR [3] argues that contrastive learning can benefit from larger batches. Inspired by this, CTVIS introduces long video training samples instead of key-reference image pairs, which leads to more robust instance embeddings.

VIS Model Training with Sparse Annotations. Annotating masks for each object instance in every frame and linking them across the video is prohibitively expensive. Furthermore, recent works [6, 12, 20] suggest that the dense video annotations for VIS are unnecessary. MinVIS [12] makes a per-frame image-level segmentation and associates the generated instance queries to obtain the video-level results. Since the training of the MinVIS model is agnostic to the temporal association of masks, it can benefit from the availability of large-scale datasets for image-level instance segmentation [18]. QDTrack [6] learns compelling instance similarity using pairs of transformed views of images. MS COCO [18], which contains abundant image-level mask annotations, is typically taken to supplement the training of models for VIS [11, 25, 26]. Following this, we propose to train VIS models with pseudo-videos generated by augmenting images instead of natural videos. We show that CTVIS models trained on pseudo-videos can surpass SOTA models [4, 11, 13, 25–27] trained with densely annotated videos by clear margins. Different from techniques taking augmentation to enrich the training set [2, 6, 7], we use augmentation to create the set, which contains pseudo-videos and the associated mask annotations (as well as their spatio-temporal tracks). Moreover, we carefully design the video generation routines based on classical augmentation techniques (i.e. *rotation*, *crop* and *copy&paste*), such that the pseudo-videos are realistic and can cover VIS challenges (including *object occlusion*, *fast-motion*, *re-identification* and *deformation*).

3. Methods

CTVIS builds upon Mask2Former [5], which is an effective image instance segmentation model (briefly reviewed in Section 3.1)¹. Our CTVIS is motivated by the inference of typical online VIS methods introduced in Section 3.2.

¹Note that CTVIS can be easily combined with other query-based instance segmentation models [2, 26, 35] with minor modifications.

Then we detail our consistent training method in Section 3.3. Finally, Section 3.4 presents our goal-oriented pseudo-video generation technique for training VIS models with sparse image-level annotations.

3.1. Brief Overview of Mask2Former

Mask2Former [5] composed of three main components: an *image encoder* \mathcal{E} (consist of a backbone and a pixel decoder), a *transformer decoder* \mathcal{T} and a *prediction head* \mathcal{P} . Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, \mathcal{E} extracts a set of feature maps $\mathbf{F} = \mathcal{E}(I)$, where $\mathbf{F} = \{F_0 \cdots F_{-1}\}$ is a sequence of multi-scale feature maps, and F_{-1} is the final output of the \mathcal{E} with 1/4 resolution of I . The N raw query embeddings $\hat{Q} \in \mathbb{R}^{N \times C}$ are learnable parameters, where N is a large enough number of outputs and C is the number of channels. Then, \mathcal{T} takes both \mathbf{F} and \hat{Q} to iteratively refine query embeddings, and consequently outputs $Q \in \mathbb{R}^{N \times C}$. Finally, the prediction head outputs the segmentation masks M and the classification scores O . For classification, $O = \mathcal{C}(Q) \in \mathbb{R}^{N \times K}$, where K is the number of object categories. For segmentation, the masks $M \in \mathbb{R}^{N \times H/4 \times W/4}$ are generated with $M = \sigma(Q * F_{-1})$, where $*$ denotes the convolution operation and $\sigma(\cdot)$ is the sigmoid function.

Our Modification. Because CTVIS employs instance embeddings to associate instances during inference, we add a head (a few MLP layers) to compute the instance embeddings $E \in \mathbb{R}^{N \times C}$ based Q .

3.2. Inference of CTVIS

CTVIS leverages Mask2Former [5] to process each frame and introduces an external memory bank [26, 27] to store the states of previously detected instances, including classification scores, segmentation masks and instance embeddings. To ease presentation, we assume that CTVIS has already processed T frames out of an input video of L frames, and there are N predicted instances with N instance embeddings $\mathbf{d}_i \in \mathbb{R}^C$ in the current frame. The memory bank stores for the previous T frames M detected instances, each of which has multiple temporal instance embeddings $\{e_j^t \in \mathbb{R}^C\}_{t=1}^T$ and a momentum-averaged instance embedding \hat{e}_j^T , which is computed according to the similarity-guided fusion [31]:

$$\hat{e}_j^T = (1 - \beta^T)\hat{e}_j^{T-1} + \beta^T e_j^T, \quad (1)$$

$$\beta^T = \max \left\{ 0, \frac{1}{T-1} \sum_{k=1}^{T-1} \Psi_d(e_j^T, e_j^{T-k}) \right\}, \quad (2)$$

where Ψ_d denotes the cosine similarity. We refer the reader to [31] for more details. Next, for each instance i detected in the current frame, we compute its bi-softmax similarity [6] with respect to the previously detected instance j using

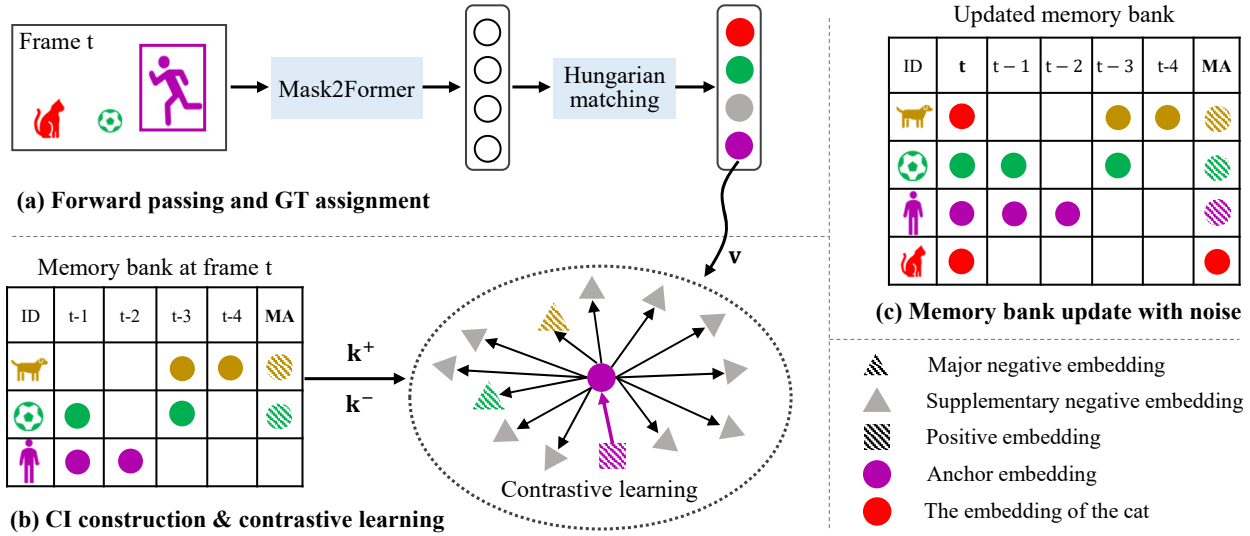


Figure 2. **Overview of the proposed CTVIS:** **a)** forward passing and GT assignment using Mask2Former and Hungarian matching; **b)** consistent training via building CIs with a memory bank. For simplicity, we only show the construction of CIs for the human instance (anchor) in the t -th frame of a training video. Through contrastive learning, positive embeddings are pulled close to the anchor embedding, while negative embeddings are pushed away from the anchor; **c)** Update the memory bank using the embeddings of frame t with noise.

$$f_{i,j} = 0.5 \cdot \left[\frac{\exp(\hat{\mathbf{e}}_j^T \cdot \mathbf{d}_i)}{\sum_k \exp(\hat{\mathbf{e}}_k^T \cdot \mathbf{d}_i)} + \frac{\exp(\hat{\mathbf{e}}_j^T \cdot \mathbf{d}_i)}{\sum_l \exp(\hat{\mathbf{e}}_l^T \cdot \mathbf{d}_i)} \right] \quad (3)$$

Finally, we find the “best” instance ID for i with

$$\hat{j} = \arg \max f_{i,j}, \forall j \in \{1, 2, \dots, M\}. \quad (4)$$

If $f_{i,\hat{j}} > 0.5$, we believe that newly detected instance i and instance \hat{j} in the memory bank correspond to the identical target. Otherwise, we initiate a new instance ID in the memory bank. When all frames are processed, the memory bank contains a certain number of instances, each of which takes a classification score list $\{c_i^t\}_{t=1}^L$ and a mask list $\{m_i^t\}_{t=1}^L$ (recall that L denotes the number of frames). For each instance i , we calculate its video-level classification score by averaging the frame-level scores of the object.

3.3. Consistent Learning

A reliable matching of instances (*i.e.* using Equation (3)) across time is required to track instances successfully. Hence the extraction of highly discriminative embeddings of objects is of great importance. We argue that the discrimination of instance embeddings extracted with recent models [14, 26] is still inadequate, especially for videos involving object-occlusion, shape-transformation and fast-motion. One main reason is that mainstream contrastive learning methods build CIs (*i.e.* $\{\mathbf{v}, \mathbf{k}^+, \mathbf{k}^-\}$) from the reference frame only, which results in the comparison of the anchor embedding against instantaneous instance embeddings in \mathbf{k}^+ and \mathbf{k}^- . Such embeddings are typically less discriminative and contain noise,

which prevents training from learning robust representations. To address this, our CTVIS leverages a memory bank to store MA embeddings, thus supporting contrastive learning from more stable representations. Here our insight is to align the embedding comparison of training with that of inference (such that the two comparisons are consistent). Figure 2 sketches our CTVIS, which processes the training video frame-by-frame. For an arbitrary frame t , CTVIS involves three steps: **a)** it first takes the Mask2Former and Hungarian matching to compute the instance embeddings, and to match them with GT (highlighted by red, green and purple); **b)** Then, it builds CIs using MA embeddings within the memory bank, and performs contrastive learning with CIs; and **c)** It updates the memory bank with noise (*e.g.* the embedding of the *cat* is deliberately added to the memory of the *dog*), which serves the learning from the next frame.

Forward passing and GT assignment. As shown in Figure 2 (a), we first feed the current frame t into Mask2Former to compute the embeddings for queries. Then we employ Hungarian matching to find an optimal match between the decoded instances and the ground truth (GT), such that each GT instance is assigned one instance embedding. Note that Hungarian matching relies on the costs calculated for all (*Decoded-Instance*, *GT-Instance*) pairs. Essentially, each cost measures the similarity between a pair of instances based on their labels and masks.

Construct CIs. After GT assignment, we build the contrastive items for each GT instance using a memory bank. The memory bank stores all detected instances of previous $t - 1$ frames, each associated with 1) a series of instance embeddings extracted at different times, and 2) its MA em-

bedding computed by Equation (1). In order to prepare the CIs $\{\mathbf{v}, \mathbf{k}^+, \mathbf{k}^-\}$ for instance i (termed as the *anchor*, e.g. the person in Figure 2 (a)) at the t -th frame, the instance embedding extracted from this frame is used as the anchor embedding v . For the positive embedding, we pick from the memory bank the MA embedding of instance i . The negative embeddings \mathbf{k}^- include the major negative embeddings and the supplementary negative embeddings. We use the MA embeddings of other instances in the memory bank as the major negative embeddings. We also sample the background query embeddings of previous $t - 1$ frames to form the supplement negative embeddings. Taking as inputs the created CIs, we compute the contrastive loss with

$$\begin{aligned} \mathcal{L}_{\text{emb}} &= -\log \frac{\exp(\mathbf{v} \cdot \mathbf{k}^+)}{\exp(\mathbf{v} \cdot \mathbf{k}^+) + \sum_{\mathbf{k}^-} \exp(\mathbf{v} \cdot \mathbf{k}^-)} \quad (5) \\ &= \log \left[1 + \sum_{\mathbf{k}^-} \exp(\mathbf{v} \cdot \mathbf{k}^- - \mathbf{v} \cdot \mathbf{k}^+) \right]. \end{aligned}$$

As shown in Figure 2 (c), training with \mathcal{L}_{emb} pulls the embeddings of positive instances close to the anchor embedding, while pushing the negative embeddings away from it.

Update memory bank. After computing the \mathcal{L}_{emb} for each instance in frame t , we need to update the memory bank, such that the updated version can be taken to build CIs for frame $t + 1$. Unlike the inference stage, for training we can get the ground truth ID of each instance so as to update the memory bank with their embeddings extracted from frame t . In comparison, inference can fail to track instances across time (i.e. the ID switch issue), especially for complicated scenarios. To alleviate this, we introduce noise to the update of the memory bank, which compels the contrastive learning to tackle the switch of instance IDs. Specifically, each disappeared instance (e.g. the dog) in frame t will have a little chance to receive an embedding of other instances (e.g. the cat, which is randomly picked from all available instances) in the same frame, which is called the *noise*. If the generated random value exceeds a threshold, as illustrated in Figure 2 (c), we use the noise as the embedding of the disappeared instance at frame t . Finally, the MA embeddings are updated for all instances using Equation (1). Due to the low similarity between the disappeared instance and the noise, such an update has quite a limited impact on the MA embedding of the instance, which is reidentified later. Indeed, training with noise is able to reduce the chance of ID switch, as demonstrated by the fish example in Figure 5.

Loss. After processing all frames, The \mathcal{L}_{emb} values of all CIs are averaged to obtain L_{emb} . The total training loss is

$$L_{\text{total}} = \lambda_{\text{emb}} L_{\text{emb}} + \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{ce}} L_{\text{ce}} + \lambda_{\text{dice}} L_{\text{dice}}, \quad (6)$$

where λ denotes loss weight. L_{cls} , L_{ce} and L_{dice} supervise the per-frame segmentation as suggested in [5].

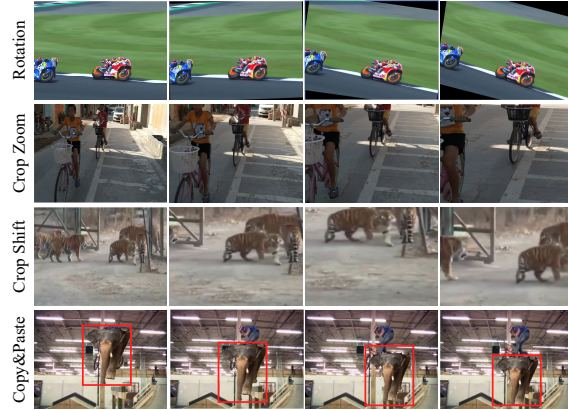


Figure 3. Generating pseudo-videos by augmenting images.

3.4. Learning from Sparse Annotation

We now elaborate on our pseudo-video and mask generation technique, which enables the training of VIS models when only sparse annotations (e.g. image data) are available. We take a few widely applied image-augmentation methods, including *random rotation*, *random crop* and *copy&paste* on source image to create pseudo-videos and the associated instance masks. Note that the pseudo-videos are created by no means to approximate real ones. Instead, they are taken to mimic the movement of targets in reality.

Rotation. As shown in the first row of Figure 3, the rotation augmentation rotates the source images with several random angles (e.g., $[-15, 15]$) to introduce subtle changes between frames of the pseudo-videos.

Crop. The rotation augmentation cannot alter the shapes and magnitudes of instances. However, instances deform or/and enter/exit the visible field due to the movement introduced either by the target or the camera. To address this, we apply random crop augmentation to the image, which allows the generated videos to mimic the zooming in/out effect of the camera lens and the shifting of targets. The second and the third rows of Figure 3 present two examples of *crop-zoom* and *crop-shift*, respectively. The pseudo-videos generated by such augmentations cover a large proportion of targets' movements.

Copy and Paste. As mentioned earlier, the trajectories of instances in pseudo-videos created by the augmentations share the identical motion direction. To incorporate the relative motion between instances, we also employ the *copy&paste* augmentation [7], which copies the instances from another image in the dataset and pastes them into random locations within the source image. Note that the pasting positions of an instance are typically different across time, which brings the relative motion between different instances (as shown in the fourth row of Figure 3).

Methods	Params.	YTVIS19 [27]					YTVIS21 [27]					OVIS [21]					
		AP	AP ₅₀	AP ₇₅	AR ₁	AR ₁₀	AP	AP ₅₀	AP ₇₅	AR ₁	AR ₁₀	AP	AP ₅₀	AP ₇₅	AR ₁	AR ₁₀	
ResNet-50 [9]	MaskTrack R-CNN [27]	-	30.3	51.1	32.6	31	35.5	28.6	48.9	29.6	26.5	33.8	10.8	25.3	8.5	7.9	14.9
	SipMask [1]	-	33.7	54.1	35.8	35.4	40.1	31.7	52.5	34	30.8	37.8	10.2	24.7	7.8	7.9	15.8
	CrossVIS [28]	-	36.3	56.38	38.9	35.6	40.7	34.2	54.4	37.9	30.4	38.2	14.9	32.7	12.1	10.3	19.8
	IFC [13]	-	41.2	65.1	44.6	42.3	49.6	35.2	55.9	37.7	32.6	42.9	13.1	27.8	11.6	9.4	23.9
	Mask2Former-VIS [4]	44	46.4	68	50	-	-	40.6	60.9	41.8	-	-	17.3	37.3	15.1	10.5	23.5
	TeViT [29]	-	46.6	71.3	51.6	44.9	54.3	37.9	61.2	42.1	35.1	44.6	17.4	34.9	15	11.2	21.8
	SeqFormer [25]	48	47.4	69.8	51.8	45.5	54.8	40.5	62.4	43.7	36.1	48.1	15.1	31.9	13.8	10.4	27.1
	MinVIS [12]	44	47.4	69	52.1	45.7	55.7	44.2	66	48.1	39.2	51.7	25	45.5	24	13.9	29.7
	IDOL [26]	43	49.5	<u>74</u>	52.9	47.7	58.7	43.9	<u>68</u>	<u>49.6</u>	38	50.9	<u>30.2</u>	<u>51.3</u>	<u>30</u>	<u>15</u>	<u>37.5</u>
	VITA [11]	57	49.8	72.6	<u>54.5</u>	<u>49.4</u>	61	45.7	67.4	49.5	<u>40.9</u>	<u>53.6</u>	19.6	41.2	17.4	11.7	26
CTVIS (Ours)	<u>44</u>	55.1	78.2	59.1	51.9	63.2	50.1	73.7	54.7	41.8	59.5	35.5	60.8	34.9	16.1	41.9	
Swin-L [19]	SeqFormer [25]	219	59.3	82.1	66.4	51.7	64.6	51.8	74.6	58.2	42.8	58.1	-	-	-	-	-
	Mask2Former-VIS [4]	216	60.4	84.4	67	-	-	52.6	76.4	57.2	-	-	25.8	46.5	24.4	13.7	32.2
	MinVIS [12]	216	61.6	83.3	68.6	54.8	66.6	55.3	76.6	62	45.9	60.8	39.4	61.5	41.3	<u>18.1</u>	43.3
	VITA [11]	229	63	86.9	67.9	<u>56.3</u>	68.1	<u>57.5</u>	80.6	61	<u>47.7</u>	<u>62.6</u>	27.7	51.9	24.9	14.9	33
	IDOL [26]	213	<u>64.3</u>	<u>87.5</u>	<u>71</u>	<u>55.5</u>	<u>69.1</u>	56.1	<u>80.8</u>	<u>63.5</u>	45	60.1	<u>42.6</u>	<u>65.7</u>	<u>45.2</u>	17.9	<u>49.6</u>
	CTVIS (Ours)	<u>216</u>	65.6	87.7	72.2	56.5	70.4	61.2	84	68.8	48	65.8	46.9	71.5	47.5	19.1	52.1

Table 1. Compare CTVIS with SOTA methods. The best and second best are highlighted by **bold** and underlined numbers, respectively.

4. Experiment

Datasets. The proposed methods are evaluated on three VIS benchmarks: YTVIS19 [27], YTVIS21 [27] and OVIS [21]. **Metrics.** Following prior studies [4, 11–13, 25–28], we use Average Precision (AP) and Average Recall (AR) as the evaluation metrics.

Implementation Details. For the hyper-parameters of Mask2Former [5], we just use its officially released version. The number of layers of the instance embedding head is 3. All models are initialized with parameters pre-trained on COCO [18], and then they are trained on 8 NVIDIA A100 GPUs. Following prior works [10, 11, 25], we use the COCO joint training (CJT) setting to train our models unless otherwise specified. We set the lengths of training videos as 8 and 10 for YTVIS19&21 and OVIS, respectively. For data augmentation, we use clip-level random crop and flip. During the training phase, we resize the input frames so that the shortest side is at least 320 and at most 640p, while the longest side is at most 768p. During inference, the input frames are downsampled to 480p. We set λ_{emb} , λ_{cls} , λ_{ce} , λ_{dice} as 2.0, 2.0, 5.0 and 5.0, respectively. The mini-batch size is 16 and the maximum training iterations is 16,000. The initial learning rate is 0.0001 and decays at 6,000 and 12,000 iterations, respectively.

4.1. Main Result

As shown in Table 1, we compare CTVIS against SOTA methods [1, 4, 11–13, 25–29], respectively using ResNet-50 [9] and Swin-L [19] as the backbone on three benchmarks. **YTVIS19 & YTVIS21.** consist of relatively simple videos with short durations. Thanks to the introduced consistent learning paradigm and the extracted discriminative embeddings, CTVIS outperforms recent best methods on AP by $\sim 5\%$ with ResNet-50 on both benchmarks. With the stronger backbone Swin-L, CTVIS surpasses the second best by 3.7% on YTVIS21. Compared with IDOL [26], CTVIS

Methods	Deformable DETR* [26]		Mask2Former [5]	
	AP ^{YV19}	AP ^{OVIS}	AP ^{YV19}	AP ^{OVIS}
IDOL [26]	49.5	30.2	51.2	31.7
CTVIS	53.7 (+4.2)	33.8 (+3.6)	55.1 (+3.9)	35.5 (+3.8)

Table 2. Comparison of different instance segmentation methods with IDOL and CTVIS, respectively. Deformable DETR* is extended to instance segmentation as suggested in [26].

considerably improves the performance in terms of all metrics with tolerable parameter overheads.

OVIS. This dataset contains longer videos and more intricate contents, on which online methods [12, 26] perform much better than offline models [5, 11, 25]. Thanks to the effective embedding learning with long video samples, CTVIS gains 5.3 and 4.3 points in terms of AP, taking as inputs ResNet-50 and Swin-L, respectively. To summarize, CTVIS is highly competitive on benchmarks with varying complexities.

4.2. Ablation Study

We conduct extensive ablation to verify the effectiveness of CTVIS. Unless specified otherwise, we take the ResNet-50 as the backbone and train models under the CJT setting. Here we report AP^{YV19} and AP^{OVIS} on YTVIS19 and OVIS.

Do improvements mainly come from better image-level instance segmentation models? The answer is no. We validate this in Table 2: 1) Compared with IDOL with Deformable DETR, IDOL with Mask2Former is 1.0 and 1.5 points higher, suggesting the influence of a better detector is not that significant; 2) Since our CTVIS is not restricted to a specific network, we implement Deformable DETR with CTVIS, which brings 4.2 and 3.6 points of AP gains. Similarly, CTVIS on Mask2Former also boosts the results by 3.9 and 3.8 points, which indicates that the improvements mainly come from our proposed CTVIS.

Long-video training. To verify the effectiveness of long-video training, we ablate the number of frames of each

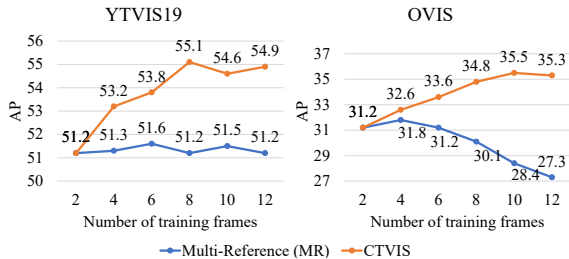


Figure 4. Ablation on the number of training frames. Multi-Reference extends IDOL by using multiple reference frames and the stronger Mask2Former as the segmentor.

video used for training. For a fair comparison, we extend IDOL [26] to a multiple references (MR) version, by replacing its segmentor with the stronger Mssk2Former and using multiple reference frames. Figure 4 shows the results. Thanks to the CI construction method employed by CTVIS, the performance has seen a dynamic increase by using more frames (peaked at 8 and 10 frames). In comparison, MR cannot benefit from long-video training and even degrades on OVIS. Hence we conclude that the performance of CTVIS stems from the effective video-level embedding learning (for tracking), rather than training an enhanced instance segmentor with larger batch sizes (more images per batch).

Components of CTVIS. First, removing all components of CTVIS sets a baseline, which utilizes a single reference to learn embeddings in a frame-by-frame way. As shown in Table 3, the baseline gets 51.6 and 32.6 on YTVIS19 and OVIS. Based on this baseline, we gradually add CTVIS components: 1) We take the latest embedding of each instance to build CIs (instead of MA embeddings), which improves AP^{YV19} and AP^{OVIS} to 52.1 and 33.3. This suggests that the sampling domain CIs do indeed influence the instance embedding learning; 2) When MA is incorporated, the results see salient increases (52.1 vs. 54.2 and 33.3 vs. 34.9), which indicates that our CI-building method renders the embedding learning more stable and consistent; 3) When incorporating noise in the memory bank, which is designed to alleviate the ID switch issue, the performance sees non-trivial increases (0.9 and 0.6 on two datasets). Put all components together, CTVIS obtains remarkable results on both datasets and outperforms the strong baseline by 3.5 and 2.9 points, which validates the significance of the temporal alignment between training and inference pipelines, at least for VIS.

Sampling of k^- . We test different ways of building the negative embeddings k^- . Table 4 presents four configurations and the corresponding results. Recall that the supplementary negative embeddings represent the background, and training with such negative samples only corrupts the performance (the 1st row). On the other hand, using major negative samples only gives decent results. A conjunctive usage of both negative-sampling types improves the performance significantly. In this line, we further consider sampling supplement-

Memory bank	Momentum	Noise	AP^{YV19}	AP^{OVIS}
			51.6	32.6
✓			52.1	33.3
✓	✓		54.2	34.9
✓		✓	55.1	35.5

Table 3. Effectiveness of different CTVIS components.

Major	Supplementary	AP^{YV19}	AP^{OVIS}
	✓	16.5	0.5
✓		50.8	31.6
✓	global	54.6	33.4
✓	local	55.1	35.5

Table 4. Ablate the sampling strategy of negative embeddings.

tary negative instances from either the local (sampled from the preceding frame only) or global domain (sampled from all previous frames). We found that the local setting gives the best results. This is probably because the model only needs to check the background in the local domain during inference. Hereafter we simply use the local setting.

4.3. Pseudo Video as Training Example

We train VIS models on pseudo-videos, which are created with COCO images and the method described in Section 3.4. Since COCO classes do not match that of VIS datasets, we only adopt the overlapping categories for training. For evaluation, we sample 421 and 140 videos with overlapping categories from the train sets of YTVIS21 and OVIS train sets, respectively. For more dataset information, please refer to the supplementary material. Specially, we denote the sampled version of YTVIS21 and OVIS as YTVIS21* and OVIS*. We use Swin-L as the backbone, and investigate the impacts of augmentation techniques in terms of generating pseudo-video datasets for training. Here *rotation* is taken as the baseline. As shown in Table 5, both *crop* and *copy&paste* bring gains on both datasets over the baseline. Because YTVIS21 is relatively simple, *crop* and *copy&paste* only improve the results by 0.2 and 0.5, respectively. However, for the complicated OVIS, they offer much larger performance gains, *i.e.* 1.3 and 2.0 on two datasets, which suggests that pseudo videos generated with stronger augmentations are especially suitable to tackle complicated VIS tasks. We also train VITA and IDOL models using the generated pseudo-samples. Again, CTVIS surpasses them by clear margins, as that demonstrated in Table 6.

4.4. Training with Limited Supervision

Following MinVIS [12], we train CTVIS and MinVIS models on only a proportion (%) of VIS training set. Specifically, we sample 1%, 5%, 10%, and 100% frames respectively from the training set to create pseudo videos for training. As shown in Table 7, with a 5% proportion, CTVIS outperforms MinVIS with 100% samples on all datasets. More



Figure 5. Visualize VIS results obtained by VITA [11], IDOL [26] and CTVIS. These examples show performance under heavy occlusion (the left example), sudden lighting-condition change (the right example) and disturbance of targets of the same category (the right example). Here, red boxes highlight inferior segmentations, and yellow ones mark incorrect IDs.

Rotation	Crop	Copy&Paste	AP ^{YV21*}	AP ^{OVIS*}
✓			48.5	27.3
✓	✓		48.7	28.6
✓		✓	49	29.3
✓	✓	✓	49.7	30.5

Table 5. Influence of augmentations on producing pseudo-videos.

Methods	Supervision	AP ^{YV21*}	AP ^{OVIS*}
MinVIS [12]	Image	43.9	24.4
VITA [11]	Pseudo video	44.4	19.1
IDOL [26]	Pseudo image pair	47.8	27.8
CTVIS	Pseudo video	49.7	30.5

Table 6. Compare with SOTA models trained with pseudo-samples, which are generated based on COCO images.

Methods	Training	AP ^{YV19}	AP ^{YV21}	AP ^{OVIS}
VITA [11]	Full	63	57.5	27.7
IDOL [26]		64.3	56.1	42.6
CTVIS (Ours)		65.6	61.2	46.9
MinVIS [12]	1%	59	52.9	31.7
	5%	59.3	54.3	35.7
	10%	61	54.9	37.2
	100%	61.6	55.3	39.4
CTVIS (Ours)	1%	62.4	57.8	36.2
	5%	63.4	59.4	41.9
	10%	64.2	60.0	42.1
	100%	64.8	60.7	44.1

Table 7. Compare with SOTA models trained with either the entire or a part ($x\%$) of training examples. Full means training with annotated videos.

importantly, CTVIS trained with pseudo videos, which are created from 100% frame samples, even surpasses the fully supervised competitors, and achieves close performance compared with CTVIS learned from genuine videos.

4.5. Qualitative Results

We visualize some VIS results obtained by SOTA offline [11] and online [26] approaches in Figure 5. The left

example includes heavy occlusion caused by moving pedestrian, the swap of instance positions, and target-disappearing-reappearing. Under such case, VITA [11] fails to segment and track the pedestrian. IDOL [26] mistakenly assigns the ID of the dog in the two rightmost images, and the squatting person is recognized as a dog. In comparison, our proposed CTVIS is able to segment, classify and track all instances successfully. For the right example, both VITA and IDOL fail to track the fish, and their ID switched after the video suddenly darkened. CTVIS also undergoes and ID switch (the middle image). Thanks to the noise introduced during training, CTVIS is more robust to tackle such occasional failure, and it reidentifies the fish later (the rightmost image).

5. Conclusion

We have presented CTVIS, a simple yet effective training strategy for VIS. CTVIS aligns the training and inference pipelines in terms of constructing contrastive items. Its ingredients include long-video training, memory bank, MA embedding and noise to facilitate the learning of better instance representations, which in turn offers more stable tracking of instances. Thanks to this design, CTVIS has demonstrated superior performance on multiple benchmarks. Additionally, to relieve the cost of the video-level annotation of masks, we propose to create pseudo videos for VIS training based on goal-oriented data augmentation. CTVIS models trained with pseudo videos, which are produced using only 10% frames extracted from the genuine training videos, achieve comparable performance, compared with SOTA models trained with full supervision.

Acknowledgement: This work was supported by National Key R&D Program of China (No. 2022ZD0118700), National Natural Science Foundation of China (No. 62272395), Zhejiang Provincial Natural Science Foundation of China (No. LY21F020024), and Qin Chuangyuan Innovation and Entrepreneurship Talent Project (No. QCYRCXM-2022-359).

References

- [1] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. SipMask: Spatial Information Preservation for Fast Image and Video Instance Segmentation. *Eur. Conf. Comput. Vis.*, 2020. [6](#)
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *Eur. Conf. Comput. Vis.*, pages 213–229. Springer, 2020. [2](#), [3](#)
- [3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Int. Conf. Mach. Learn.*, pages 1597–1607. PMLR, 2020. [3](#)
- [4] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2Former for Video Instance Segmentation. *arXiv preprint arXiv:2112.10764*, 2021. [1](#), [2](#), [3](#), [6](#)
- [5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-Attention Mask Transformer for Universal Image Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1290–1299, 2022. [1](#), [2](#), [3](#), [5](#), [6](#)
- [6] Tobias Fischer, Jiangmiao Pang, Thomas E Huang, Linlu Qiu, Haofeng Chen, Trevor Darrell, and Fisher Yu. QDTrack: Quasi-Dense Similarity Learning for Appearance-Only Multiple Object Tracking. *arXiv preprint arXiv:2210.06984*, 2022. [3](#)
- [7] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple Copy-Paste Is a Strong Data Augmentation Method for Instance Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2918–2928, 2021. [3](#), [5](#)
- [8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE Int. Conf. Comput. Vis.*, pages 2961–2969, 2017. [1](#), [2](#)
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. [6](#)
- [10] Miran Heo, Sukjun Hwang, Jeongseok Hyun, Hanjung Kim, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. A Generalized Framework for Video Instance Segmentation. *arXiv preprint arXiv:2211.08834*, 2022. [2](#), [6](#)
- [11] Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. VITA: Video Instance Segmentation via Object Token Association. In *Adv. Neural Inform. Process. Syst.*, 2022. [1](#), [2](#), [3](#), [6](#), [8](#)
- [12] Huang, De-An and Yu, Zhiding and Anandkumar, Anima. MinVIS: A Minimal Video Instance Segmentation Framework without Video-based Training. In *Adv. Neural Inform. Process. Syst.*, 2022. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#)
- [13] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. Video Instance Segmentation using Inter-Frame Communication Transformers. *Adv. Neural Inform. Process. Syst.*, 34:13352–13363, 2021. [1](#), [2](#), [3](#), [6](#)
- [14] Zhengkai Jiang, Zhangxuan Gu, Jinlong Peng, Hang Zhou, Liang Liu, Yabiao Wang, Ying Tai, Chengjie Wang, and Liqing Zhang. STC: Spatio-Temporal Contrastive Learning for Video Instance Segmentation. In *Eur. Conf. Comput. Vis. Worksh.*, pages 539–556. Springer, 2023. [2](#), [4](#)
- [15] Xiangtai Li, Henghui Ding, Wenwei Zhang, Haobo Yuan, Guangliang Cheng, Pang Jiangmiao, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-Based Visual Segmentation: A Survey. *arXiv pre-print*, 2023. [1](#)
- [16] Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. Video K-Net: A Simple, Strong, and Unified Baseline for Video Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18847–18857, June 2022. [2](#)
- [17] Zhuang Li, Leilei Cao, and Hongbin Wang. Limited Sampling Reference Frame for MaskTrack R-CNN. In *IEEE Int. Conf. Comput. Vis. Worksh.*, pages 3854–3857, 2021. [1](#)
- [18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Eur. Conf. Comput. Vis.*, pages 740–755. Springer, 2014. [3](#), [6](#)
- [19] 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 *IEEE Int. Conf. Comput. Vis.*, pages 10012–10022, 2021. [6](#)
- [20] Thuy C Nguyen, Tuan N Tang, Nam LH Phan, Chuong H Nguyen, Masayuki Yamazaki, and Masao Yamanaka. 1st Place Solution for YouTubeVOS Challenge 2021: Video Instance Segmentation. *arXiv preprint arXiv:2106.06649*, 2021. [3](#)
- [21] 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: A Benchmark. *Int. J. Comput. Vis.*, 130(8):2022–2039, 2022. [1](#), [2](#), [6](#)
- [22] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. In *IEEE Int. Conf. Comput. Vis.*, pages 9627–9636, 2019. [1](#)
- [23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. *Adv. Neural Inform. Process. Syst.*, 30, 2017. [2](#), [3](#)
- [24] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-End Video Instance Segmentation With Transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8741–8750, 2021. [1](#), [2](#)
- [25] Junfeng Wu, Yi Jiang, Song Bai, Wenqing Zhang, and Xiang Bai. SeqFormer: Sequential Transformer for Video Instance Segmentation. In *Eur. Conf. Comput. Vis.*, pages 553–569. Springer, 2022. [1](#), [2](#), [3](#), [6](#)
- [26] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In Defense of Online Models for Video Instance Segmentation. In *Eur. Conf. Comput. Vis.*, pages 588–605. Springer, 2022. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)
- [27] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *IEEE Int. Conf. Comput. Vis.*, pages 5188–5197, 2019. [1](#), [2](#), [3](#), [6](#)

- [28] 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 *IEEE Int. Conf. Comput. Vis.*, pages 8043–8052, 2021. [1](#), [2](#), [6](#)
- [29] Shusheng Yang, Xinggang Wang, Yu Li, Yuxin Fang, Jiemin Fang, Wenyu Liu, Xun Zhao, and Ying Shan. Temporally Efficient Vision Transformer for Video Instance Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2885–2895, 2022. [6](#)
- [30] Kaining Ying, Zhenhua Wang, Cong Bai, and Pengfei Zhou. ISDA: Position-Aware Instance Segmentation with Deformable Attention. In *Int. Conf. Acoustics, Speech, & Signal Process.*, pages 2619–2623. IEEE, 2022. [2](#)
- [31] En Yu, Zhuoling Li, and Shoudong Han. Towards Discriminative Representation: Multi-View Trajectory Contrastive Learning for Online Multi-Object Tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8834–8843, 2022. [3](#)
- [32] Tao Zhang, Xingye Tian, Haoran Wei, Yu Wu, Shunping Ji, Xuebo Wang, Yuan Zhang, and Pengfei Wan. 1st Place Solution for PVUW Challenge 2023: Video Panoptic Segmentation. *arXiv preprint arXiv:2306.04091*, 2023. [2](#)
- [33] Tao Zhang, Xingye Tian, Yu Wu, Shunping Ji, Xuebo Wang, Yuan Zhang, and Pengfei Wan. DVIS: Decoupled Video Instance Segmentation Framework. *IEEE Int. Conf. Comput. Vis.*, 2023. [2](#)
- [34] Tao Zhang, Shiqing Wei, and Shunping Ji. E2ec: An end-to-end contour-based method for high-quality high-speed instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4443–4452, 2022. [1](#)
- [35] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *Int. Conf. Learn. Represent.*, 2020. [1](#), [2](#), [3](#)