DVIS: Decoupled Video Instance Segmentation Framework

Tao Zhang[1] Xingye Tian[2] Yu Wu[1] Shunping Ji[1*]
Xuebo Wang[2] Yuan Zhang[2] Pengfei Wan[2]

[1]Wuhan University [2]Y-tech, Kuaishou Technology

Abstract

*Video instance segmentation (VIS) is a critical task with diverse applications, including autonomous driving and video editing. Existing methods often underperform on complex and long videos in real world, primarily due to two factors. Firstly, offline methods are limited by the tightly-coupled modeling paradigm, which treats all frames equally and disregards the interdependencies between adjacent frames. Consequently, this leads to the introduction of excessive noise during long-term temporal alignment. Secondly, online methods suffer from inadequate utilization of temporal information. To tackle these challenges, we propose a decoupling strategy for VIS by dividing it into three independent sub-tasks: segmentation, tracking, and refinement. The efficacy of the decoupling strategy relies on two crucial elements: 1) attaining precise long-term alignment outcomes via frame-by-frame association during tracking, and 2) the effective utilization of temporal information predicated on the aforementioned accurate alignment outcomes during refinement. We introduce a novel referring tracker and temporal refiner to construct the **D**ecoupled **VIS** framework (**DVIS**). DVIS achieves new SOTA performance in both VIS and VPS, surpassing the current SOTA methods by 7.3 AP and 9.6 VPQ on the OVIS and VIPSeg datasets, which are the most challenging and realistic benchmarks. Moreover, thanks to the decoupling strategy, the referring tracker and temporal refiner are super light-weight (only 1.69% of the segmenter FLOPs), allowing for efficient training and inference on a single GPU with 11G memory. The code is available at https://github.com/zhang-tao-whu/DVIS.*

1. Introduction

Video Instance Segmentation (VIS) is a critical computer vision task that involves identifying, segmenting, and tracking all interested instances in a video simultaneously. This task was first introduced in [31]. The importance of VIS lies in its ability to facilitate many downstream computer vision applications, such as online autonomous driving and offline video editing.

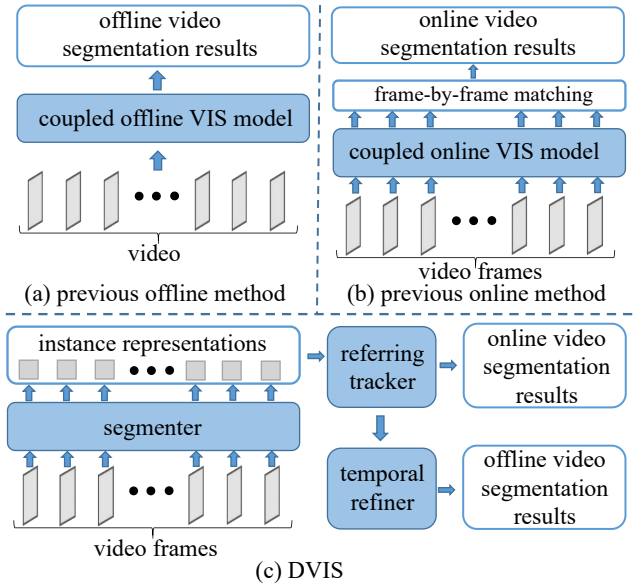

Figure 1. **Pipelines of previous offline (a), online (b), and proposed DVIS (c) frameworks.** Unlike previous methods that rely on tightly coupled networks, DVIS consists of independent components, including a segmenter, a referring tracker, and a temporal refiner.

Previous studies [12, 6, 28, 10] have demonstrated successful performance validation on videos with short durations and simple scenes [31]. However, in real-world scenarios, videos often present highly complex scenes, severe instance occlusions, and prolonged durations [21]. As a result, these approaches [12, 6, 28, 10] have exhibited poor performance on videos [21] that are more representative of real-world scenarios.

We believe that the fundamental reason for the failure of the aforementioned methods [12, 6, 28, 10] lies in the assumption that a coupled network can effectively predict the video segmentation results for any video, irrespective of its length, scene complexity, or instance occlusion lev-

*Corresponding author.

els. In the case of lengthy videos (*e.g.* 100 seconds and 500 frames), with intricate scenes, the same instance may exhibit significant variations in position, shape, and size between the first and last frames [21]. Even for experienced humans, accurately associating the same instance in two frames that are separated by a considerable interval is challenging without observing its gradual transformation trajectory over time. Therefore, the alignment/tracking difficulty is significantly increased in complex scenarios and lengthy videos, and even cutting-edge methods such as [5] face challenges in achieving convergence [11].

To tackle the aforementioned challenges, we propose to decouple the VIS task into three sub-tasks that are independent of video length and complexity: segmentation, tracking, and refinement. Segmentation aims to extract all appearing objects and obtain their representations from a single frame. Tracking aims to link the same object between adjacent frames. Refinement utilizes all temporal information of the object to optimize both segmentation and association results. Thus we have our decoupled VIS framework, as illustrated in Figure 1 (c). It contains three separate and independent components, *i.e.*, a segmenter, a tracker, and a refiner. Given the extensive research on the segmenter in the field of image instance segmentation, our focus is to design an effective tracker for robustly associating objects across adjacent frames and a refiner for improving the quality of segmentation and tracking.

To achieve effective instance association, we propose the following principles: (1) encourage sufficient interaction between instance representations of adjacent frames to fully exploit their similarity for better association. (2) avoid mixing their information during the interaction process to prevent introducing indistinguishable noise that may interfere with the association results. Current SOTA methods, such as [29, 11], violate principle 1 by utilizing heuristic algorithms to match adjacent frame instance representations without any interaction, resulting in a significant performance gap compared to our method. While [9, 35] achieve interaction between instance representations of adjacent frames by passing instance representations, they violate principle 2. Following both principles, we designed the Referring Cross Attention (RCA) module, which serves as the core component of our highly effective referring tracker. RCA is a modified version of standard cross-attention [4] that introduces identification to avoid the blending of instance representations in consecutive frames and efficiently utilize their similarities. We further propose a novel temporal refiner that leverages 1D convolution and self-attention to effectively integrate temporal information, and cross-attention to correct instance representations.

An decoupled VIS framework, called DVIS, is then naturally constructed by combining the segmenter, the referring tracker, and the temporal refiner. DVIS achieves new SOTA performance on all the VIS datasets, surpassing previous SOTA method [29] by 7.3 AP on the most challenging OVIS dataset [21]. Additionally, DVIS can be seamlessly extended to other video segmentation tasks, such as video panoptic segmentation (VPS) [13], without any modification. DVIS also achieves new SOTA performance on the video panoptic segmentation dataset VIPSeg [20], surpassing previous SOTA method [1] by 9.6 VPQ. DVIS achieved **1st place** in the VPS Track of the PVUW challenge at CVPR 2023.

Our decoupling strategy not only significantly improves the performance of video segmentation, but also dramatically reduces hardware resource requirements. Specifically, our proposed tracker and refiner operate exclusively on the instance representations output by the segmenter, avoiding the significant computational cost associated with interacting with image features. As a result, the computation cost of the tracker and refiner is negligible (only 5.18%/1.69% of the segmenter with R50/Swin-L backbone). Thanks to the decoupling design of the VIS task and framework, the tracker and refiner can be trained separately while keeping other components frozen. These advantages allow DVIS to be trained on a single GPU with 11G memory.

In summary , our main contributions are:

- We investigate the failure reasons of current methods on complex and lengthy real-world videos, and we address these challenges by introducing a novel decoupling strategy for VIS, which involves decomposing it into three decoupled sub-tasks: segmentation, tracking, and refinement.
- Following the decoupling strategy, we propose DVIS, which includes a simple yet effective referring tracker and temporal refiner to produce precision alignment results and efficiently utilize temporal information, respectively.
- DVIS achieves new SOTA performance in both VIS and VPS, as validated on five major benchmarks: OVIS [21], YouTube-VIS [31] 2019, 2021, and 2022, as well as VIPSeg [20]. Notably, DVIS significantly reduces the resources required for video segmentation, enabling efficient training and inference on a single GPU with 11G memory.

2. Related Works

Online Video Instance Segmentation. Most mainstream online VIS methods follow a pipeline of segmenting and associating instances. MaskTrack R-CNN [31] incorporates a tracking head based on [8] and associates instances in adjacent frames using multiple cues such as similarity score, semantic consistency, spatial correlation, and detection confidence. [3] replaces the segmenter in the

above pipeline with a one-stage instance segmentation network. [32] proposes a crossover learning scheme that segments the same instances in another frame using the instance features of the current frame. With stronger segmenters and the widespread application of transformers in vision tasks [6, 4, 33], recent works such as [11, 29, 34] have achieved outstanding performance. [11] proposes a minimal VIS framework based on [6] that achieves instance association by measuring the similarity between the same instances in adjacent frames. [29, 34] introduce contrastive learning in VIS to obtain a more discriminative instance representation. [35, 9] completely remove heuristic matching algorithms by delivering instance representations and modeling inter-frame association. Inspired by [30, 9, 11, 19], DVIS also performs tracking based on instance representations, which significantly reduces memory requirements. Our proposed DVIS introduces a novel component called the referring tracker, which models inter-frame association by denoising current instance representations with the help of previous frame instance representations.

Offline Video Instance Segmentation. Previous offline video instance segmentation (VIS) methods have used various approaches to model the spatio-temporal representations of instances in the video. In [2], instance spatio-temporal embeddings are modeled using a 3D CNN. The first transformer-based VIS architecture proposed in [26] uses learnable initial embeddings for each instance of each frame, making it challenging to model instances with complex motion trajectories. [12] introduces inter-frame communication, which reduces computation and memory overhead while improving performance. By directly modeling a video-level representation for each instance, [5] achieves impressive results. [28] constructs a VIS framework based on deformable attention [36] to separate temporal and spatial interactions between instance representations and videos. To significantly reduce memory consumption and enable offline methods to handle long videos, [10] constructs the video-level instance representation from the instance representations of each frame. While [9] implements a semi-online VIS framework by replacing frames with clips, no significant gains were observed compared to the online version. The current SOTA methods for VIS have been demonstrated to overlook the importance of the refinement sub-task. Specifically, the refinement process has been neglected by [29, 11, 35, 9], while [12, 5, 28, 10] exhibit a lack of clear separation between refinement and other aspects of the segmentation and tracking sub-tasks. Our proposed DVIS achieves SOTA performance by decoupling the VIS task and designing an efficient temporal refiner to fully utilize the information of the overall video.

3. Method

By reflecting on and summarizing the shortcomings of [11, 10], we have proposed DVIS, a novel decoupled framework for VIS that consists of three independent components: a segmenter, a referring tracker, and a temporal refiner, illustrated in Figure 1(c). Specifically, we use Mask2Former [6] as the segmenter in DVIS. The referring tracker is introduced in Section 3.1, while the temporal refiner is presented in Section 3.2.

3.1. Referring Tracker

The referring tracker models the inter-frame association as a referring denoising task. The referring cross-attention is the core component of the referring tracker that effectively utilizes the similarity between instance representations of adjacent frames while avoiding their mixture.

Architecture. Figure 2 illustrates the architecture of the referring tracker. It takes in the instance queries $\{Q^i_{seg}|i \in [1,T]\}$ generated by the segmenter and outputs the instance queries $\{Q^i_{Tr}|i \in [1,T]\}$ corresponding to the instances in the previous frame for the current frame, where T is the length of the video. Firstly, the hungarian matching algorithm [14] is employed to match Q_{seg} of adjacent frames, as is done in [11]:

$$\begin{cases} \tilde{Q}^i_{seg} = \text{Hungarian}(\tilde{Q}^{i-1}_{seg}, Q^i_{seg}), & i \in [2,T] \\ \tilde{Q}^i_{seg} = Q^i_{seg}, & i = 1 \end{cases}, \quad (1)$$

where $\tilde{Q}_{seg}$ is the matched intance queries of the segmenter. The hungarian matching algorithm is not strictly necessary and omitting it results in only a slight performance degradation, as shown in Section 4.2. $\tilde{Q}_{seg}$ can be considered as the tracking result with noise and serves as the initial query for the referring tracker. To denoise the initial query $\tilde{Q}^i_{seg}$ of the current frame, the online tracker uses the denoised instance queries Q^{i-1}_{Tr} from the previous frame as a reference.

The objective of the referring tracker is to refine the initial value with noise, which may contain incorrect tracking results, and produce accurate tracking results. The referring tracker comprises a sequence of L transformer denoising blocks, each of which consists of a referring cross-attention, a standard self-attention, and a feedforward network (FFN).

The referring cross-attention (RCA) is a crucial component of the denoising block, designed to capture the correlation between the current frame and its historical frames. Since the instance representations in adjacent frames are highly similar but differ in position, shape, size, etc., using the previous frame's instance representation as the initial instance representation for the current frame (as done by [35, 9]) can introduce ambiguous information that makes the denoising task more difficult. RCA overcomes this issue by introducing identification (ID), while still effectively utilizing the similarity between the query (Q) and key (K) to

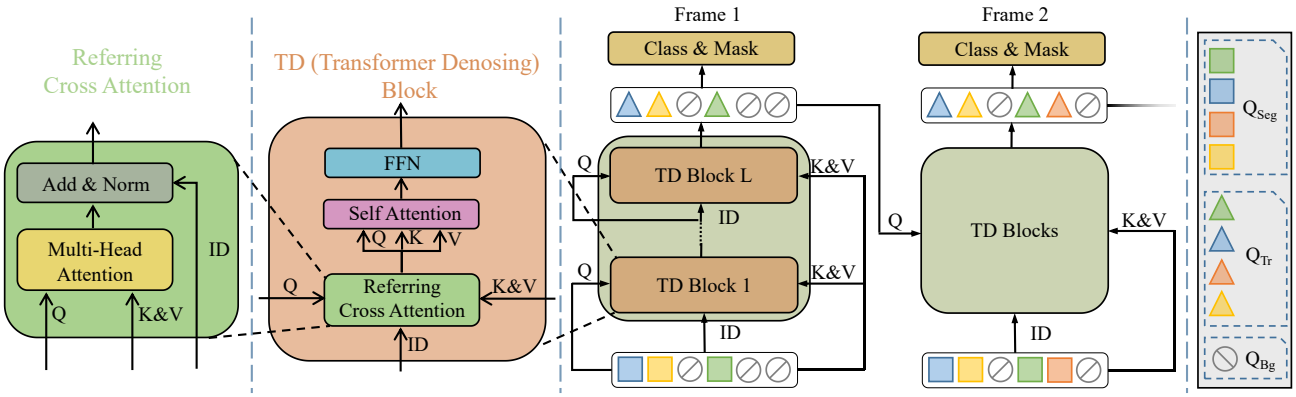

Figure 2. **The framework of the referring tracker.** The instance representations output by the segmenter (Q_{seg}) and referring tracker (Q_{Tr}) are represented by squares and triangles, respectively. Instances with the same ID are assigned the same color.

generate the correct output. As shown in Figure 2, RCA is inspired by [23] and differs only slightly from the standard cross-attention:

$$RCA(ID, Q, K, V) = ID + MHA(Q, K, V). \quad (2)$$

MHA refers to Multi-Head Attention [25], while ID, Q, K, and V denote identification, query, key, and value, respectively.

Finally, the denoised instance query Q_{Tr} is utilized as an input for the class head and mask head, which produce the category and mask coefficient output, respectively.

Losses. The referring tracker tracks instances frame by frame, and as such, the network is supervised using a loss function that aligns with this paradigm. Specifically, the instance label and prediction $\hat{y}_{Tr}$ are only matched on the frame where the instance first appears. To expedite convergence during the early training phase, the prediction of the frozen segmenter $\hat{y}_{seg}$ is used for matching instead of the referring tracker's prediction.

$$\begin{cases} \hat{\sigma} = \arg\min_{\sigma} \sum_{i=1}^{N} \mathcal{L}_{match}(y_i^{f(i)}, \hat{y}_{\sigma(i)}^{f(i)}) \\ \hat{y} = \hat{y}_{Tr} \;\; if \;\; Iter \geq \dfrac{Max\_Iter}{2} \;\; else \;\; \hat{y}_{seg} \end{cases}, \quad (3)$$

where $f(i)$ represents the frame in which the i-th instance first appears. $\mathcal{L}_{match}(y_i^{f(i)}, \hat{y}_{\sigma(i)}^{f(i)})$ is a pair-wise matching cost, as used in [5], between the ground truth y and the prediction $\hat{y}$ having index $\sigma(i)$ on the $f(i)$ frame.

The loss function $\mathcal{L}$ is exactly the same as that in [5].

$$\mathcal{L}_{Tr} = \sum_{t=1}^{T} \sum_{i=1}^{N} \mathcal{L}(y_i^t, \hat{y}_{\hat{\sigma}(i)}^t). \quad (4)$$

3.2. Temporal Refiner

The failure of previous offline video instance segmentation methods can mainly be attributed to the challenge of effectively leveraging temporal information in highly coupled networks. Additionally, previous online video instance segmentation methods lacked a refinement step. To address these issues, we developed an independent temporal refiner to effectively utilize information from the entire video and refine the output of the referring tracker.

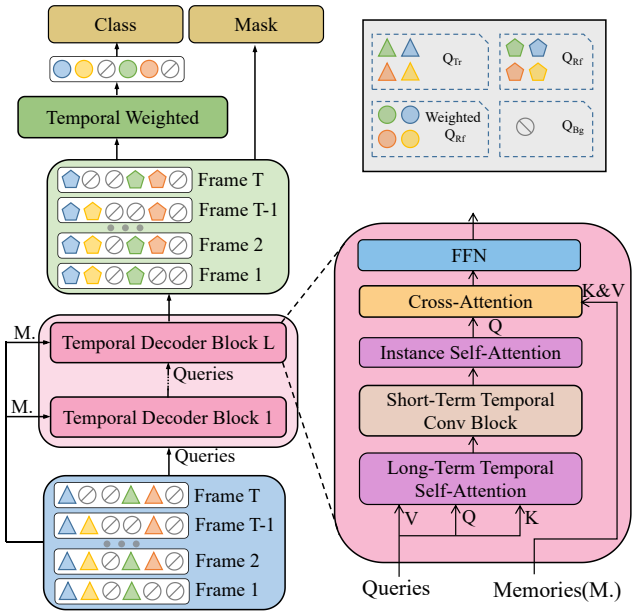

Figure 3. **The framework of the temporal refiner.** Instance representations for each frame (Q_{Rf}) are denoted by pentagons, while the instance representations for the entire video ($\hat{Q}_{Rf}$) are denoted by circles. Different colors indicate different instance IDs.

Architecture. Figure 3 shows the architecture of the temporal refiner. It takes the instance query Q_{Tr} output from the referring tracker as input and outputs the instance query Q_{Rf} after fully aggregating the overall information of the video. The temporal refiner is composed of L temporal decoder blocks that are cascaded together. Each temporal decoder block consists of two main components, namely the

short-term temporal convolution block and the long-term temporal attention block. The short-term temporal convolution block exploits motion information while the long-term temporal attention block exploits information from the entire video. These blocks are implemented using 1D convolution and standard self-attention, respectively, and both operate in the time dimension.

Lastly, the mask coefficients for each instance in each frame are predicted by the mask head based on the refined instance query Q_{Rf}. The class head predicts the category and score for each instance across the entire video, using the temporal weighting of Q_{Rf}. The temporal weighting process can be defined as follows:

$$\hat{Q}_{Rf} = \sum_{t=1}^{T} SoftMax(Linear(Q_{Rf}^{t}))Q_{Rf}^{t}, \quad (5)$$

where $\hat{Q}_{Rf}$ is the temporal weighting of Q_{Rf}.

Losses. The same matching cost and loss functions as [5] are used to supervise the temporal refiner during training. The segmenter and referring tracker are frozen during training, and therefore the referring tracker's prediction results are used for matching in the early training phase to guide the network towards faster convergence.

$$\begin{cases} \hat{\sigma} = \arg\min_{\sigma} \sum_{i=1}^{N} \mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)}) \\ \hat{y} = \hat{y}_{Rf} \;\; if \;\; Iter \geq \dfrac{Max\_Iter}{2} \;\; else \;\; \hat{y}_{Tr} \end{cases}, \quad (6)$$

where $\hat{y}_{Rf}$ is the prediction result of the temporal refiner. The loss function is:

$$\mathcal{L}_{Rf} = \sum_{i=1}^{N} \mathcal{L}(y_i, \hat{y}_{\hat{\sigma}(i)}). \quad (7)$$

4. Experiments

We evaluate the performance of DVIS for VIS on the OVIS [21], YouTube VIS 2019, 2021, and 2022 [31] datasets, and for VPS on the VIPSeg [20] dataset. In Appendix, the descriptions of these datasets can be found in Section A, while implementation details, including network training and inference settings, are provided in Section B.

4.1. Main Results

We compare DVIS with current SOTA online and offline VIS methods on the OVIS, YouTube-VIS 2019, 2021, and 2022 datasets. When compared with online methods, DVIS will discard the temporal refiner as it utilizes information from future frames, in order to maintain a fair comparison. The results are reported in Tables 1, 2, and 3, respectively. We adopt MinVIS [11] as our baseline because DVIS is essentially identical to MinVIS after removing the referring tracker and temporal refiner. We also compare DVIS with current SOTA methods for VPS, and the results are shown in Table 4. The visualization of DVIS's prediction results on these datasets is available in Figures I, II, and III of the Appendix.

		Method	OVIS				
			AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Online	ResNet50	MaskTrack R-CNN [31]	10.8	25.3	8.5	7.9	14.9
		CMaskTrack R-CNN [21]	15.4	33.9	13.1	9.3	20.0
		CrossVIS [32]	14.9	32.7	12.1	10.3	19.8
		VISOLO [7]	15.3	31.0	13.8	11.1	21.7
		MinVIS [11]	25.0	45.5	24.0	13.9	29.7
		MinVIS† [11]	26.4	49.6	25.2	13.3	31.1
		IDOL [29]	28.2	51.0	28.0	14.5	**38.6**
		IDOL† [29]	30.2	51.3	30.0	15.0	37.5
		ROVIS [35]	30.2	53.9	30.1	13.6	36.3
		Ours	30.2	**55.0**	30.5	14.5	37.3
		Ours†	**31.0**	54.8	**31.9**	**15.2**	37.6
	Swin-L	MinVIS [11]	39.4	61.5	41.3	18.1	43.3
		MinVIS† [11]	41.6	65.2	42.8	19.3	45.1
		IDOL [29]	40.0	63.1	40.5	17.6	46.4
		IDOL† [29]	42.6	65.7	45.2	17.9	49.6
		ROVIS [35]	41.6	65.0	42.9	18.7	46.9
		ROVIS† [35]	42.6	64.7	42.6	18.4	49.1
		GenVis⋆ [9]	45.2	69.1	48.4	19.1	48.6
		Ours	45.9	71.1	48.3	18.5	51.5
		Ours†	**47.1**	**71.9**	**49.2**	**19.4**	**52.5**
Offline	ResNet50	IFC [12]	13.1	27.8	11.6	9.4	23.9
		SeqFormer [28]	15.1	31.9	13.8	10.4	27.1
		Mask2Former-VIS [5]	17.3	37.3	15.1	10.5	23.5
		VITA⋆ [10]	19.6	41.2	17.4	11.7	26.0
		Ours	33.8	**60.4**	**33.5**	15.3	39.5
		Ours†	**34.1**	59.8	32.3	**15.9**	**41.1**
	Swin-L	VITA⋆ [10]	27.7	51.9	24.9	14.9	33.0
		Mask2Former-VIS [5]	25.8	46.5	24.4	13.7	32.2
		GenVIS⋆ [9]	45.4	69.2	47.8	18.9	49.0
		MDQE† [15]	42.6	67.8	44.3	18.3	46.5
		Ours	48.6	74.7	50.5	18.8	53.8
		Ours†	**49.9**	**75.9**	**53.0**	**19.4**	**55.3**

Table 1. **Results on the OVIS validation set.** † denotes training and evaluation at 720px. ⋆ denotes using COCO pseudo videos. The best metrics in each group are bolded.

Performance on the OVIS Dataset. In online mode, DVIS achieves 31.0 AP with ResNet50 and 47.1 AP with Swin-L on the OVIS validation set, outperforming the baseline MinVIS [11] by 4.6 AP and 5.5 AP, respectively. The referring tracker has shown significant performance gains, especially for medium and heavily occluded objects, as discussed in Section 4.2. DVIS outperforms the current SOTA online VIS methods IDOL [29] and RO-VIS [35] by 4.5 AP. This demonstrates the successful design of the referring tracker for robust tracking results particularly in heavily occluded scenarios.

	Method	Backbone	Youtube-VIS 2019					Youtube-VIS 2021				
			AP	AP_{50}	AP_{75}	AR_1	AR_{10}	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Online	MaskTrack R-CNN [31]	ResNet-50	30.3	51.1	32.6	31.0	35.5	28.6	48.9	29.6	26.5	33.8
	SipMask [3]	ResNet-50	33.7	54.1	35.8	35.4	40.1	31.7	52.5	34.0	30.8	37.8
	CrossVIS [32]	ResNet-50	36.3	56.8	38.9	35.6	40.7	34.2	54.4	37.9	30.4	38.2
	VISOLO [7]	ResNet-50	38.6	56.3	43.7	35.7	42.5	36.9	54.7	40.2	30.6	40.9
	MinVIS [11]	ResNet-50	47.4	69.0	52.1	45.7	55.7	44.2	66.0	48.1	39.2	51.7
	IDOL [29]	ResNet-50	49.5	**74.0**	52.9	**47.7**	58.7	43.9	68.0	49.6	38.0	50.9
	Ours	ResNet-50	**51.2**	73.8	**57.1**	47.2	**59.3**	**46.4**	**68.4**	**49.6**	**39.7**	**53.5**
	MinVIS [11]	Swin-L	61.6	83.3	68.6	54.8	66.6	55.3	76.6	62.0	45.9	60.8
	IDOL [29]	Swin-L	**64.3**	**87.5**	**71.0**	55.6	**69.1**	56.1	**80.8**	63.5	45.0	60.1
	Ours	Swin-L	63.9	87.2	70.4	**56.2**	69.0	**58.7**	80.4	**66.6**	**47.5**	**64.6**
Offline	EfficientVIS [30]	ResNet-50	37.9	59.7	43.0	40.3	46.6	34.0	57.5	37.3	33.8	42.5
	IFC [12]	ResNet-50	41.2	65.1	44.6	42.3	49.6	35.2	55.9	37.7	32.6	42.9
	Mask2Former-VIS [5]	ResNet-50	46.4	68.0	50.0	-	-	40.6	60.9	41.8	-	-
	SeqFormer [28]	ResNet-50	47.4	69.8	51.8	45.5	54.8	40.5	62.4	43.7	36.1	48.1
	VITA [10]	ResNet-50	49.8	72.6	54.5	**49.4**	**61.0**	45.7	67.4	49.5	**40.9**	53.6
	Ours	ResNet-50	**52.6**	**76.5**	**58.2**	47.4	60.4	**47.4**	**71.0**	**51.6**	39.9	**55.2**
	SeqFormer [28]	Swin-L	59.3	82.1	66.4	51.7	64.4	51.8	74.6	58.2	42.8	58.1
	Mask2Former-VIS [5]	Swin-L	60.4	84.4	67.0	-	-	52.6	76.4	57.2	-	-
	VITA [10]	Swin-L	63.0	86.9	67.9	56.3	68.1	57.5	80.6	61.0	47.7	62.6
	Ours	Swin-L	**64.9**	**88.0**	**72.7**	**56.5**	**70.3**	**60.1**	**83.0**	**68.4**	**47.7**	**65.7**

Table 2. **Results on the validation set of YouTube-VIS 2019 & 2021.** The best metrics in each group are bolded.

	Method	YouTube-VIS 2022				
		AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Swin-L	VITA [10]	41.1	63.0	44.0	39.3	44.3
	MinVIS [11]	33.1	54.8	33.7	29.5	36.6
	Ours	**45.9**	69.0	**48.8**	37.2	**51.8**

Table 3. **Results on the YouTube-VIS 2022 long videos.** The best metrics in each group are bolded.

	Method	VIPSeg			
		VPQ	VPQ^{Th}	VPQ^{St}	STQ
R50	VPSNet [13]	14.0	14.0	14.2	20.8
	VPSNet-SiamTrack [27]	17.2	17.3	17.3	21.1
	VIP-Deeplab [22]	16.0	12.3	18.2	22.0
	Clip-PanoFCN [20]	22.9	25.0	20.8	31.5
	Video K-Net [17]	26.1	-	-	31.5
	TarVIS [1]	33.5	39.2	28.5	**43.1**
	Tube-Link [16]	39.2	-	-	39.5
	Video-kMax [24]	38.2	-	-	39.9
	Ours	**43.2**	**43.6**	**42.8**	42.8
Swin-L	TarVIS [1]	48.0	58.2	39.0	52.9
	Ours	**57.6**	**59.9**	**55.5**	**55.3**

Table 4. **Results on the VIPSeg dataset.** The best metrics in each group are bolded.

In offline mode, DVIS achieves 34.1 AP with ResNet50 and 49.9 AP with Swin-L on the OVIS validation set, surpassing DVIS running in online mode by 3.1 AP and 2.8 AP, respectively. More impressively, DVIS outperforms the baseline MinVIS by 8.3 AP. Additionally, DVIS surpasses the previous pure offline VIS methods Mask2Former-VIS [5] and VITA [10] by 24.1 AP and 22.2 AP, respectively. Thus, DVIS achieves a new SOTA performance, demonstrating the superiority of the decoupled framework in complex scenarios compared to the previous coupled framework.

Performance on the YouTube-VIS 2019 and 2021 Datasets. In online mode, DVIS achieved 51.2 AP with ResNet50 and 63.9 AP with Swin-L on the YouTube-VIS 2019 validation set, outperforming MinVIS [11] by 3.8 AP and 2.3 AP, respectively. For YouTube-VIS 2021 validation set, DVIS achieved 46.4 AP with ResNet50 and 58.7 AP with Swin-L in online mode, outperforming MinVIS [11] by 2.2 AP and 3.4 AP, respectively. On the YouTube-VIS datasets, DVIS running in online mode shows comparable performance with the current SOTA method IDOL [29].

In offline mode, DVIS achieved 52.6 AP with ResNet50 and 64.9 AP with Swin-L on the YouTube-VIS 2019 validation set, outperforming DVIS running in online mode by 1.4 AP and 1.0 AP. Similarly, on the YouTube-VIS 2021 validation set, DVIS achieved 47.4 AP with ResNet50 and 60.1 AP with Swin-L, outperforming DVIS (online mode) by 1.0 AP and 1.4 AP. DVIS achieves a new SOTA performance on the YouTube-VIS 2019 and 2021 datasets.

Performance on the YouTube-VIS 2022 Dataset. DVIS achieves a new SOTA performance of 52.8 AP on the validation set of the YouTube-VIS 2022 dataset, with 59.6 AP on short videos and 45.9 AP on long videos. Since the short videos of the YouTube-VIS 2022 dataset largely overlap with the YouTube-VIS 2021 dataset, we compare the performance of DVIS with other methods only on long videos, as shown in Table 3. DVIS outperforms the baseline method MinVIS [11] by 12.8 AP and the current SOTA method VITA [10] by 4.8 AP.

Performance on the VIPSeg Dataset. DVIS achieves

	AP_{all}	AP_l	AP_m	AP_h
baseline	41.6	64.8	49.0	20.5
+Tracker	47.1(+5.5)	64.7(-0.1)	54.2(+5.2)	24.8(+4.3)
+Refiner	49.9(+8.3)	67.1(+2.3)	56.0(+7.0)	29.8(+9.4)

Table 5. **Ablation study of the proposed components.** The baseline is MinVIS [11]. All models use Swin-L as the backbone and are evaluated on the OVIS validation set with 720p input. AP_l, AP_m and AP_h refer to the AP of the light, medium, and heavily occluded instances, respectively.

43.2 VPQ and 57.6 VPQ on the VIPSeg validation set when using ResNet50 and Swin-L backbones, respectively, surpassing the current SOTA VPS method TarVIS [1] by 9.7 VPQ and 9.6 VPQ. These results demonstrate the outstanding performance of DVIS on video panoptic segmentation (VPS) and its potential to achieve SOTA performance on all video segmentation tasks.

4.2. Ablation Experiments

Ablation experiments were conducted on the OVIS dataset, with DVIS evaluated using ResNet50 and input resized to 360p unless otherwise specified.

Effectiveness of Referring Tracker and Temporal Refiner. We conducted ablation experiments on the OVIS dataset to evaluate the effectiveness of the referring tracker and temporal refiner. The results of the experiments are presented in Table 5. Our findings indicate that the referring tracker leads to significant performance gains when processing medium and heavily occluded objects, resulting in an increase of 5.2 AP_m and 4.3 AP_h, respectively. However, there is a slight decrease of 0.1 AP_l in the case of lightly occluded objects, indicating that the improvement in the referring tracker is primarily in tracking quality rather than segmentation quality. We further illustrate our findings by presenting an instance of a completely occluded panda with ID 1 in the third frame, which is tracked well by DVIS but not by MinVIS [11], in Figure 4.

The temporal refiner leads to performance gains in both segmentation quality and tracking quality, with improvements of 2.3 AP_l, 7.0 AP_m, and 9.4 AP_h across the board. The temporal refiner effectively utilizes the entire video information, leading to more significant improvements for heavily occluded objects, as demonstrated in Figure 4 where the green rectangles highlight a highly occluded panda. Despite this challenge, the temporal refiner produces accurate segmentation results, while the referring tracker fails due to its inability to leverage the full video information.

Initial Instance Representation of Referring Tracker. Our proposed referring tracker represents the VIS as referring denoising, making it crucial to select an appropriate initial value with noise. We evaluate the performance with different initial values and report the results in Table 6. The best performance is achieved when using the Q_{seg} obtained by matching with the hungarian algorithm as the initial value. When zero is used as the initial value, the denoising task becomes a more challenging reconstruction problem, leading to a drop of 1.6 AP. Using the Q_{Tr} of the previous frame as the initial value results in a 2.2 AP performance degradation, as it contains too much interference information. The network also performs well when using the unmatched Q_{seg}, where the initial values of the instance queries of each frame are linked by the learnable prior information, demonstrating the robustness of the referring tracker.

Initial Value	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Zero	28.9	52.5	27.9	14.7	35.7
Q_{Tr}^{pre}	28.3	50.1	27.3	14.5	33.8
Q_{seg}	29.8	54.3	28.3	14.8	36.5
Matched Q_{seg}	**30.5**	**54.7**	**30.1**	**15.0**	**36.5**

Table 6. **Ablation study of the initial instance representation in the referring tracker.** Q_{Tr}^{pre} denotes the instance representation in the previous frame, and Q_{seg} denotes the instance representation output by the segmenter.

Cross Attn Type	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Standard	2.9	5.0	2.8	2.4	3.4
Referring	**30.5**	**54.7**	**30.1**	**15.0**	**36.5**

Table 7. **Ablation study on the type of cross-attention in the referring tracker.**

	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Baseline	32.2	57.9	31.3	15.1	38.7
w/o Long-term Attn.	31.0	56.2	29.3	14.9	37.8
w/o Short-term Conv.	31.8	56.6	30.2	14.8	37.9
w/o Cross Attn.	30.6	55.5	28.7	14.8	36.6

Table 8. **Ablation study on the components of the temporal decoder block.** Attn. denotes attention and Conv. denotes convolutional."w/o" refer to without.

Referring Cross-Attention. The referring cross-attention is a crucial component of the referring tracker, responsible for linking historical frames with the current frame. We evaluated the importance of the referring cross-attention by comparing it to the standard cross-attention, where ID is set to Q in Equation 2. The results in Table 7 demonstrate that replacing the referring cross-attention with the standard cross-attention leads to an extreme drop in performance. This finding highlights the critical role of inter-frame associations modeled by the referring cross-attention in the success of the referring tracker.

Impact of Different Components of Temporal Decoder Block. To evaluate the impact of different components of the temporal decoder block, we conducted experiments by removing each component individually and reporting the corresponding performance in Table 8. Our results show that removing long-term temporal self-attention led to a performance degradation of 1.2 AP. Although the function of the long-term attention overrides the function of the short-term convolution, removing the short-term convo-

Figure 4. **Visualization results comparing DVIS with current SOTA online and offline VIS methods.** VITA shows poor segmentation quality (highlighted with red circles) and tracking stability (highlighted with red rectangles). The referring tracker demonstrates strong tracking ability (highlighted with blue rectangles), while the temporal refiner effectively utilizes contextual information from previous and future frames (highlighted with green rectangles).

lution still resulted in a performance degradation of 0.4 AP, suggesting that it is beneficial for utilizing information in adjacent frames. Moreover, the removal of cross-attention resulted in a significant drop of 1.6 AP since incorrect instance queries cannot be efficiently corrected without it, even though information from different frames can still be utilized.

Performance of DVIS in Semi-Online Mode. In real-world scenarios, videos are often of infinite length, making it impossible to run VIS models in pure offline mode. We conduct experiments to measure the performance difference between semi-online and offline modes, as shown in Table II of Appendix. When videos are cut into clips of length 1 as input to DVIS, i.e., no other frame information available for the current frame, the performance was only comparable to that of DVIS without temporal refiner. However, as the clip length increased, the performance of the semi-online mode gradually approached that of the pure offline mode, and achieved comparable performance after the clip length exceeded 80 frames (33.8 $vs.$ 33.8).

Computational Cost. The computational cost of DVIS components was measured by evaluating the parameters, MACs, and inference time of the segmenter, referring tracker, and temporal refiner. Table 9 presents the results. When using Mask2Former with ResNet50 and Swin-L as the segmenter, the referring tracker and temporal refiner combined only accounted for 5.18% and 1.69% of the segmenter's computation, respectively. This demonstrates that the referring tracker and temporal refiner can efficiently achieve VIS with almost negligible computational cost.

Component	Inp.	N_Q	Params(M)	MACs(G)	Time(ms)
M2F(R50)	480p	100	43.95	103.73	48.10
Tracker	480p	100	9.68	1.68	7.63
Refiner	480p	100	14.41	3.69	1.11
M2F(SwinL)	720p	200	215.30	851.00	275.19
Tracker	720p	200	9.68	5.13	7.97
Refiner	720p	200	14.41	9.27	2.00

Table 9. **Computational cost of DVIS components.** M2F refers to the Mask2Former used as the segmenter of DVIS. Inp. denotes the size of the input video, and N_Q denotes the number of queries. The inference time per frame is measured on a 1080Ti GPU.

5. Conclusion

In this paper, we propose DVIS, a decoupled VIS framework that separates the VIS task into three sub-tasks: segmentation, tracking, and refinement. Our contributions are three-fold: 1) we decouple the VIS task and introduce the DVIS framework, 2) we propose the referring tracker, which enhances tracking robustness by modeling inter-frame associations as referring denoising, and 3) we propose the temporal refiner, which utilizes information from the entire video to refine segmentation results, a capability that was missing in previous methods. Our results show that DVIS achieves SOTA performance on all VIS datasets, outperforming all existing methods, supporting the effectiveness of our decoupling standpoint and the design of DVIS. Additionally, DVIS's SOTA performance on VPS demonstrates its potential and versatility. We believe that DVIS will serve as a strong and fundamental baseline, and our decoupling insights will inspire future works in both online and offline VIS.

References

[1] Ali Athar, Alexander Hermans, Jonathon Luiten, Deva Ramanan, and Bastian Leibe. Tarvis: A unified approach for target-based video segmentation. *arXiv preprint arXiv:2301.02657*, 2023.

[2] Ali Athar, Sabarinath Mahadevan, Aljosa Osep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 158–177. Springer, 2020.

[3] 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. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 1–18. Springer, 2020.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[5] 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.

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022.

[7] Su Ho Han, Sukjun Hwang, Seoung Wug Oh, Yeonchool Park, Hyunwoo Kim, Min-Jung Kim, and Seon Joo Kim. Visolo: Grid-based space-time aggregation for efficient online video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2896–2905, 2022.

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

[9] 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.

[10] 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.

[11] De-An Huang, Zhiding Yu, and Anima Anandkumar. Minvis: A minimal video instance segmentation framework without video-based training. *arXiv preprint arXiv:2208.02245*, 2022.

[12] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. Video instance segmentation using inter-frame communication transformers. *Advances in Neural Information Processing Systems*, 34:13352–13363, 2021.

[13] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9859–9868, 2020.

[14] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[15] Minghan Li, Shuai Li, Wangmeng Xiang, and Lei Zhang. Mdqe: Mining discriminative query embeddings to segment occluded instances on challenging videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10524–10533, 2023.

[16] Xiangtai Li, Haobo Yuan, Wenwei Zhang, Guangliang Cheng, Jiangmiao Pang, and Chen Change Loy. Tube-link: A flexible cross tube baseline for universal video segmentation. *arXiv preprint arXiv:2303.12782*, 2023.

[17] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18847–18857, 2022.

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

[19] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8844–8854, 2022.

[20] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21033–21043, 2022.

[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. *International Journal of Computer Vision*, 130(8):2022–2039, 2022.

[22] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3997–4008, 2021.

[23] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 208–223. Springer, 2020.

[24] Inkyu Shin, Dahun Kim, Qihang Yu, Jun Xie, Hong-Seok Kim, Bradley Green, In So Kweon, Kuk-Jin Yoon, and Liang-Chieh Chen. Video-kmax: A simple unified approach for online and near-online video panoptic segmentation. *arXiv preprint arXiv:2304.04694*, 2023.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia

Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[26] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8741–8750, 2021.

[27] Sanghyun Woo, Dahun Kim, Joon-Young Lee, and In So Kweon. Learning to associate every segment for video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2705–2714, 2021.

[28] Junfeng Wu, Yi Jiang, Song Bai, Wenqing Zhang, and Xiang Bai. Seqformer: Sequential transformer for video instance segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 553–569. Springer, 2022.

[29] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In defense of online models for video instance segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 588–605. Springer, 2022.

[30] Jialian Wu, Sudhir Yarram, Hui Liang, Tian Lan, Junsong Yuan, Jayan Eledath, and Gerard Medioni. Efficient video instance segmentation via tracklet query and proposal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 959–968, 2022.

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

[32] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8043–8052, 2021.

[33] Kaining Ying, Zhenhua Wang, Cong Bai, and Pengfei Zhou. Isda: Position-aware instance segmentation with deformable attention. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2619–2623. IEEE, 2022.

[34] Kaining Ying, Qing Zhong, Weian Mao, Zhenhua Wang, Hao Chen, Lin Yuanbo Wu, Yifan Liu, Chengxiang Fan, Yunzhi Zhuge, and Chunhua Shen. Ctvis: Consistent training for online video instance segmentation. *arXiv preprint arXiv:2307.12616*, 2023.

[35] Zitong Zhan, Daniel McKee, and Svetlana Lazebnik. Robust online video instance segmentation with track queries. *arXiv preprint arXiv:2211.09108*, 2022.

[36] 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.