Slot-VPS: Object-centric Representation Learning for Video Panoptic Segmentation

Yi Zhou[1], Hui Zhang[1], Hana Lee[2], Shuyang Sun[3], Pingjun Li[1], Yangguang Zhu[1],
ByungIn Yoo[2], Xiaojuan Qi[4*], Jae-Joon Han[2*]
[1]Samsung Research China - Beijing (SRC-B)
[2]Samsung Advanced Institute of Technology (SAIT), South Korea
[3]University of Oxford [4]The University of Hong Kong
{yi0813.zhou, hui123.zhang, hana.hn.lee, byungin.yoo, jae-joon.han}@samsung.com
kevinsun@robots.ox.ac.uk, xjqi@eee.hku.hk

Abstract

*Video Panoptic Segmentation (VPS) aims at assigning a class label to each pixel, uniquely segmenting and identifying all object instances consistently across all frames. Classic solutions usually decompose the VPS task into several sub-tasks and utilize multiple surrogates (*e.g. *boxes and masks, centers and offsets) to represent objects. However, this divide-and-conquer strategy requires complex post-processing in both spatial and temporal domains and is vulnerable to failures from surrogate tasks. In this paper, inspired by object-centric learning which learns compact and robust object representations, we present Slot-VPS, the first end-to-end framework for this task. We encode all panoptic entities in a video, including both foreground instances and background semantics, with a unified representation called panoptic slots. The coherent spatio-temporal object's information is retrieved and encoded into the panoptic slots by the proposed Video Panoptic Retriever, enabling to localize, segment, differentiate, and associate objects in a unified manner. Finally, the output panoptic slots can be directly converted into the class, mask, and object ID of panoptic objects in the video. We conduct extensive ablation studies and demonstrate the effectiveness of our approach on two benchmark datasets, Cityscapes-VPS (val and test sets) and VIPER (val set), achieving new state-of-the-art performance of* 63.7, 63.3 *and* 56.2 *VPQ, respectively.*

1. Introduction

Video panoptic segmentation (VPS) [17, 35, 47] aims at classifying all foreground instances (*things*), *e.g.* cars, people, *etc.*, and countless background semantics (*stuff*), *e.g.* sky, road, *etc.*, segmenting and tracking all object instances consistently across all frames. It is beneficial to many high-level video understanding tasks, such as Video Question Answering [33] and Video Captioning [50], and various real-world applications, such as autonomous driving and video editing.

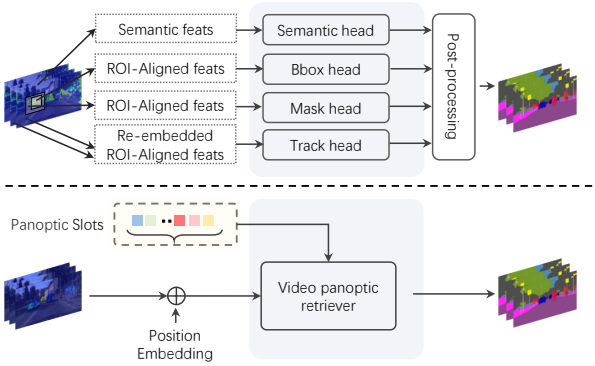

Figure 1. **Comparison between previous works [17, 42] and the proposed Slot-VPS.** VPSNet represents objects with multiple representations, relies on several sub-networks, and requires complex post-processing (*e.g.* NMS, things-stuff fusion, similarity score fusion for tracking), while we introduce panoptic slots to uniformly represent panoptic objects (*i.e.* things and stuff) in a video, enabling a unified end-to-end framework. Code will be made publicly available at https://github.com/SAITPublic/SlotVPS.

Existing methods [17, 35, 47] model things and stuff in a video separately with several sub-networks (as shown in Figure 1) tailored to different sub-tasks including semantic segmentation [6, 30], instance segmentation [2, 14, 21], and tracking [38, 46]. Complicated post-processing in both spatial (*e.g.* things-stuff fusion) and temporal (*e.g.* similarity scores fusion for instance association) domains is needed to

*Corresponding author.

fuse predictions of different subtasks into final VPS results. However, such a decomposed pipeline suffers from several issues. First, the complicated post-processing is time-consuming and requires manual parameter tuning, which is likely to produce sub-optimal results. Second, erroneous predictions from different branches will adversely affect each other and harm the overall performance. For example, the inaccurate boxes will also lead to incomplete segmentation masks, and missing centers will deteriorate the temporal tracking results, which can barely be corrected by post-processing. Third, end-to-end training is blocked, potentially hindering the model from learning features directly optimized for the VPS task.

To solve the aforementioned problems, motivated by object-centric representation learning which learns the compact and robust representations of objects, we introduce a unified end-to-end framework, Slot-VPS, as illustrated in Figure 1. All *panoptic objects* (including both stuff and things) in the video are represented as a unified representation named *panoptic slots*. Panoptic slots are a set of learnable parameters and can be updated through interacting with features extracted from videos. Each panoptic slot is responsible for a stuff class or an object instance in the video, enabling the direct predictions of the class, mask, and object ID of each panoptic object in an end-to-end fashion.

To encode the spatio-temporal information of video-level panoptic objects into the panoptic slots, we introduce the Video Panoptic Retriever (VPR). VPR incorporates a Panoptic Retriever to retrieve the location and appearance information from the spatial features for panoptic object localization and segmentation, and a Video Retriever to correlate slots across different time steps for temporally associating object instances. Furthermore, during the above process, softmax-based operation, which normalizes the contributing weights of each slot, is performed to encourage the panoptic slots to compete and be distinct from each other such that the redundancy among slots will be suppressed. Finally, the spatio-temporal coherent panoptic slots, carrying both object's spatial information and temporal identification information, can be utilized to directly predict final results, *i.e*. class, mask, and object ID of panoptic objects in the video.

To our best knowledge, this is the first fully unified end-to-end framework for the VPS task. It does not rely on any surrogates in both spatial and temporal domains and hence bypasses the drawbacks of dependence on complex post-processing and influence from sub tasks' failures.

Experimental results on the Cityscapes-VPS [17] and VIPER [17] datasets demonstrate the effectiveness of our method. Thanks to the unified end-to-end framework and the object-centric learning, our method outperforms the state-of-the-art [17, 35] on the *val* and test sets (63.7, 63.3 VPQ) of Cityscapes-VPS, and the *val* set (56.2 VPQ) of VIPER with better efficiency.

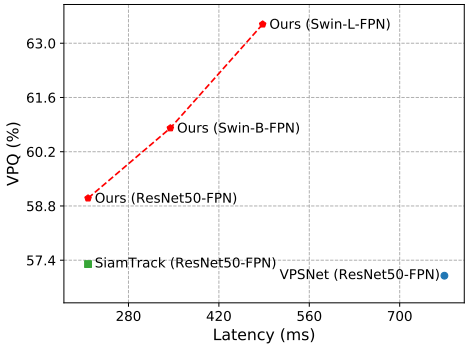

Figure 2. **Speed-Accuracy trade-off curve on the Cityscapes-VPS *val* set.** The latency is measured on V100 GPU.

Our main contributions can be summarized as follows:

- We propose to uniformly represent all panoptic objects in the video with a unified representation (*i.e*. panoptic slots), and introduce Slot-VPS, the first unified end-to-end pipeline for the VPS task.
- To spatially localize, segment, differentiate and temporally associate objects, Video Panoptic Retriever (VPR) is developed to retrieve and encode spatio-temporal coherent objects' information into panoptic slots.
- Our method outperforms the state-of-the-arts [17, 35] on both Cityscapes-VPS and VIPER datasets. What's more, as shown in Figure 2, our model has better efficiency.

2. Related Work

Panoptic Segmentation (PS). Unifying semantic and instance segmentation on the image level, the PS task [7, 18–20, 22, 48] requires assigning a class label for all pixels and uniquely segmenting all object instances. Early attempts [19] in PS follow the decomposition pipeline, separately predicting the semantic and instance segmentation results and then adopting the things-stuff fusion process in later stages. Several works try to simplify the process and improve the accuracy through replacing the post things-stuff fusion with the parameter-free [48] or trainable [23] panoptic head. Furthermore, more researchers [24, 39] try to abandon the separated branches and build an end-to-end unified framework. A recent trend in PS task [3, 8, 43] is regarding this task as the set prediction problem and try to construct the concise end-to-end networks with the help of transformers. However, note that all these unified methods only consider the unification of things and stuff on the spatial domain, excluding the temporal domain.

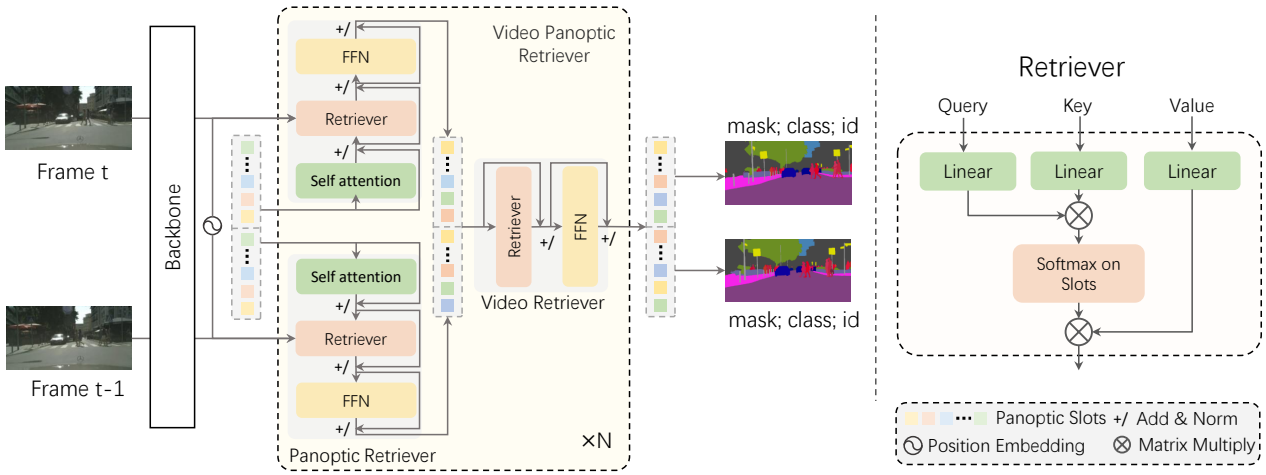

Figure 3. **Overview of the Slot-VPS.** Take two frames (t and t-1) as an example, the multi-scale features extracted from the backbone with position embedding and panoptic slots are fed into Video Panoptic Retriever (VPR) modules for N stages to generate the spatio-temporal coherent panoptic slots. Panoptic slots, representing the panoptic objects in the video, are shared among all frames initially. Finally they are directly converted into objects' masks, classes and IDs. Note that FFN stands for Feed Forward Network.

Video Panoptic Segmentation (VPS). As a direct extension of the PS task into video domain, previous approaches in the VPS task [17, 35] usually apply the image-level approach [7, 48] to each video frame and try to associate the result of all frames by using an extra temporal association head (*e.g.* track head [17, 47], temporal center regression head [35]), *etc*. However, all these methods represent objects with multiple representations (*e.g.* boxes and masks, centers and offsets), requiring multiple separate networks to handle sub-tasks of VPS and complex post-processing (*e.g.* NMS, things-stuff fusion, similarity fusion for tracking) on both spatial and temporal domains. To the best of our knowledge, this is the first time that the unification of things and stuff in both spatial and temporal domains are discussed simultaneously for this task.

Transformers. Inspired by the success of Transformers [40] in Natural Language Processing (NLP) tasks, numerous researches have also been investigated in the computer vision community, *e.g.* Object Detection [3, 52], Panoptic Segmentation [8, 43], and Video Instance Segmentation [26, 45, 49], *etc*. In these pipelines, objects are represented as a set of vectors containing all related information such as location, appearance, *etc*. Attention in transformers is responsible for localizing and segmenting objects, and the bipartite-matching based mechanisms help separate different objects. With the help of these technologies, above tasks can be converted into direct set prediction problem without many hand-designed components. Different from all these works, we introduce the slots competing mechanism into the learning process to enhance the discriminability of objects in both spatial and temporal domains. Jointly representing stuff and things on the video level with panoptic slots, we propose the fully unified end-to-end framework in which all operations in both spatial and temporal domains are fulfilled based on the panoptic slots.

Object-centric learning. Object-centric representation learning [1, 12, 13, 27, 29] mainly focuses on learning robust, generalizable object representations in various scenarios such as unsupervised object discovery, novel-viewpoint prediction, *etc*. In [29], Francesco *et al*. propose the Slot Attention to predict a set of task-dependent abstract representations called *slots* from the perceptual representations such as the output of a convolutional neural network. Slots are exchangeable and can bind to any object in the input. Different from the attention in transformer, the Slot Attention lets the randomly sampled slots compete with each other in the iterative learning process, which will further help the discrimination of objects. However, the Slot Attention is mainly applied to synthetic dataset scenarios, assuming a normal distribution for the slots. In this paper, motivated by the object-centric representations and the competing mechanism, we introduce the panoptic slots and the VPR module to successfully bring object-centric representation learning to real-world data, and set up new state-of-the-art for the VPS task on two datasets.

3. Method

3.1. Model Architecture

As shown in Figure 3, the Slot-VPS framework consists of a backbone (including Resnet50 [15], FPN [25], several deformable convolutions [11, 48]) for extracting multi-scale

features and Video Panoptic Retriever (VPR) modules for N stages. Each stage, containing several VPR modules, is responsible for features of a certain scale. Given two consecutive frames of an input video as an example, we denote features of two frames at a certain scale extracted from backbone as $\mathbf{X_t}, \mathbf{X_{t-1}} \in \mathbb{R}^{D\times C}$ where D, C, t refer to the spatial size (height $\times$ width), the number of channels of feature maps, and time index, respectively. The VPR takes the certain scale features of two frames ($\mathbf{X_t}$, $\mathbf{X_{t-1}}$), position embedding ($\mathbf{P} \in \mathbb{R}^{D\times C}$), and *panoptic slots* ($\mathbf{S} \in \mathbb{R}^{L\times C}$) as inputs to generate spatio-temporal coherent panoptic slots, where L, C denote the number of slots and channels, respectively. The final prediction heads further utilize the output panoptic slots to predict the classes, masks, and IDs of panoptic objects in the video. Note that the dimension index of the mini-batch is omitted for clarity, and the detailed network structure is shown in the Supplementary Material.

Panoptic slots. To unify the representation in the video, we define panoptic slots, a set of learnable parameters $\mathbf{S} \in \mathbb{R}^{L\times C}$, to represent all panoptic objects (both *things* and *stuff*) within a video. Each slot corresponds to an object, hence the slot number L represents the number of possible panoptic objects (*e.g.* 100) in the video. Panoptic slots are randomly initialized and can be gradually optimized through interacting with spatio-temporal information.

3.2. Video Panoptic Retriever (VPR)

The VPR consists of the Panoptic Retriever and the Video Retriever. For each module $i \in \{1, ..., U\}$, where U is the total number of the VPR modules in the network, the Panoptic Retriever associates the input panoptic slots $\mathbf{S}^i_t, \mathbf{S}^i_{t-1} \in \mathbb{R}^{L\times C}$ with the features of each frame to produce spatially coherent output panoptic slots $\hat{\mathbf{S}}^i_t, \hat{\mathbf{S}}^i_{t-1} \in \mathbb{R}^{L\times C}$ for each frame. In this process, Panoptic Retriever retrieves the object's information from features through an attention structure called *Retriever*. Then the Video Retriever, taking the panoptic slots $\hat{\mathbf{S}}^i_t$ and $\hat{\mathbf{S}}^i_{t-1}$ as input, further utilizes the Retriever to extract temporal correlations between these panoptic slots for temporally enhanced panoptic slots. The spatio-temporal refined panoptic slots are then forwarded into the next VPR for iterative refinements. Note that the $\mathbf{S}^0_t$, $\mathbf{S}^0_{t-1}$ are identical to $\mathbf{S}^0$ for the first stage.

Retriever (RE). We introduce the Retriever, which is for retrieving information correlated to the query. This module can be cast in two different aspects. For the spatial domain, it is considered as a learning process to map from spatial features to panoptic slots. For the temporal domain, it can be regarded as learning association between slots of different time frames. Different from classic dot-product attention [40], we incorporate a slots competing mechanism [29] to Retriever, which enables better discrimination of objects, considering that each object should be distinct from other objects both in spatial and temporal domains.

Here we introduce the formulation of Retriever. Denote the inputs of Retriever as Query $\mathbf{Q} \in \mathbb{R}^{L_q\times C}$, Key $\mathbf{K} \in \mathbb{R}^{D_v\times C}$ and Value $\mathbf{V} \in \mathbb{R}^{D_v\times C}$, as shown in Figure 3. The process of Retriever can be explained by three steps , including information transformation, correlation calculation, and correlated information retrieval.

Three Linear layers are first applied to transform panoptic slots and target information into the common space. We can denote the transformed slots, keys, and values as $\mathbf{Q}_\theta \in \mathbb{R}^{L_q\times C}$, $\mathbf{K}_\phi \in \mathbb{R}^{D_v\times C}$, $\mathbf{V}_g \in \mathbb{R}^{D_v\times C}$ respectively.

In the second step, the correlation between L_q slots and D_v vectors is associated with matrix multiplication operation, resulting in a correlation matrix $\mathbf{M} \in \mathbb{R}^{D_k\times L_q}$. If the specific object represented by the slot is correlated with a specific vector, then the corresponding value in the correlation matrix will be high. Furthermore, to alleviate the phenomenon that two slots correspond to the same target vector, we let the slots compete with each other through applying Softmax on the slot dimension. The above process can be formulated as:

$$\begin{aligned} \mathbf{M} &= \mathbf{K}_\phi \cdot \mathbf{Q}^T_\theta, \\ \mathbf{A}_{x,y} &= \frac{e^{M_{x,y}}}{\sum_l e^{M_{x,l}}}, \text{ for } x = 1, ..., D_v; y = 1, ..., L_q, \end{aligned} \tag{1}$$

where $\mathbf{M}_{x,y}$, $\mathbf{A}_{x,y}$ are the values at position (x, y) of correlation matrix $\mathbf{M}$ and resulting attention matrix $\mathbf{A}$ respectively. The operation of applying softmax on slot dimension normalizes the contributing weights of each slot, hence slots will be distinct from each other and the redundancy among slots will be suppressed.

In the final step, the resulting attention matrix will be applied to the value features $\mathbf{V}_g$ to retrieve related object's information. This can be formulated as:

$$\mathbf{O} = \mathbf{A}^T \cdot \mathbf{V}_g, \tag{2}$$

where $\mathbf{O} \in \mathbb{R}^{L_q\times C}$ represents the retrieved object's information.

Panoptic Retriever. Panoptic Retriever processes the features with position embedding and panoptic slots of each frame sequentially. Self-attention, Retriever, and Feed Forward Network (FFN) make up the Panoptic Retriever. Self-attention and FFN are for refining the panoptic slots before and after the association of the panoptic slots with spatial features through Retriever, respectively. Take one frame t as an example, given features $\mathbf{X_t}$ of certain scale, panoptic slots $\mathbf{S}^{i-1}_t$ and position embedding $\mathbf{P}$, then the procedure of the Panoptic Retriever can be written as:

$$\begin{aligned} \hat{\mathbf{S}}^i_t &= \mathbf{S}^{i-1}_t + \text{SA}(\mathbf{S}^{i-1}_t), \\ \hat{\mathbf{S}}^i_t &= \hat{\mathbf{S}}^i_t + \text{RE}(\hat{\mathbf{S}}^i_t, (\mathbf{X_t} + \mathbf{P}), \mathbf{X_t}), \\ \hat{\mathbf{S}}^i_t &= \hat{\mathbf{S}}^i_t + \text{FFN}(\hat{\mathbf{S}}^i_t), \end{aligned} \tag{3}$$

where SA, RE refers to Self-attention and Retriever. $\hat{\mathbf{S}}_t^i \in \mathbb{R}^{L\times C}$ denotes the output panoptic slots refined with spatial information through Panoptic Retriever. The same operations will be applied to the frame $t-1$, and the output panoptic slots of frame $t-1$ can be denoted as $\hat{\mathbf{S}}_{t-1}^i \in \mathbb{R}^{L\times C}$. Note that Layer Normalization after summation operation is omitted in the equations of this paper for simplicity.

In the above process, Retriever retrieves the object's information (*e.g.* location, appearance information) from spatial features through associating panoptic slots with every pixel in the spatial features. Panoptic slots ($\hat{\mathbf{S}}_t^i$ or $\hat{\mathbf{S}}_{t-1}^i$) in a single frame act as the $\mathbf{Q}$ in Retriever, and $\mathbf{K}, \mathbf{V}$ in Retriever are based on the spatial features of each frame. Position embedding is added to the spatial features to enhance spatial information. The slots competing mechanism of Retriever facilitate that panoptic slots are mutually exclusive so that only one panoptic object is allocated to one panoptic slot.

Video Retriever. The Video Retriever consists of Retriever and FFN. It aims at relating the panoptic slots across frames and helping these slots corresponding to the same object refine with each other's information. In this way, the consistency of the panoptic slots describing the same object across frames will be much improved.

Given the output panoptic slots ($\hat{\mathbf{S}}_t^i, \hat{\mathbf{S}}_{t-1}^i$) from Panoptic Retriever, these panoptic slots will be concatenated along the slot dimension and then forwarded into the Retriever. The following FFN is applied to refine the output panoptic slots of Retriever. And the final refined panoptic slots will be re-distributed to the panoptic slots of the corresponding frame. The process in Video Retriever can be summarized as follows:

$$\begin{aligned} \hat{\mathbf{S}}_a^i &= [\hat{\mathbf{S}}_t^i, \hat{\mathbf{S}}_{t-1}^i], \\ \mathbf{S}_a^i &= \hat{\mathbf{S}}_a^i + \text{RE}(\hat{\mathbf{S}}_a^i, \hat{\mathbf{S}}_a^i, \hat{\mathbf{S}}_a^i), \\ \mathbf{S}_a^i &= \mathbf{S}_a^i + \text{FFN}(\mathbf{S}_a^i), \\ [\mathbf{S}_t^i, \mathbf{S}_{t-1}^i] &= \hat{\mathbf{S}}_a^i + \mathbf{S}_a^i, \end{aligned} \tag{4}$$

where $\hat{\mathbf{S}}_a^i \in \mathbb{R}^{2L\times C}$ denotes the concatenated panoptic slots, RE refers to Retriever, $[\cdot, \cdot]$ indicates the concatenation operation, $\mathbf{S}_a^i$ represents the refined concatenated panoptic slots, and $\mathbf{S}_t^i, \mathbf{S}_{t-1}^i \in \mathbb{R}^{L\times C}$ denote the spatio-temporally refined output panoptic slots of frame t and $t-1$, respectively.

Different from the Retriever in Panoptic Retriever, the concatenated panoptic slots serve as $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ in Retriever, which utilizes the slots competing mechanism to help make sure the unique connection between two specific panoptic slots across frames. This also paves the way for ID assignment in ID prediction head, which already can achieve good performance without relying on any other information but only the spatio-temporally refined panoptic slots. Besides, the whole process of Video Retriever is friendly for variational frame numbers due to the parameters are only related to the length of slot vector.

Prediction heads. After the above Panoptic Retriever and Video Retriever operations, the panoptic slots will contain the object's information in each frame and slots corresponding to the same object across frames will be as consistent as possible. To produce classification, mask, and object ID predictions from panoptic slots, we utilize three prediction heads, each consisting of an FFN (two Linear layers) and the respective functional layer. In the classification head, the FFN is followed by another Linear layer to output the class prediction for each panoptic object. In the mask head, the mask of the panoptic object is obtained by applying panoptic slots on the feature map through matrix multiplication operation. In the ID head, the ID of the panoptic object is predicted by calculating the similarity matrix between the panoptic slots in the current frame and previous frames. Benefitting from the coherent panoptic slots, all prediction heads can provide precise predictions based on the slot information and no complex fusion operations are required.

4. Experiments

4.1. Implementation Details

Cityscapes-VPS. Cityscapes-VPS [17] is built on top of the validation set of the Cityscapes dataset [9]. It provides dense panoptic annotations for 3000 frames, sampling six frames from every 500 video clips where a single video clip contains 30 frames of 1024×2048 resolution with 19 classes (11 stuff and 8 things), and instance ID association across frames is also provided. The split of train, validation, and test sets are $400, 50, 50$ clips respectively. We report results on its validation set and test set.

VIPER. VIPER dataset for the VPS task [17] is reformatted based on the synthetic VIPER dataset [36] extracted from the GTA-V game engine. The re-formatted VIPER dataset contains panoptic annotations for 23 classes (13 stuff and 10 things) on 184K frames of ego-centric driving scenes at 1080×1920 resolution. We follow the public train and val splits as [17]. For training, 19 videos with total 41464 frames are utilized. For evaluation, there are total 600 images, consisting of the first 60 frames of 10 validation videos from the day scenario.

Evaluation metric. The Video Panoptic Quality (VPQ) [17] is adopted for evaluation. As the video extension of Image Panoptic Quality (PQ) [19], VPQ is designed for evaluating the spatio-temporal consistency between the predicted and ground truth panoptic video segmentation. For a video sequence, denote the temporal window size as $k \in \{0, 5, 10, 15\}$, several snippets can be obtained by sliding the window through the video, and the calculation of snippet-level IoU, |TP|, |FP| and |FN| is performed, accord-

ingly. At a dataset level, all these values are collected across all predicted videos and the dataset-level VPQ^k result is computed for each class and averaged across all classes. The final VPQ is computed by averaging over different k values.

Training. The implementation is based on the MMDetection [4] toolbox. For most experiments, ResNet50 [15] and FPN [25] serve as a backbone network. Except for ResNet50, we also validate our method on other backbones, such as Swin-B, Swin-L Transformer [28], *etc*. Training losses include four image-level losses [43] (*i.e*. PQ loss, pixel-wise instance discrimination loss, per-pixel mask-ID cross-entropy loss, and semantic segmentation loss) and an identification loss as a video-level loss. Corresponding loss weights are empirically set to $3.0, 1.0, 0.3, 0.5, 0.5$ respectively. As for training data of the Cityscapes-VPS dataset, we find that the temporal annotation is not very consistent, hence we mainly utilize the image annotations and generate the simulated videos from these images through random scaling and translating [51].

Distributed training with 8 GPUs is utilized and batch size is set as 1 for each GPU. For all datasets, the optimizer, base learning rate, weight decay, and learning rate scheduler are set to AdamW [31], $0.0001, 0.0001$ and StepLR respectively. For Cityscapes-VPS, we train for 96 epochs and apply lr decay at 64 and 88 epochs. Image panoptic segmentation pretraining for backbone and Panoptic Retriever parts are adopted with Mapillary Vistas [32] and Cityscapes train sequences with pseudo labels [5]. For VIPER, we train for 12 epochs and apply lr decay at 8 and 11 epochs. Image panoptic segmentation pretraining is adopted on VIPER dataset. The remaining layers, *e.g*. Video Retriever and ID prediction head are initialized by Kaiming initialization.

Inference. For Cityscapes-VPS, all 30 frames of a video are predicted but only 6 frames with ground truth labels are evaluated. For VIPER, all frames are predicted and evaluated. For all datasets, simple mask filtering and removing are adopted. In this process, object masks are filtered with a class confidence score of 0.85 and a pixel confidence score of 0.4 is applied on things masks. The things masks with an overlapping ratio greater than 0.03 and stuff masks with area less than 4096 are removed.

4.2. Ablation Study

We first analyze the panoptic slots, then study several aspects to discuss the effect of different settings on overall performance. Unless otherwise specified, experiments are conducted with ResNet50 and FPN on Cityscapes-VPS *val* in this section.

What have panoptic slots learnt? To better understand what panoptic slots are and what panoptic slots have learnt, we show the visualization of activation maps on the attention maps at different stages in Figure 4. The maps

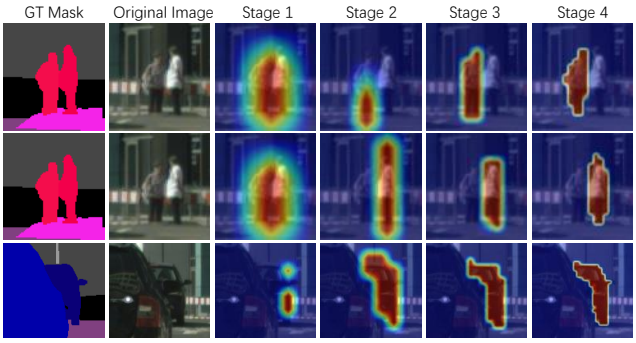

Figure 4. **Attention maps at different stages.** Based on the visualization of the CAM (Class Activation Map), We can observe that our panoptic slots only roughly localize objects in early stages and gradually match the object boundaries better and better.

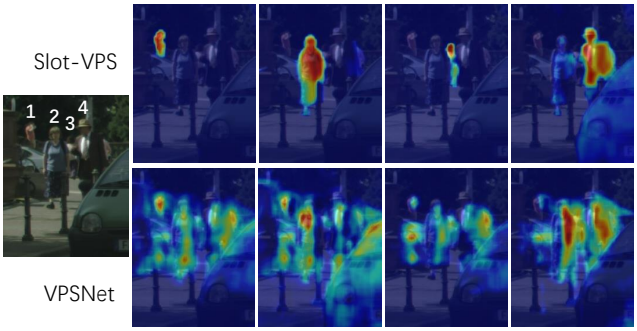

Figure 5. **Qualitative comparison of feature maps between Slot-VPS (*top*) and VPSNet [17](*bottom*).** We can find that each of our CAM focuses on a specific object, which verifies that the learned features are object-centric; while the CAMs from [17] show that their features can not distinguish instances, thus requiring surrogates (*e.g*. bbox, center) to further localize objects. Best viewed when zoomed in.

show the most contributing region to a specific panoptic object in the image. They are computed with SEG-GRAD-CAM [41], which is the extension of Grad-CAM [37] to semantic segmentation. Thanks to the unified end-to-end framework of Slot-VPS, we can compute the loss directly for a predicted instance mask, and obtain its intra-class-distinguishable CAM (Class Activation Map) for any feature or attention map. As shown in Figure 4, at early stages, multiple objects may be activated together and the focusing area is large. With the iterative learning process, other irrelevant information in the scene is suppressed and the target object becomes more clearly distinguishable.

Retriever. There are two main differences between our Retriever and the classic attention [40]. The first one is the dimension on which the softmax operation is performed. We apply softmax to the slot dimension (*i.e*. Query's dimension) instead of the dimension of spatial size (height $\times$ width) of feature maps (*i.e*. Key's dimension). Applying

Softmax dim	*w/* Projection	PQ / PQth / PQst
slot		60.5 / 48.4 / 69.3
spatial		53.5 / 37.5 / 65.1
slot	✓	60.7 / 47.9 / 70.1

Table 1. **Variations in Retriever.** Rows rendered in yellow are the final settings. PQ, PQth, PQst are the averaged scores for all / Things / Stuff classes respectively.

	Encoder	*w/* multi-scale	PQ / PQth / PQst
DETR [3]		✓	38.6 / 27.0 / 47.1
	✓	✓	56.3 / 44.5 / 64.8
Ours			58.9 / 44.6 / 69.4
		✓	**60.5 / 48.4 / 69.3**

Table 2. **Comparison between Panoptic Retriever and DETR [3].** "*w/* multi-scale" stands for whether utilizing the multi-scale features.

Slot number	PQ / PQth / PQst
50	57.4 / 42.7 / 68.1
80	57.6 / 46.4 / 65.7
100	**60.5 / 48.4 / 69.3**
200	58.7 / 47.6 / 66.7
300	58.2 / 45.7 / 67.4

Table 3. **Comparison between different slot number.**

FFN Hidden dim	scale / 32	scale / 16	scale / 8	scale / 4	PQ / PQth / PQst
1024	1	2	2	2	56.4 / 40.9 / 67.6
	2	2	2	2	58.9 / 44.7 / 69.2
	2	3	3	3	60.1 / 48.3 / 68.7
2048	1	1	1	1	51.6 / 30.1 / 67.2
	1	1	1	2	56.1 / 45.4 / 63.8
	1	2	2	2	**60.5 / 48.4 / 69.3**

Table 4. **Comparison between module number variance on multi-scale features.**

softmax to the spatial size dimension enhances the pixel-level discriminability so that the location and appearance information of objects can be obtained from features but object-level relations are not explored, while applying softmax on the slot dimension can facilitate the object-level competition so that the discriminability of objects can be enhanced. The second one is whether there is a projection layer after applying the attention matrix on the Value features. We conduct ablation studies on these two factors of Retriever on Image Panoptic Segmentation task. As shown in Table 1, we can observe that changing the softmax's applying dimension from slot to spatial size degrades the performance a lot, which validates that the slots competing mechanism at object-level is effective. Adding the final projection layer only brings minor performance gain, so we do not keep this layer for overall efficiency.

Panoptic Retriever *vs.* Transformers in DETR. Our Panoptic Retriever is inspired by the decoder structure in the transformer (*e.g.* DETR [3]). The transformers in DETR [3] consists of encoder, decoder, and panoptic mask head. Concretely, there are three key differences between the Panoptic Retriever and the transformer structure in DETR:

(1) The attention mechanism in DETR applies the softmax along the spatial dimension, which only discriminates different pixels instead of competing among objects. Our Panoptic Retriever applies softmax on the slot dimension to encourage the competition between slots so that the information of each slot becomes mutually exclusive. Table 1 validates the efficacy of such a competition mechanism.

(2) DETR highly relies on the transformer encoders on top of the backbone. The transformer encoder brings great computational cost and greatly slows down the entire framework. Without the help of the encoder, our Panoptic Retriever already achieves good performance and can be applied efficiently.

(3) DETR only uses the low-resolution feature maps in the encoder and fuses multi-scale features in the extra panoptic mask head, which could be harmful to the modeling of small objects. Instead, we feed and fuse multi-scale features into Panoptic Retriever without requiring additional heads.

As shown in the first two rows of Table 2, removing the encoder of the transformer will lead to great degradation, while our network can perform much better without the help of the encoder (4th row). Even without the use of multi-scale features (the 3rd row), we can still outperform the DETR with six transformer encoders by 2.6 PQ. Note that utilizing multi-scale features mainly improves things' performance.

Slot number. In this part, we explore the effect of different slots numbers as shown in Table 3. As it can be observed, since a slot is a structure designed to take full responsibility for a single object, it is desirable to set the slot number close to the number of available panoptic objects in a scene (*e.g.* 100 in this setting). If the slot number is too small, the multiple objects are more likely to be assigned to the same panoptic slot, resulting in confusion between panoptic objects. Some objects will be missed due to insufficient slots. In contrast, if the slot number is too large, a single object might spread into multiple slots as fragments during the competition, which generates less reliable slots compared to the ideal case of each slot responsible for a single object.

Module number in each stage. To sufficiently refine the slots with spatio-temporal coherent information, we apply the VPR module multiple times with multi-scale features. We survey the effect of different module numbers on each scale under two settings, setting the hidden dim in FFN of model to 1024 and 2048 respectively. As shown in Table 4, the performance is improved with the increase of per-scale module number for both settings. However, this will bring extra computation complexity. By default, we empirically set the module number of four scales features to $1, 2, 2, 2$ respectively. As in Figure 2, such a setting can achieve a better balance between the performance and latency (59.7

	Methods	Temporal window size k				VPQ/VPQth/VPQst	FPS
		k=0	k=5	k=10	k=15		
Cityscapes-VPS val	VPSNet [17]	64.5 / 58.1 / 69.1	57.4 / 45.2 / 66.4	54.1 / 39.5 / 64.7	52.2 / 36.0 / 64.0	57.0 / 44.7 / 66.0	1.3
	SiamTrack [47]	64.6 / 58.3 / 69.1	57.6 / 45.6 / 66.6	54.2 / 39.2 / 65.2	52.7 / 36.7 / 64.6	57.3 / 44.7 / 66.4	4.6
	Ours	65.7 / 57.9 / 71.4	60.0 / 47.7 / 68.9	57.8 / 44.4 / 67.6	55.5 / 40.2 / 66.7	**59.7 / 47.5 / 68.6**	**4.6**
Cityscapes-VPS test	VPSNet [17]	64.2 / 59.0 / 67.7	57.9 / 46.5 / 65.1	54.8 / 41.1 / 63.4	52.6 / 36.5 / 62.9	57.4 / 45.8 / 64.8	1.3
	SiamTrack [47]	63.8 / 59.4 / 66.6	58.2 / 47.2 / 65.9	56.0 / 43.2 / 64.4	54.7 / 40.2 / 63.2	57.8 / 47.5 / 65.0	4.6
	Ours	66.4 / 58.8 / 71.2	60.9 / 48.8 / 68.5	57.5 / 44.2 / 65.9	55.8 / 40.8 / 65.4	**60.1 / 48.2 / 67.8**	**4.6**

Table 5. **Comparison to the state-of-the-art with ResNet50-FPN on Cityscapes-VPS *val* and *test*.** VPQ, VPQth, VPQst are the averaged scores for all / Things / Stuff classes respectively.

	DenseContext	AutoAug	RFP	SSL	TTA	Backbone	Image model M-adds (B)	val VPQ	test VPQ
ViP-Deeplab [35]	✓					WR-41	3246	60.9	-
	✓	✓	✓					61.9	-
	✓	✓	✓	✓	✓			63.1	62.5
Ours						Swin-L-FPN	1917	62.2	61.6
					✓			**63.7**	**63.3**

Table 6. **Comparison to the state-of-the-art [35] on Cityscapes-VPS *val* and *test*.** We list the tricks applied in both methods including DenseContext [35], AutoAugment [10], Recursive Feature Pyramid (RFP) [34], Semi-Supervised Learning (SSL) with pseudo labels [5] and Test Time Augmentation (TTA) [35].

Methods	Backbone	VPQ	FPS
SiamTrack [47]	ResNet50-FPN	50.2	5.1
VPSNet [17]	ResNet50-FPN	51.9 (+1.7)	1.0
Ours	ResNet50-FPN	53.2 (+3.0)	4.2
	Swin-L-FPN	**56.2** (+6.0)	2.0

Table 7. **Comparison to the state-of-the-art on VIPER *val*.**

VPQ with 4.6 FPS on Cityscapes-VPS *val* set).

Video Retriever. Previous works exploit the temporal information through estimating optical flow [16] or applying attention mechanisms [44]. However, these techniques are mostly applied to the extracted features, which contain all objects' information in the scene. This also leads to the requirement for extra surrogates (*e.g.* bbox, center) to localize objects from features. In comparison, our Video Retriever is directly applied to the object-centric representations, which will eliminate the effect of irrelevant background noises and benefit the object-level mutual refinement. As shown in Figure 5, take VPSNet [17] as an example, thanks to object-centric learning, our feature maps are more object-centric while the feature maps of [17] are lack of specific objects' information. Experiment shows that there is 1.5 VPQ drop when Video Retriever is removed.

4.3. Comparison with the State-of-the-art.

The VPS results on Cityscapes-VPS are shown in Table 5. With the same backbone (ResNet50-FPN), our model outperforms the SiamTrack [47] by 2.4, 2.3 VPQ on Cityscapes-VPS *val* and *test* respectively with the fastest inference speed. As exhibited in Table 6, our method can also outperform the state-of-the-art [35] by 0.6, 0.8 VPQ respectively with much less computation and fewer tricks applied, which indicates that our framework could be further boosted with similar tweaks in the future. The VPS results on the VIPER dataset are shown in Table 7. With the same backbone (ResNet50-FPN), our model outperforms the state-of-the-art [17] by 1.3 VPQ on the val set with 4× FPS as [17]. With the larger backbone (Swin-L and FPN), our performance can be further improved by 3.0 VPQ while achieving 2× faster as [17], thanks to the unified framework and the elimination of complex CPU-based post-processing. Qualitative visualizations will be shown in the Supplementary Material.

5. Conclusion

In this paper, to alleviate the disadvantages of the decomposed pipeline for the VPS task, we introduce a fully unified end-to-end framework, Slot-VPS, based on object-centric representation learning. Panoptic objects (including things and stuff) in the video are represented with a unified representation called panoptic slots. The proposed Video Panoptic Retriever (VPR) retrieves and encodes the spatio-temporal information of objects in the video into the panoptic slots. Finally, the spatio-temporal coherent panoptic slots can be utilized for directly predicting the class, mask, and object ID of panoptic objects in the video. Experimental results on two datasets of the VPS task validate the effectiveness of our method.

References

[1] Hossein Adeli, Seoyoung Ahn, and Gregory Zelinsky. Recurrent attention models with object-centric capsule representation for multi-object recognition. *arXiv preprint arXiv:2110.04954*, 2021. 3

[2] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166, 2019. 1

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2, 3, 7

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[5] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *European Conference on Computer Vision*, pages 695–714. Springer, 2020. 6, 8

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[7] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12485, 2020. 2, 3

[8] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *arXiv preprint arXiv:2107.06278*, 2021. 2, 3

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5

[10] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 8

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3

[12] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019. 3

[13] Ruohan Gao, Dinesh Jayaraman, and Kristen Grauman. Object-centric representation learning from unlabeled videos. In *Asian Conference on Computer Vision*, pages 248–263. Springer, 2016. 3

[14] 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. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 6

[16] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 8

[17] 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. 1, 2, 3, 5, 6, 8

[18] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 2

[19] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9404–9413, 2019. 2, 5

[20] Justin Lazarow, Kwonjoon Lee, Kunyu Shi, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10720–10729, 2020. 2

[21] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020. 1

[22] Qizhu Li, Anurag Arnab, and Philip HS Torr. Weakly- and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 102–118, 2018. 2

[23] Qizhu Li, Xiaojuan Qi, and Philip HS Torr. Unifying training and inference for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13328, 2020. 2

[24] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. *arXiv preprint arXiv:2012.00720*, 2020. 2

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3, 6

[26] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9816–9825, 2021. 3

[27] Iou-Jen Liu, Zhongzheng Ren, Raymond A Yeh, and Alexander G Schwing. Semantic tracklets: An object-centric representation for visual multi-agent reinforcement learning. *arXiv preprint arXiv:2108.03319*, 2021. 3

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 6

[29] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055*, 2020. 3, 4

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[32] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4990–4999, 2017. 6

[33] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Bridge to answer: Structure-aware graph interaction network for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15526–15535, 2021. 1

[34] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10213–10224, 2021. 8

[35] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. *arXiv preprint arXiv:2012.05258*, 2020. 1, 2, 3, 8

[36] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2213–2222, 2017. 5

[37] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 6

[38] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 1

[39] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 282–298. Springer, 2020. 2

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3, 4, 6

[41] Kira Vinogradova, Alexandr Dibrov, and Gene Myers. Towards interpretable semantic segmentation via gradient-weighted class activation mapping. *arXiv preprint arXiv:2002.11434*, 2020. 6

[42] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6578–6588, 2020. 1

[43] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021. 2, 3, 6

[44] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 8

[45] 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. 3

[46] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 107–122. Springer, 2020. 1

[47] 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. 1, 3, 8

[48] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 2, 3

[49] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5188–5197, 2019. 3

[50] Qi Zheng, Chaoyue Wang, and Dacheng Tao. Syntax-aware action targeting for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13096–13105, 2020. 1

[51] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020. 6

[52] 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. 3