Prototypical Cross-Attention Networks for Multiple Object Tracking and Segmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Multiple object tracking and segmentation requires detecting, tracking, and segmenting objects belonging to a set of given classes. Most approaches only exploit the temporal dimension to address the association problem, while relying on single frame predictions for the segmentation mask itself. We propose **P**rototypical **C**ross-attention **N**etwork (**PCN**), capable of leveraging rich spatio-temporal information for online multiple object tracking and segmentation. PCN first distills a space-time memory into a set of prototypes and then employs cross-attention to retrieve rich information from the past frames. To segment each object, PCN adopts a prototypical appearance module to learn a set of contrastive foreground and background prototypes, which are then propagated over time. Extensive experiments demonstrate that PCN outperforms current video instance tracking and segmentation competition winners on both Youtube-VIS and BDD100K datasets, and shows efficacy to both one-stage and two-stage segmentation frameworks.

1 Introduction

Multiple object tracking and segmentation (MOTS), also known as Video Instance Segmentation (VIS), is an important problem with many real-world applications, including autonomous driving [10, 21] and video analysis [4, 39]. The task involves tracking and segmenting all objects within a video from a given set of semantic classes. We are witnessing rapidly growing research interest on MOTS thanks to the introduction of large scale benchmarks [39, 42, 31]. State-of-the-art methods [39, 5, 31, 24] for MOTS mainly follow the tracking-by-detection paradigm, where objects are first detected and segmented in individual frames and then associated over time.

Although methods based on the popular tracking-by-detection philosophy have shown promising results, temporal modeling is limited to the object association phase [39, 5, 18] and only between two adjacent frames [31, 14]. On the other hand, the temporal dimension carries rich information about the scene. The information encoded in multiple temporal views of an object has the potential of improving the quality of predicted segmentation, localization, and categories. However, effectively and efficiently leveraging the rich temporal information remains a challenge. While sequential modeling has been applied for video processing [34, 35, 9, 23, 12], these methods generally operate directly on the high-resolution deep features, requiring large computational and memory consumption, which greatly limits their use.

We propose a **P**rototypical **C**ross-attention **M**odule, termed **PCM**, to leverage temporal information for multiple object tracking and segmentation. As illustrated in Figure 1, the module first distills spatio-temporal information into condensed prototypes using clustering based on Expectation Maximization. The resulting prototypes, composed of Gaussian Components, yield a rich and generalizable yet compact representation of the past visual features. Given a deep feature embedding of the current frame, PCM then employs prototypical cross-attention to read relevant information from prior frames.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

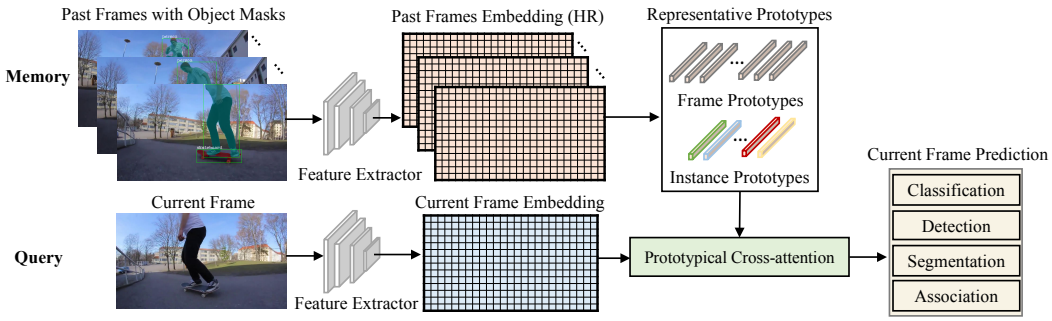

Figure 1: We propose Prototypical Cross-attention Network for MOTS, which first condenses the space-time memory and high-resolution frame embeddings into frame-level and instance-level prototypes. These are then employed to retrieve rich temporal information from past frames by our efficient prototypical cross-attention operation.

Based on the noise-reduced clustered video features information, we further develop a **P**rototypical **C**ross-attention **N**etwork (**PCN**) for MOTS, that integrates the general PCM at two stages in the network: on the frame-level and instance-level. The former reconstructs and aligns temporal past frame features with current frame, while the instance level integrates specific information about each object in the video. For robustness to object appearance change, PCN represents each object instance by learning sets of contrastive foreground and background prototypes, which are propagated in an online manner. With a limited number of prototypes for each instance or frame, PCN efficiently performs long-range feature aggregation and propagation in a video with linear complexity. Consequently, our PCN outperforms standard non-local attention [34] and video transformer [35] on both the large-scale Youtube-VIS and BDD-MOTS benchmarks.

Our main contributions are summarized as follows: (i) We introduce the PCN module for efficiently utilizing long-term spatio-temporal video information. (ii) We develop a MOTS approach that employs PCN on frame and instance-level. (iii) We further represent the appearance of each video tracklet with contrastive foreground and background prototypes, which are propagated over time. (iv) We extensively analyze our approach. Our PCN outperforms previous approaches on the challenging self-driving dataset BDD100k [42] and the semantically diverse YouTube-VIS dataset [39].

2 Related work

Video instance segmentation (VIS) Existing VIS methods [39, 2, 17] widely adapt the two-stage paradigm of Mask R-CNN [11] by adding an additional tracking branch. Thus, their typical pipelines first detect regions of interest (RoIs) and then use the instance features after RoIAlign to regress object mask and associate cross-frame instances. More recent works [5, 14, 18] employ a one-stage instance segmentation method, e.g. the anchor-free FCOS detector [28], which predicts a linear combination of mask bases [3] as its final segmentation. The aforementioned approaches make very limited use of temporal information to enhance the quality of the segmentation, instead relying on single image-based mask prediction, or only model short-term temporal correlation between two consecutive frames [14, 25]. In the context of long-term temporal association, the offline method VisTr [35] adapts vision transformer [6] for VIS, but suffers from a huge computational burden and memory consumption due to the dense pixel-level attention operations over long sequences. Compared to these methods, our PCN temporally aggregates and propagates the prototypical features with both the long-term benefit and linear complexity.

Multiple Object Tracking and Segmentation (MOTS) Similar to VIS, MOTS methods [31, 22, 24] mainly follow the tracking-by-detection paradigm. Objects are first detected and segmented, followed by association between frames. Track R-CNN [31] integrates temporal context feature from two neighboring frames using 3D convolutions. TrackFormer [20] performs joint object detection and tracking by recurrently using Transformers, while Stem-Seg [1] adopts a short 3D convolutional spatio-temporal volume to learn pixel embedding by treating segmentation as a bottom-up grouping. In contrast, our approach clusters appearance features in a long spatio-temporal volume with explicit foreground and background prototypes that are updates online. Besides, the mixture Gaussian

components in instance appearance module equips PCN a stronger modeling ability compared to instance-level average pooling [27, 41] or single Gaussian model [43, 13].

Temporal attention models Video understanding usually requires long-range sequential modeling of relations between spatio-temporal locations. Recently, attention-based approaches, such as non-local attention [34, 33, 23, 12] and transformers [8, 29], have been successfully adopted in video classification and action recognition. These tasks [19, 26, 37] involve dense pixel-level attention, leading to quadratic complexity in the sequence length, thus making them excessively expensive for long sequences. Improved temporal attention models mainly include double attention mechanism [7] on image recognition with global-local decomposition, and clustered attention Transformer [32] for language sequence modeling. Besides, recent prototypical methods [15, 38] use the EM algorithm for single-image semantic segmentation or few-shot learning [27]. Unlike these methods, our PCN uses compact prototypical representation both for temporal feature aggregation and compact instance appearance feature propagation.

3 Method

We propose an approach for Multiple Object Tracking and Segmentation. Given a video sequence, the goal is to detect, track, and segment objects from a predefined set of object categories. Specifically, we consider the online setting, where the predictions only depend on current and past frames.

3.1 Traditional Cross-Attention

To utilize the rich temporal information to improve the segmentation prediction, recent approaches [23, 12] have employed cross-attention. We consider past spatio-temporal information encoded in a memory $\mathbf{M}$, consisting of deep features of size $H \times W \times T \times C$. The memory encapsulates valuable information about the past appearances and predictions of objects and background in a scene. To attend to the memory, the information is first separately embedded into key $\mathbf{k}^M$ and value $\mathbf{v}^M$ feature vectors. The keys are used to address relevant memories whose corresponding values are returned. The standard memory reading process is a non-local operation computed as the weighted sum,

$$y_i = \frac{1}{Z_i} \sum_{j=1}^{H \times W \times T} \exp(\mathbf{k}_i^Q \cdot \mathbf{k}_j^M) \mathbf{v}_j^M \,, \tag{1}$$

where $\mathbf{k}^Q$ denotes query key map, which is predicted from the current frame. Further, i and j are the index of each query and the memory location, and $Z_i = \sum_j \exp(\mathbf{k}_i^Q \cdot \mathbf{k}_j^M)$ is the normalizing factor.

Although proven effective, the standard attention operation (1) is known to suffer from poor computational and memory scaling properties [16]. In particular, since all queries are matched to all keys, it experiences a quadratic scaling $\mathcal{O}((HW)^2)$ of computations in the spatial size HW of the feature map. This is particularly problematic for segmentation tasks, where fine-grained high-resolution information is desired to improve the quality of the predictions.

3.2 Prototypical Cross-Attention

To address the aforementioned limitations of the standard cross-attention, we introduce the prototypical cross-attention. Our approach is based on a clustered memory $\mathbf{M}_c$. We call these clusters prototypes, since they correspond to representative items in the memory. While clustering effectively reduces the number of items in the memory, it also serves to deprecate noisy information, leading to a more generalizable and robust representation of the memory.

To employ an attention mechanism, similar to (1), we require a clustering of the memory that generates a principled continuous and differentiable clustering assignment function. We therefore cluster the keys in the memory by fitting a Gaussian Mixture Model (GMM),

$$p(\mathbf{k}) = \frac{1}{N} \sum_{j=1}^{N} p(\mathbf{k}|z=j) \,, \qquad p(\mathbf{k}|z=j) = \frac{1}{(2\pi\sigma^2)^{\frac{D}{2}}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{k} - \mathbf{k}_j^\mu\|^2\right) \tag{2}$$

Here, N denotes the number of Gaussian mixtures, D is the feature dimension of the keys. We use a constant variance parameter σ^2 and uniform cluster priors $p(z=j) = \frac{1}{N}$, where z denotes the latent

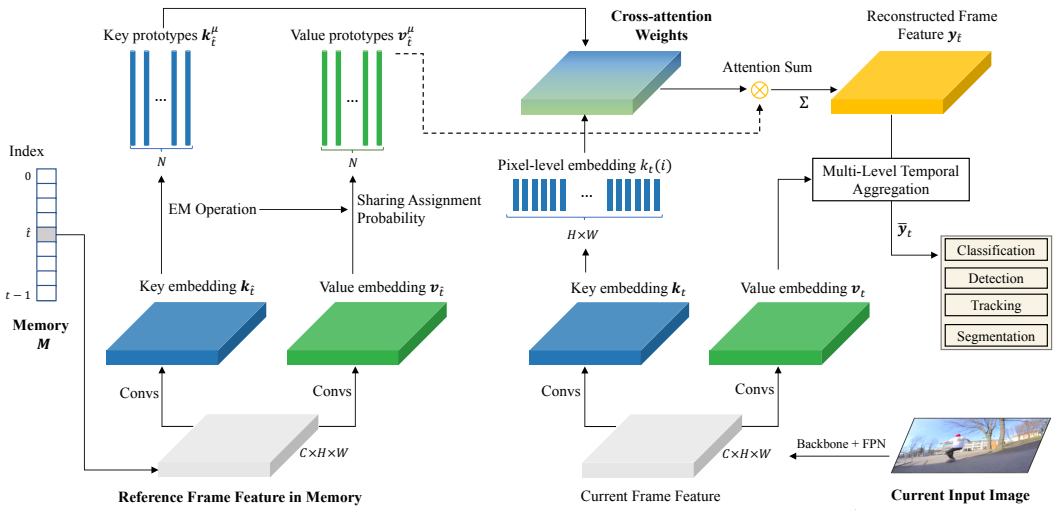

Figure 2: Overview of our frame-level prototypical cross-attention. For a frame $\hat{t}$ in the memory we first perform GMM-based clustering to achieve the key $\mathbf{k}_{\hat{t}j}^{\mu}$ and value $\mathbf{v}_{\hat{t}j}^{\mu}$ prototypes. Given the key encoding $\mathbf{k}_t$ of the current frame, we attend to the prototypes to generate the reconstructed feature $\mathbf{y}_{\hat{t}}$, which are then aggregated temporally and fused with the current value encoding $\mathbf{v}_t$.

cluster assignment variable. The component means $\mathbf{k}^{\mu}$ represent the prototype keys in the memory. We generate the clustering (2) using the standard Expectation-Maximization algorithm.

The GMM allows us to compute a soft cluster assignment by evaluating the posterior probability of the latent assignment variable z. Using Bayes rule, the probability of a key value $\mathbf{k}$ to be assigned to the jth prototype is derived as,

$$p(z=j|\mathbf{k}) = \frac{p(\mathbf{k}|z=j)p(z=j)}{\sum_{l=1}^{N} p(\mathbf{k}|z=l)p(z=l)} = \frac{\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}-\mathbf{k}_j^{\mu}\|^2\right)}{\sum_{l=1}^{N}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}-\mathbf{k}_l^{\mu}\|^2\right)} \,. \tag{3}$$

The resulting cluster assignment can thus be written as a SoftMax operation, where the corresponding logits are provided by the negative cluster distance $\|\mathbf{k}-\mathbf{k}_j^{\mu}\|^2$ scaled with a temperature of $2\sigma^2$.

Since the clustering is performed in the key space of the memory, we next retrieve the corresponding value prototypes. To this end, we employ the key cluster assignment probabilities in (3) to compute the values for each memory prototype,

$$\mathbf{v}_j^{\mu} = \sum_{l=1}^{N} p(z=j|\mathbf{k}_l^M)\mathbf{v}_l^M \,. \tag{4}$$

For attending to our clustered memory, we first predict the key encodings $\mathbf{k}_i^Q$ of the query image. We then read from the clustered memory by computing the average over the value prototypes $\mathbf{v}_j^{\mu}$, weighted with the cluster assignment probabilities,

$$\mathbf{y}_i = \sum_{j=1}^{N} p(z=j|\mathbf{k}_i^Q)\mathbf{v}_j^{\mu} = \frac{1}{Z_i}\sum_{j=1}^{N}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}_i^Q-\mathbf{k}_j^{\mu}\|^2\right)\mathbf{v}_j^{\mu} \,. \tag{5}$$

The final attention operation has much similarity with the original dot-product cross attention (1). Note that the key-query similarity in our approach is measured by Euclidian distance instead of a dot-product. Importantly, our formulation (5) attends to a reduced set of N prototypes, while the original attention (1) requires attending to the full spatio-temporal memory of size $H \times W \times T$.

3.3 Prototypical Cross-attention Network

Here, we propose the Prototypical Cross-attention Network (PCN) for MOTS by integrating our prototypical cross-attention module into both the frame-level and instance-level. The former aims to align and aggregate temporal frame features stored in memory, while the latter is for propagating the

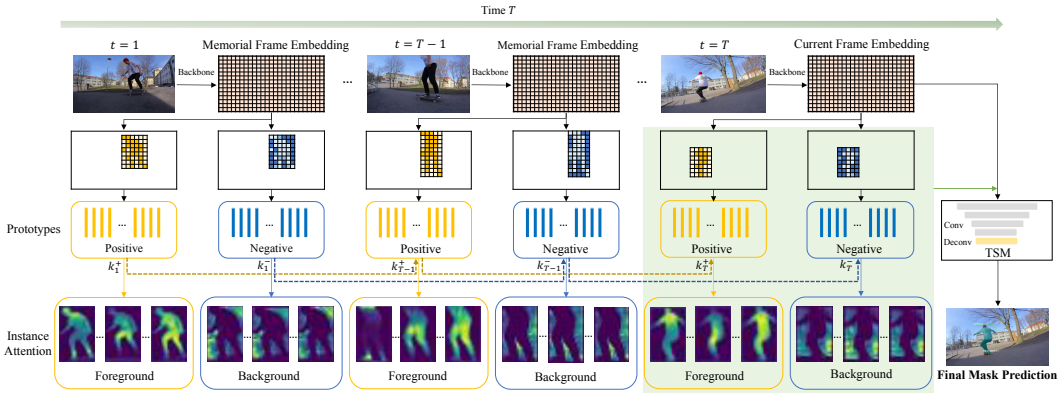

Figure 3: Our instance-level prototypical attention with foreground and background prototypes and temporal propagation. The foreground/background attention maps from (bottom) demonstrate the localized and discriminative appearance representation. Temporal Segmentation Module (TSM) takes the current frame, initial mask, and instance attention maps as input and generates the final mask.

instance appearance features over time and produce instance cross-attention maps to help segmentation. Besides, we also design a prototypical instance appearance module to represent each video tracklet with contrastive mixture foreground and background prototypes.

3.3.1 Frame-level Prototypical Cross-attention

In Figure 2, prototypical cross-attention first produces prototypes by fitting a Gaussian mixtures model (2) to the feature in the memory. To provide further flexibility when dynamically updating the memory $\mathbf{M}$, we first perform frame-wise clustering for each reference frame feature at index $\hat{t}$ to compute the N key prototypes $\{\mathbf{k}^{\mu}_{\hat{t}i}\}_{j=1}^{N}$, and retrieve the corresponding value embeddings $\{\mathbf{v}^{\mu}_{\hat{t}j}\}_{j=1}^{N}$ using (4) for each memory frame $\hat{t}$ independently. The key and value features are predicted using two parallel convolutional layers.

Frame-wise prototypical memory attention Given the query key encoding $\mathbf{k}^{Q}_{ti}$ of the current frame t, we perform prototypical cross-attention to each memory frame $\hat{t}$ independently using our formulation (3) as,

$$\mathbf{y}_{\hat{t}i} = \frac{1}{Z_{\hat{t}i}} \sum_{j=1}^{N} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}_{ti} - \mathbf{k}^{\mu}_{\hat{t}j}\|^2\right) \mathbf{v}^{\mu}_{\hat{t}j}\,, \qquad Z_{\hat{t}i} = \sum_{l=1}^{N} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{k}^{Q}_{ti} - \mathbf{k}^{\mu}_{\hat{t}l}\|^2\right)\,. \tag{6}$$

Note that the index i refers to a spatial coordinate in the current frame. The resulting feature map $\mathbf{y}_{\hat{t}}$ can intuitively be seen as a projection of features from frame $\hat{t}$ to the current frame. This projection essentially aligns the condensed feature information in frame $\hat{t}$ with the current frame.

Temporal feature aggregation Since frame-wise attention does not fuse temporal information, we perform a temporal aggregation. The temporal information $\mathbf{y}_{\hat{t}}$ in (6) from different frames $\hat{t}$ are fused as a linear combination, weighted by the feature similarity with the current frame. Specifically, the temporally aggregated representation is obtained as

$$\bar{\mathbf{y}}_{ti} = \sum_{\hat{t}=1}^{t} w_{\hat{t}i}\mathbf{y}_{\hat{t}i}\,, \qquad w_{\hat{t}i} = \frac{\exp(\mathbf{y}_{ti}\cdot\mathbf{y}_{\hat{t}i})}{\sum_{s=1}^{t}\exp(\mathbf{y}_{ti}\cdot\mathbf{y}_{si})}\,. \tag{7}$$

Note that $\hat{t} = t$ in the sum refers to the value embedding $\mathbf{y}_{ti} = \mathbf{v}^{Q}_{ti}$ extracted from the current frame. The contribution of each frame $\hat{t}$ is thus weighted by the similarity to this current frame prediction using the attention weights $w_{\hat{t}i}$. This strategy ensures that incorrect or dissimilar regions are suppressed when computing the final aggregated feature embedding $\bar{\mathbf{y}}_t$. To handle object with large-scale variation and produce more fine-grained instance mask prediction, we further extend temporal aggregation to multi-level using different levels of the extracted FPN features, as detailed in the supplementary material.

3.3.2 Instance-level Prototypical Cross-attention

Contrastive foreground and background representation In additional to the condensed frame-level representation, for more accurate segmentation results, we further encode each tracked object with compact and robust appearance prototypes. To further empower our proposed attention mechanism, we utilize the initially detected object mask to identify each foreground instance. We then separately model the extracted foreground and background features using a GMM (2). We denote the resulting foreground prototypes as $\mathbf{k}_{tj}^{+}$ and background prototypes as $\mathbf{k}_{tj}^{-}$. The former thus focuses on the appearance of the specific object, creating a rich and dynamic appearance model. When employed in our prototypical cross-attention framework (Section 3.2), it provides fine-grained attention from localized prototypes that naturally learn to focus specific parts of views of the object, as visualized in Fig. 3. Furthermore, the background prototypes $\mathbf{k}_{tj}^{-}$ capture valuable information about the background appearance, which can greatly alleviate the segmentation process. For each object instance we attend to the foreground and background prototypes separately using (3). The results are concatenated together with the initial mask detection to the Temporal Segmentation Head (TSM) for final prediction, as illustrated in Figure 3.

Tracklet feature propagation and updating To effectively model the object appearance change and preserve the most relevant information, we design a recurrent instance appearance updating scheme. From the first video frame where object appears, the accumulated prototypes $\bar{\mathbf{k}}_{tj}^{+}$, $\bar{\mathbf{k}}_{tj}^{-}$ for the instance are propagated to the subsequent frames and updated with new appearance prototypes $\mathbf{k}_{tj}^{+}$, $\mathbf{k}_{tj}^{-}$ using an update rate λ as,

$$\bar{\mathbf{k}}_{tj}^{+} = (1-\lambda)\bar{\mathbf{k}}_{t-1,j}^{+} + \lambda \mathbf{k}_{tj}^{+}\,, \qquad \bar{\mathbf{k}}_{tj}^{-} = (1-\lambda)\bar{\mathbf{k}}_{t-1,j}^{-} + \lambda \mathbf{k}_{tj}^{-}\,. \tag{8}$$

Figure 3 also reveals the consistency of the attended region of a specific prototype j.

4 Experiments

Here, we present comprehensive evaluation and analysis of our approach. Experiments are performed on two large scale datasets, namely YouTube-VIS [39] and BDD100K [42]. Code and models will be released upon publication.

4.1 Experiment setup

Youtube-VIS YouTube-VIS [39] dataset contains 2,883 high quality videos with 131k annotated object instances belonging to 40 diverse categories. The task is to simultaneously classifying, segment and track object instances belonging to these categories. The evaluation metrics for this task are an adaptation of the Average Precision (AP) and Average Recall (AR) of image instance segmentation.

BDD100K We also evaluate on the large-scale tracking and segmentation dataset of BDD100K [42], which is a challenging self-driving dataset with 154 videos (30,817 images) for training, 32 videos (6,475 images) for validation, and 37 videos (7,484 images) for testing. The dataset provides 8 annotated categories for evaluation, where the images in the tracking set are annotated per 5 FPS with 30 FPS frame rate. We adopt the well-established MOTS metrics [31] to our task.

Implementation details We implement PCN based on two different existing MOTS approaches. For Youtube-VIS, we adopt ResNet with FPN pre-trained on COCO as the backbone, and build our segmentation tracker on the one-stage segmentation model [5]. Both the instance and frame cross-attention is built on the extracted FPN features. Our model is trained with initial learning rate 0.0025 on 4 GPUs using SGD, and executes with a speed of 15.0 FPS on ResNet-50. Similar to [39, 18, 14], we use the input size 360×640 for training. On BDD100K, we build PCN by extending the two-stage MOT method [24] with our temporal segmentation modules. We follow the same training strategy of QDTrack-mots [24]. More details can be found in supplemental material.

4.2 State-of-the-Art Comparison

We compare our approach with the state-of-the-art methods on the aforementioned large-scale MOTS/VIS benchmarks Youtube-VIS and BDD100K, where PCN outperforms all existing methods without bells and whistles, and shows efficacy to both one-stage and two-stage segmentation frameworks. We follow the official metrics of each benchmark to evaluate our model.

Table 1: Comparison with state-of-the-art on the YouTube-VIS validation set. Results are reported in terms of mask accuracy (AP) and recall (AR). Asterisks * denote concurrent works on arXiv.

Method	Backbone	Type	Online	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
VisTr* [35]	ResNet-50	Transformer	×	34.4	55.7	36.5	33.5	38.9
OSMN [40]	ResNet-50	Two-stage	✓	23.4	36.5	25.7	28.9	31.1
FEELVOS [30]	ResNet-50	Two-stage	✓	26.9	42.0	29.7	29.9	33.4
DeepSORT [36]	ResNet-50	Two-stage	✓	26.1	42.9	26.1	27.8	31.3
MaskTrack R-CNN [39]	ResNet-50	Two-stage	✓	30.3	51.1	32.6	31.0	35.5
STEm-Seg [1]	ResNet-50	One-stage	×	30.6	50.7	33.5	31.6	37.1
SipMask [5]	ResNet-50	One-stage	✓	32.5	53.0	33.3	33.5	38.9
STMask* [14]	ResNet-50	One-stage	✓	33.5	52.1	36.9	31.1	39.2
SG-Net* [18]	ResNet-50	One-stage	✓	34.8	**56.1**	36.8	35.8	40.8
PCN (Ours)	ResNet-50	One-stage	✓	**36.1**	54.9	**39.4**	**36.3**	**41.6**
STMask* [14]	ResNet-101	One-stage	✓	36.3	55.2	39.9	33.7	42.0
SG-Net* [18]	ResNet-101	One-stage	✓	36.3	57.1	39.6	35.9	43.0
PCN (Ours)	ResNet-101	One-stage	✓	**37.6**	**57.2**	**41.3**	**37.2**	**43.9**

Table 2: State-of-the-art comparison on the BDD100K segmentation tracking validation set. I: ImageNet. C: COCO. S: Cityscapes. B: BDD100K. "-fix" means adopting the pretrained model from the BDD100K tracking set, fixing the existing parts, and only training the added mask head.

Method	Pretrained	Online	mMOTSA↑	mMOTSP↑	mIDF↑	ID sw.↓	mAP↑
SortIoU	I, C, S	✓	10.3	59.9	21.8	15951	22.2
MaskTrackRCNN [30]	I, C, S	✓	12.3	59.9	26.2	9116	22.0
STEm-Seg [1]	I, C, S	×	12.2	58.2	25.4	8732	21.8
QDTrack-mots [24]	I, C, S	✓	22.5	59.6	40.8	1340	22.4
QDTrack-mots-fix [24]	I, B	✓	23.5	66.3	44.5	973	25.5
PCN (Ours)	I, B	✓	**27.4**	**66.7**	**45.1**	**876**	**26.6**

Youtube-VIS The results of Youtube-VIS benchmark is in Table 1, where PCN achieves the best mask AP of 36.1% using ResNet-50 and 37.6% using ResNet-101 respectively, while being an online method. Our approach consistently surpasses most recent SOTA methods, including STMask [14] and SG-Net [18] by a significant margin. These methods only conduct temporal modeling between two adjacent frames for feature correlation. Compared to our baseline SipMask [5], a single-image based segmentation with object centerness association, PCN improves the mask AP from 32.5% to 36.1%, which shows the effectiveness of long-term temporal modeling in helping object tracking and segmentation.

BDD100K Table 2 shows our results on BDD100K tracking and segmentation benchmark, where PCN outperforms the strong baseline methods MaskTrackRCNN [39] and QDTrack-mots [24]. Our approach achieves a large advantage in mMOTSA, with over 3 points gain and around 10% ID switches decrease. MOTSA measures segmentation as well as tracking quality, while ID Switches can measure the performance of identity consistency. The significant advancements demonstrate that our method with prototypical cross-attention enables more accurate pixel-wise object tracking by effectively exploiting temporal information.

4.3 Ablation study and analysis

We conduct detailed ablation studies on Youtube-VIS validation set, where we investigate the effect of our proposed prototypical cross-attention components for MOTS during training and testing.

Effect of frame-level prototypical cross-attention module To study the importance of temporal information amount, we conduct an ablation study on models with different input temporal window lengths in Table 3. A temporal length of 1 thus means that no prior temporal information guidance is used during video instance segmentation. By varying the frame length from 1 to 32, the mask AP increases from 32.5% to 35.4%, which reveals that richer temporal information with multiple views of a segmented object indeed brings more gain to model performance.

Table 3: Results of varying temporal memory length in our PCN on YouTube-VIS.

Length	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
1	32.5	53.0	33.3	33.5	38.9
2	33.7	53.8	35.3	33.9	39.5
4	33.9	**54.0**	36.8	34.1	40.0
8	34.2	53.7	37.6	34.4	40.3
16	34.6	53.7	38.3	35.4	40.5
32	**35.4**	53.8	**39.1**	**35.9**	**41.0**

Table 4: Effect of multi-layer prototypical feature fusion with tube length 4 on YouTube-VIS.

FPN Layer	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
P3	30.8	51.7	32.0	32.6	37.0
P4	32.0	51.5	34.1	32.6	37.2
P5	32.9	52.1	35.9	33.2	38.6
P3-P4	33.1	52.3	35.6	33.6	38.5
P3-P5	**33.9**	**54.0**	**36.8**	**34.1**	**40.0**

Table 5: Comparison with non-local attention [33] and transformer [6, 35] on YouTube-VIS.

Length	Prototypical Cross-attention			Non-local Attention			Transformer (Multi-Head Self-Attention)		
	AP	FLOPs(B)	Memory(M)	AP	FLOPs(B)	Memory(M)	AP	FLOPs(B)	Memory(M)
2	33.7	5.8	323	33.2	24.3	2497	24.6	103.8	5321
4	33.9	12.0	652	33.3	49.1	4763	25.8	387.2	9844
8	**34.2**	23.7	1419	33.6	99.6	9631	28.3	1413.3	18762

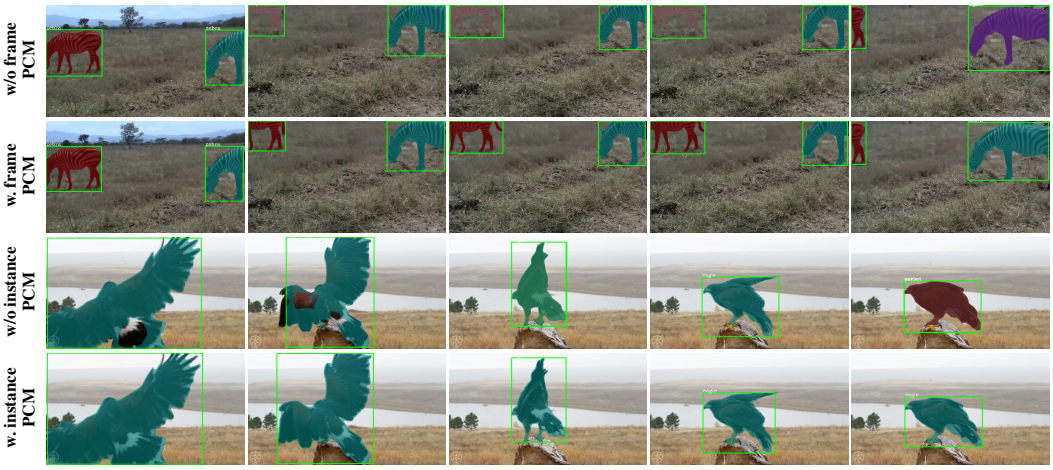

Figure 4: Qualitative impact of our PCM on YouTube-VIS. Mask colors encode object identity. Our frame-level PCM (second row) helps provide consistent detections and preserve identities compared to the baseline (first row). The instance-level PCM (fourth row) provides more accurate masks, while further improving identity consistency compared to not employing our module (third row).

Effect of multi-layer temporal aggregation Since we perform temporal feature aggregation on the extracted FPN features, to help deal with objects with partial occlusion and large-scale variation, we also study the effect of using different levels of the extracted FPN features. In Table 4, we select the FPN feature map from P3-P5 layers for (excluding P6 and P7 due to impractical computation cost), and perform prototypical temporal aggregation on each FPN layer. We find that multi-layer information is also important to final model performance.

Computation and memory efficiency In Table 5 we analyze different attention mechanisms. Compared to standard space-time memory reading using non-local attention [33, 23] or recent popular transformer [35, 6] with multi-head self-attention layer, the prototypical cross-attention with condensed prototypes not only enjoys high accuracy advantage, but also largely reduces the memory consumption and computation amount. For input tube length 8, the prototypical memory consumption is less than 10% of the transformer with negligible FLOPs computation due to the small number of representative prototypes in (5).

Effect of instance-level prototypical appearance module We analyze the instance-level prototypical cross-attention module, which represents each video tracklet using the contrastive prototypes. In Table 6, we study the influence of instance prototype number and the effect of foreground-background contrasting. Using both positive and negative prototypes improves AP from 32.5% to 33.9%. Compared to the single prototype representation, the GMM demonstrate a stronger appearance modeling ability. We further find that the performance saturates when the number is larger than 60. In the supplementary file we provide additional instance cross-attention maps visualization to highlight the various attended regions, which also reveal the implicit unsupervised temporal consistency for prototype focusing specific object region over time.

Table 6: Ablation study on instance-level prototypes on YouTube-VIS.

Pos. Proto. Number	Neg. Proto. Number	AP	AP_{50}
0	0	32.5	53.0
1	0	32.4	52.3
0	1	32.1	52.4
1	1	32.7	52.8
5	5	33.1	53.6
30	30	**33.9**	**54.1**
50	50	33.6	53.8

Table 7: Ablation on instance-level EM feature propagation and updating on YouTube-VIS.

version	AP	AP_{50}
No instance prototype propagation	33.5	53.2
Using initial instance prototype	33.0	52.8
Update momentum = 0.2	**34.3**	**53.8**
Update momentum = 0.5	34.0	53.6

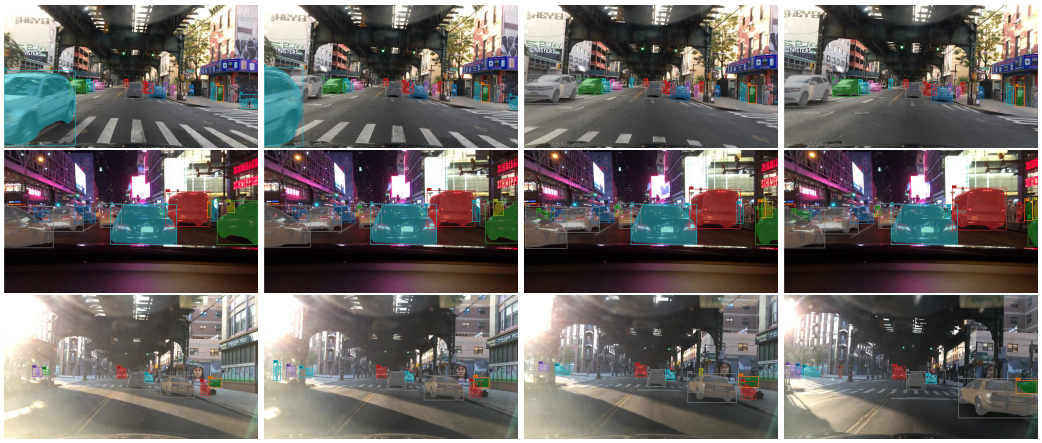

Figure 5: Qualitative results of our method on BDD100K. PCN produces robust tracking and segmentation results under large motion and appearance changes (1st row) and heavy traffic in low-light conditions (2nd row). In the 3rd row, PCN misses a detection (the person to the left in 1st frame), and produces tracking errors (2nd frame) when it covers totally different regions of the car with low appearance similarity. Zoom for better view. Video results are in the suppl. file.

In Table 7, we investigate the effectiveness of instance prototype (including the both positive and negative ones) propagation in an online manner, and compared it with using the instance prototype in the initial frame or current frame. We find that updating object prototypes recurrently with a momentum of 0.2 improves video segmentation AP of 1.3%.

Qualitative analysis In Figure 4, we showcase qualitative ablation results of PCN on Youtube-VIS. Compared to the baseline, we see that our model results in more consistent segmentation and better tracking using prototypical cross-attention module. We also provide visual results on BDD100K in Figure 5, where PCN produces robust tracking and segmentation results even under large object appearance change (first row) or low illumination (second row). In the 3rd row, PCN has limitations in handling missing detections (the person in the first frame) with limited appearance information under extreme lighting, and produce tracking errors in the second frame when visible parts of the same car is totally different across frame and with low appearance similarity.

Societal impact PCN has high potential impact in important applications, such as transportation, sports analysis, and self-driving vehicles. However, this powerful technology can be deployed in human monitoring and surveillance as well which raise ethical and privacy issues. Potential negative impact can be avoided by enforcing a strict and secure data privacy regulation such as the GDPR, proper technology management education, and having an open dialogue among various stakeholders on how such technology should be deployed and regulated.

5 Conclusion

We present PCN, a new online method for MOTS. PCN first distills the space-time memory into a set of frame-level and instance-level prototypes, followed by cross-attention to retrieve rich information from the past frames. In contrast to most previous MOTS methods with limited temporal consideration, PCN efficiently performs long-term temporal propagation and aggregation, and achieves large performance gain on the two largest MOTS benchmarks with low computation and memory cost. We validate the efficacy of PCN on both the existing one-stage and two-stage trackers. We believe PCN will significantly benefit more video understanding tasks in the future.

References

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

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

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019.

[4] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[5] 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 *ECCV*, 2020.

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

[7] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. a^2-nets: Double attention networks. In *NeurIPS*, 2018.

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

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[12] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *CVPR*, 2020.

[13] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *CVPR*, 2019.

[14] Minghan Li, Shuai Li, Lida Li, and Lei Zhang. Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation. In *CVPR*, 2021.

[15] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *ICCV*, 2019.

[16] Yingwei Li, Xiaojie Jin, Jieru Mei, Xiaochen Lian, Linjie Yang, Cihang Xie, Qihang Yu, Yuyin Zhou, Song Bai, and Alan L Yuille. Neural architecture search for lightweight non-local networks. In *CVPR*, 2020.

[17] Chung-Ching Lin, Ying Hung, Rogerio Feris, and Linglin He. Video instance segmentation tracking with a modified vae architecture. In *CVPR*, 2020.

[18] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *CVPR*, 2021.

[19] Xiankai Lu, Wenguan Wang, Danelljan Martin, Tianfei Zhou, Jianbing Shen, and Van Gool Luc. Video object segmentation with episodic graph memory networks. In *ECCV*, 2020.

[20] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[21] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[22] Anton Milan, Laura Leal-Taixé, Konrad Schindler, and Ian Reid. Joint tracking and segmentation of multiple targets. In *CVPR*, 2015.

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

[24] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, 2021.

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

[26] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020.

[27] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[28] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

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

[31] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019.

[32] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. *NeurIPS*, 2020.

[33] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[34] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

[35] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503v1*, 2020.

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

[37] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *CVPR*, 2021.

[38] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *ECCV*, 2020.

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

[40] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018.

[41] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020.

[42] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020.

[43] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4.3.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 4.3.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 3.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] But the code and trained models will be released upon publication.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please refer to Section 4.1 and supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Following prior work in this field, error bars are not reported because it would be too computationally expensive.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please refer to supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] Please refer to the supplementary material for the license and details of the employed datasets and code.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] Code and models will be released upon publication.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Please refer to the supplementary material for more details on the datasets.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Please refer to the supplementary material for more details on the datasets.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing or conduct research with human subjects.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We did not use crowdsourcing or conduct research with human subjects.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not use crowdsourcing or conduct research with human subjects.