Supplementary Material: Context-Aware Relative Object Queries to Unify Video Instance and Panoptic Segmentation

Introduction

In this paper we develop a simple approach to unify the video segmentation tasks VIS, VPS, and MOTS. The developed approach propagates what we refer to as 'context-aware relative object queries.'

In this supplementary material we provide additional details, analysis and results to support the points we made in Sec. 3 of the main paper. Specifically, in Sec. A, we discuss computation of the relative positional encodings ξ^{rel} which are used to calculate the cross attention matrix $\alpha^{\text{rel},l}_{\tau_t}$ described in Eq. (3). We then discuss our training procedure in Sec. B. In Sec. C, we provide some additional qualitative results on the OVIS, Youtube-VIS 2021 and Cityscapes-VPS data and perform additional ablation studies on the OVIS data. Finally, we provide more implementation details in Sec. D.

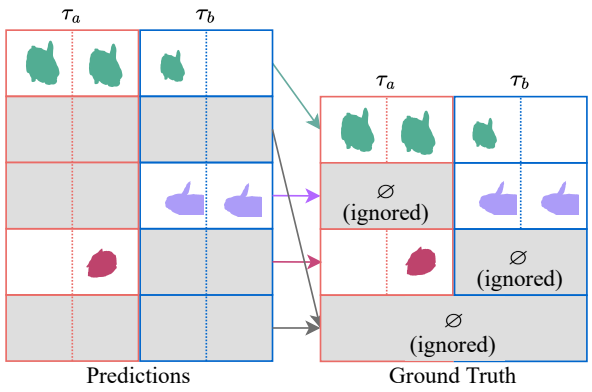

Figure S1. Hungarian matching. The grey boxes denote the no-object classes to be ignored during the calculation of $\mathcal{L}^{\text{match}}$. These no-objects don't vote towards obtaining the optimal matching.

A. Relative Position Encodings

In Sec. 3.3, we introduced the relative positional encodings $\xi^{\text{rel},l} \in \mathbb{R}^{H^l W^l T \times C}$ in Eq. (3). Each row of $\xi^{\text{rel},l}$ refers to the relative distance between two positions of the object queries and context features. In this section, inspired by [13], we discuss how to efficiently compute $\xi^{\text{rel},l}$.

The relative positional encodings $\xi^{\text{rel},l}$ are calculated from a learnt set of embeddings $E^{\text{rel}} \in \mathbb{R}^{L_{\max} \times C}$ where $L_{\max}$ is the maximum possible difference between the index of any row of the query vectors q^{prev} and the index of any row in the context-features $U^l_{\tau_t}$. The relative distance between the i_1-th row of q^{prev} and the i_2-th row of $U^l_{\tau_t}$ is stored in the $(N - i_1 + i_2 - 1)$-th row in E^{rel}. Since the row indices of q^{prev} vary from 0 to $N-1$ and the row indices of $U^l_{\tau_t}$ vary from 0 to $H^l W^l T$, the maximally possible difference for layer l is $N + H^l W^l T - 1$. Let $H_{\max} \times W_{\max}$ be the maximum resolution of the context-features supported (such that $H^l \leq H_{\max}$ and $W^l \leq W_{\max}$ always), then $L_{\max} = N + H_{\max} W_{\max} T - 1$. Intuitively, E^{rel} contains all the relative distances and acts as a look-up table for obtaining $\xi^{\text{rel},l}$, which is then used to calculate the relative attention matrix as described in Eq. (3).

B. Training

During training we minimize a spatio-temporal set prediction objective. Although our inference is frame-by-frame, we adopt a clip-based training procedure to introduce temporal context in the transformer decoder during training.

Specifically, we construct a training sample using 2 clips, each having T random frames (in ascending order) from a given video. T refers to the context-length used during inference (see Sec. 3.4). Given the 2 clips, τ_a and τ_b, we have a set of N predictions and a set of ground truth objects (padded with a no-object category $\varnothing$ to equal the number of predictions N). We first match the ground truth objects with the predictions jointly for both clips by minimizing a matching cost using Hungarian matching [39]. The optimal matching is then used to calculate the final objective function. Importantly, in contrast to prior work, in our setting the ground truth class labels for the same object may differ for the two clips τ_a and τ_b. For example, an object could be nascent in τ_a ($\varnothing$ class label) and expressed in τ_b (the actual object class label). To deal with this, we ignore the matching-cost for the $\varnothing$ object in the corresponding clip. We provide more details regarding the two step procedure next.

Step 1: Matching. The ground truth of an object corresponding to a clip consists of the object's class labels and segmentation masks in that clip. Specifically, each ground truth object i is represented by two pairs, one for the target class labels $(c^i_{\tau_a}, c^i_{\tau_b})$ and one for the ground truth segmentation masks $(s^i_{\tau_a}, s^i_{\tau_b})$. The elements within each pair correspond to the 2 clips τ_a and τ_b. The target class labels $c^i_{\tau_a}$ and $c^i_{\tau_b}$ are scalars and represent the i^{th} elements of the ground truth class vectors $C^{\text{gt}}_{\tau_a}$ and $C^{\text{gt}}_{\tau_b}$, where $C^{\text{gt}}_{\tau_a}, C^{\text{gt}}_{\tau_b} \in \{\varnothing, 1, \ldots, K\}^N$. Each target class label belonging to a clip represents the true object class if the object has appeared in any of the frames in the clip, otherwise it is $\varnothing$. Note, in our formulation, $c^i_{\tau_a}$ and $c^i_{\tau_b}$ may differ even though they represent the same object, which differs from prior work [7,8]. For instance, they could be $\varnothing$ in one clip if the object is nascent, and the actual object class in the other clip if the object is expressed. The segmentation masks $s^i_{\tau_a}$ and $s^i_{\tau_b}$ represent the i^{th} elements of the ground truth segmentation mask tensors $S^{\text{gt}}_{\tau_a}$ and $S^{\text{gt}}_{\tau_b}$, where

$S^{\text{gt}}_{\tau_a}, S^{\text{gt}}_{\tau_b} \in \{0,1\}^{N \times T \times H \times W}$.

The goal of step 1 is to obtain an optimal matching $\hat{\sigma}$ that minimizes a pair-wise matching cost $\mathcal{L}^{\text{match}}$ between the ground truth and predicted objects.

Let us use σ to represent any matching between the ground truth objects and the predictions. The prediction σ_i matched to object i consists of the predicted probabilities of the ground truth classes, $p^{\sigma_i}_{\tau_a}(c^i_{\tau_a})$ and $p^{\sigma_i}_{\tau_b}(c^i_{\tau_b})$, and the mask predictions , $s^{\sigma_i}_{\tau_a}$ and $s^{\sigma_i}_{\tau_b}$, for clips τ_a and τ_b. The pair-wise matching cost ($\mathcal{L}^{\text{match}}(\sigma_i)$) between the ground truth object i and a prediction with index σ_i consists of classification losses, $\mathcal{L}^{\text{cls.}}_{\tau_a}(\sigma_i)$ and $\mathcal{L}^{\text{cls.}}_{\tau_b}(\sigma_i)$, and segmentation losses, $\mathcal{L}^{\text{seg.}}_{\tau_a}(\sigma_i)$ and $\mathcal{L}^{\text{seg.}}_{\tau_b}(\sigma_i)$, for each clip. Here, $\mathcal{L}^{\text{cls.}}_{\tau_a}(\sigma_i) = -\log p^{\sigma_i}_{\tau_a}(c^i_{\tau_a})$ and $\mathcal{L}^{\text{cls.}}_{\tau_b}(\sigma_i) = -\log p^{\sigma_i}_{\tau_b}(c^i_{\tau_b})$ are the clip-wise cross entropy losses; $\mathcal{L}^{\text{seg.}}_{\tau_a}(\sigma_i)$ and $\mathcal{L}^{\text{seg.}}_{\tau_b}(\sigma_i)$ are the losses corresponding to the clip-wise segmentation masks of the [prediction, ground truth] pair. We follow [8] to compute the segmentation losses and classification losses for individual clips. However, note that in a [8]-like setting, the inter-clip matching cost is straightforwardly given as $\mathcal{L}^{\text{match}}(\sigma_i) = \mathbb{1}_{\{c^i_{\tau_a}=c^i_{\tau_b}\neq\varnothing\}}[\mathcal{L}^{\text{cls.}}_{\tau_a}(\sigma_i) + \mathcal{L}^{\text{seg.}}_{\tau_a}(\sigma_i) + \mathcal{L}^{\text{cls.}}\tau_b(\sigma_i) + \mathcal{L}^{\text{seg.}}_{\tau_b}(\sigma_i)]$, since $c^i_{\tau_a}$ is always equal to $c^i_{\tau_b}$. Different from this prior work, in our setting, the class labels for the same object can differ between frames τ_a and τ_b. Hence, the loss $\mathcal{L}^{\text{match}}(\sigma_i)$ has to be modified. We modify the loss as follows:

$$\begin{aligned}\mathcal{L}^{\text{match}}(\sigma_i) = \mathbb{1}_{\{c^i_{\tau_a}\neq\varnothing\}}[\mathcal{L}^{\text{cls.}}_{\tau_a}(\sigma_i) + \mathcal{L}^{\text{seg.}}_{\tau_a}(\sigma_i)] \\ +\mathbb{1}_{\{c^i_{\tau_b}\neq\varnothing\}}[\mathcal{L}^{\text{cls.}}_{\tau_b}(\sigma_i) + \mathcal{L}^{\text{seg.}}_{\tau_b}(\sigma_i)]. \quad \text{(S1)}\end{aligned}$$

Intuitively, if the class labels are the same for both frames ($c^i_{\tau_a} = c^i_{\tau_b}$), a [8]-like setting is directly applicable. We only keep the class probabilities corresponding to a true object (no-objects are rejected) while calculating the matching loss. This is because no-objects shouldn't vote towards obtaining the best match. For unequal class labels, we adopt the strategy of ignoring the no-object category: If $c^i_{\tau_a} = \varnothing$ and $c^i_{\tau_b} \neq \varnothing$, or $c^i_{\tau_a} \neq \varnothing$ and $c^i_{\tau_b} = \varnothing$, i.e., the ground truth object is expressed in one of the clips and nascent in the other, we ignore the no-object category in the corresponding clip. These different scenarios are shown in Fig. S1. We later show the effectiveness of our modified loss in Sec. C.2 and Tab. S2.

We find the optimal bipartite matching $\hat{\sigma}$ that minimizes $\mathcal{L}^{\text{match}}$ using the Hungarian algorithm [39].

Step 2: Final objective. Once the best matching $\hat{\sigma}$ is obtained, the final training objective is the sum of the classification loss and the segmentation loss of the optimally matched [prediction, ground truth] pairs, but the objects nascent in both clips aren't ignored any more in the class loss (this differs from step 1) and they contribute to the overall training objective. The final training objective $\mathcal{L}^{\text{obj.}}$ reads as follows:

$$\begin{aligned}\mathcal{L}^{\text{obj.}}(\hat{\sigma}_i) = \mathbb{1}_{\{c^i_{\tau_a}\neq\varnothing\}}[\mathcal{L}^{\text{cls.}}_{\tau_a}(\hat{\sigma}_i) + \mathcal{L}^{\text{seg.}}_{\tau_a}(\hat{\sigma}_i)] \\ +\mathbb{1}_{\{c^i_{\tau_b}\neq\varnothing\}}[\mathcal{L}^{\text{cls.}}_{\tau_b}(\hat{\sigma}_i) + \mathcal{L}^{\text{seg.}}_{\tau_b}(\hat{\sigma}_i)] \\ +\mathbb{1}_{\{c^i_{\tau_b}=c^i_{\tau_b}=\varnothing\}}[\mathcal{L}^{\text{cls.}}_{\tau_a}(\hat{\sigma}_i) + \mathcal{L}^{\text{cls.}}_{\tau_b}(\hat{\sigma}_i)]. \quad \text{(S2)}\end{aligned}$$

Intuitively, all the class probabilities for N objects are included in the class loss for the optimal matching, regardless of whether the objects are nascent in both clips or present in either. However, note that if an object is nascent in one clip but expressed in the other, we consider it in the class loss only for the clip it is expressed in.

Training data. A training sample consists of two randomly chosen clips τ_a and τ_b from a given video in the training dataset, ensuring that clip τ_a precedes τ_b. Each clip is of length T and, during training, consists of randomly picked video frames in order. For example, if $T = 2$, one training sample consists of 4 frames (2 clips), $\{\tau_a, \tau_b\} = \{(t_{a_0}, t_{a_1}), (t_{b_0}, t_{b_1})\}$, where $t_{a_0}, t_{a_1}, t_{b_0}, t_{b_1}$ represent the 4 frames and $a_0 < a_1 < b_0 < b_1$. We perform query vector propagation on τ_a and τ_b as described in Sec. 3.2. The query vectors for τ_a are obtained using the learnt query embeddings and are propagated to obtain the query vectors for τ_b. Note, this differs from the inference procedure where the frames are processed consecutively.

C. Additional Results

In Sec. 4, we show the effectiveness of the proposed approach on the task of VIS, MOTS and VPS. Importantly, the approach of query vector propagation can be generalized to a wide range of tasks. In this section, we provide additional results and analysis. First we show some additional qualitative results on the OVIS, Youtube-VIS 2021 and Cityscapes-VPS data followed by additional ablation studies.

C.1. Additional Qualitative results

Fig. S2 provides qualitative results on 4 videos from the OVIS data. The first video frame is shown in each example for reference. The top-left example shows a zebra and a tiger camouflaged with the environment. The top-right and bottom-left videos show occluded scenes with elephants and people (top-right), and rabbits (bottom-left). The bottom-right video shows people and a bicycle occluding one another. The proposed method generates time-consistent identities for all objects of interest as seen from these examples. Additional examples are provided as a mp4 video in the supplementary zip file.

Fig. S3 shows a qualitative example from the Youtube-VIS 2021 data, where the proposed method is able to detect the partially hidden parrot across all frames, but Mask2Former [9] fails to do so.

We show more qualitative results on the Cityscapes-VPS dataset for the task of video panoptic segmentation. The

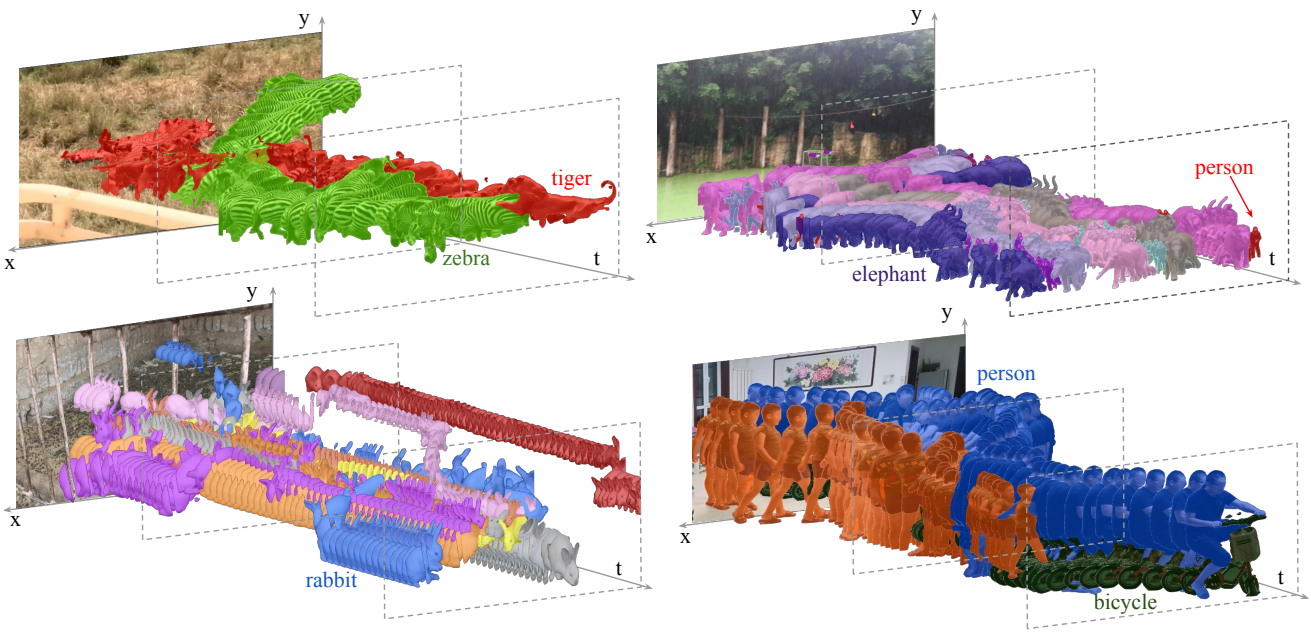

Figure S2. Video instance segmentation on 4 videos from the OVIS data. t represents the time axis, (x, y) represent the image axes. The first video frame is shown in each example for reference. Only the instance, overlaid with a unique color that represents the object ID is shown in subsequent frames. The top-left example shows a zebra and a tiger camouflaged with the environment. The top-right and bottom-left videos show occluded scenes with elephants and people (top-right), and rabbits (bottom-left). The bottom-right video shows people and a bicycle occluding one another.

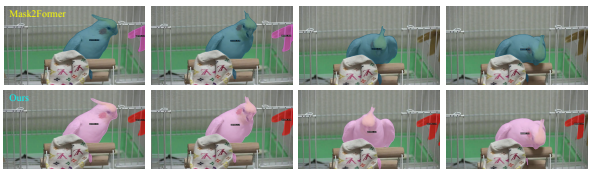

Figure S3. Comparison of Mask2Former [9] (top row) with the proposed method (bottom row) on a video from the Youtube-VIS 2021 data. Mask2Former misses the partially visible parrot (with pink segmentation mask) in the last 2 frames. The proposed method is able to detect this parrot (with red segmentation mask) in all the frames correctly.

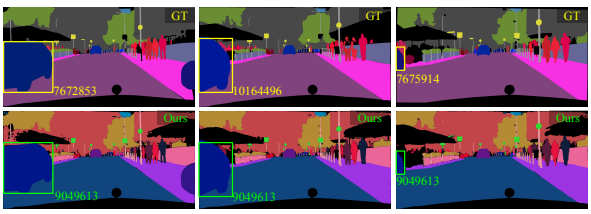

Figure S4. Comparison with the ground truth for the Cityscapes-VPS data. The proposed method (bottom row) generates temporally consistent identities of objects which are sometimes missing from the ground truth (top row).

Method	OVIS AP	OVIS AP_{50}	OVIS AP_{75}	YTVIS 2019 AP	YTVIS 2019 AP_{50}	YTVIS 2019 AP_{75}	YTVIS 2021 AP	YTVIS 2021 AP_{50}	YTVIS 2021 AP_{75}
Ours	**25.8**	**47.9**	**25.4**	**46.7**	**70.4**	**50.9**	**43.3**	**64.9**	**47.1**
Ours (absolute pos.)	25.0	46.5	24.5	46.2	69.9	50.5	42.7	64.1	46.4

Table S1. Effect of relative positional encodings on the VIS datasets.

bottom row of Fig. S4 shows how the proposed method generates consistent IDs over time. It is worth noting that some of the annotations in the Cityscapes-VPS dataset are not temporally consistent. Fig. S4 shows such an example where the ground truth (top row) has inconsistent temporal identities of a car (marked with yellow boxes). The IDs are reported in the figure. In contrast, the proposed method (bottom row) preserves the identities of the car over time (marked with green boxes), improving over the "ground truth."

C.2. Additional Ablation Studies

Sec. 4.2 shows ablation studies to establish the importance of the different components of the proposed approach. We provide some more ablation studies here. Tab. S1 and Tab. S2 summarize these studies.

Effect of Relative Positional Encodings on VIS Datasets. In Tab. 6, we showed how the relative positional encodings

	AP	AP_{50}	AP_{75}
l=0	19.6	43.1	21.3
l=1	**25.8**	**47.9**	25.4
l=2	25.5	47.8	**25.5**
l=3	23.4	44.4	22.6
Ours (w/o ML)	24.1	45.1	22.8
Ours	**25.8**	**47.9**	**25.4**

Table S2. Additional ablation studies (see Sec. C.2).

improve the association accuracy in the MOTS task. We show the effect of relative positional encodings on the VIS task in Tab. S1. "Ours (absolute pos.)" refers to using absolute positional encodings. We observe that the results improve when using relative encodings ("Ours"), although the effects aren't as drastic as the ones reported in Tab. 6.

Number of decoder layers to skip. The first 4 rows in Tab. S2 show the performance change observed if we use all the decoder layers (row 1, $l = 0$), if we skip the first and start from the second layer (row 2, $l = 1$), if we skip the first and second and start from the third layer (row 3, $l = 2$), or if we skip the first 3 layers (row 4, $l = 3$) of the transformer-decoder respectively while performing query vector propagation from the previous frame, as discussed in Sec. 3.2. We observe that skipping the first decoder layer (and starting from $l = 1$) or skipping the first 2 decoder layers (and starting from $l = 2$) achieves comparable results. In all experiments in this paper, we skip only the first decoder-layer and start from $l = 1$.

Effect of Modified Loss. 'Ours (w/o ML)' in Tab. S2 shows results when a standard bipartite matching is used to obtain the best matching between [prediction, ground truth] pairs following [8]. Hence, the loss is not modified as described in Sec. B. Specifically, the ground truth class labels for an object corresponding to the 2 training clips τ_a and τ_b, are always equal. If the object appears in any of the frames in τ_a and τ_b, the ground truth class label is the actual object class, otherwise the class label is $\varnothing$. Since the ground truth class labels corresponding to τ_a and τ_b are always equal, no modification to the standard loss used in [7, 8] is necessary. We observe a performance drop in AP from 25.8 (our approach, last row in Tab. S2) to 24.1 without the modified loss.

D. Implementation Details

In this section, we provide the experimental details, model parameters and the choice of hyper-parameters for all experiments discussed in Sec. 4 and Appendix C. We also discuss the resources and the licenses of the code-bases and datasets used in the paper.

Model-Parameters. Tab. S3 compares the total number of trainable parameters for some of the approaches mentioned in Tab. 1, while using an R50 backbone. We observe that our approach uses a lower number of parameters than IDOL [52], MinVIS [22], and SeqFormer [51]. We use slightly more parameters than Mask2Former-VIS [8], due to the learnable relative positional encodings.

Method	IDOL	MinVIS	SeqFormer	Mask2Former-VIS	Ours
Params.	43.07M	43.96M	48.40M	**42.38M**	42.98M

Table S3. Number of model parameters for different approaches.

Hyper-Parameters. We now discuss the hyper-parameters used in this work. In all experiments, the maximum number of objects (N) in a given video for a R50 backbone is 100, and for a Swin-L backbone is 200. We use a feature dimension C (Sec. 3.1) of 256 in all models. We trained the models with an initial learning rate of 0.0001 and ADAMW [33] optimizer with a weight decay of 0.05. We use a batch size of 8 for $T = 2$. For video instance segmentation, the networks were first initialized with weights from Mask2Former [9] trained on the COCO image instance segmentation dataset. We then fine-tune the model using the training procedure discussed in Sec. B on the respective Youtube-VIS or OVIS datasets for $10,000$ iterations. For KITTI-MOTS and MOTS 2020, we use the same setting for $6,000$ iterations. For the video panoptic segmentation task, we initialize the network with Mask2Former weights trained on Cityscapes image panoptic segmentation. We fine-tune the model for $10,000$ iterations.

Resources. We used 4 NVIDIA A100 and 8 V100 GPUs to run the experiments presented in this paper. Each experiment took roughly 8 GPU hours of training on the A100 GPUs for VIS and VPS and 5 GPU hours for MOTS.

Licenses. Our code is built on Mask2Former [9] which is majorly licensed under the MIT license, with some portions under the Apache-2.0 License. The Youtube-VIS datasets are licensed under a Creative Commons Attribution 4.0 License. The OVIS dataset, Cityscapes-VPS and MOTS datasets are released under the Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) License.