Learning to Associate Every Segment for Video Panoptic Segmentation

Sanghyun Woo[1] Dahun Kim[1] Joon-Young Lee[2] In So Kweon[1]

[1]KAIST [2]Adobe Research

Abstract

Temporal correspondence - linking pixels or objects across frames - is a fundamental supervisory signal for the video models. For the panoptic understanding of dynamic scenes, we further extend this concept to every segment. Specifically, we aim to learn coarse ***segment-level*** *matching and fine* ***pixel-level*** *matching together. We implement this idea by designing two novel learning objectives. To validate our proposals, we adopt a deep siamese model and train the model to learn the temporal correspondence on two different levels (i.e., segment and pixel) along with the target task. At inference time, the model processes each frame independently without any extra computation and post-processing. We show that our per-frame inference model can achieve new state-of-the-art results on Cityscapes-VPS and VIPER datasets. Moreover, due to its high efficiency, the model runs in a fraction of time (3×) compared to the previous state-of-the-art approach.*

1. Introduction

A holistic understanding of a video requires pixel-level information of different semantics, object instances, background stuffs as well as their temporal changes. Despite being very challenging, having such video understanding is crucial for various vision applications such as autonomous driving, robot control, video editing, and augmented reality. Video semantic segmentation has been regarded as one of the representative proxies for this ambitious goal. The community has proposed a large number of learning-based approaches under two main research directions of improving accuracy [9,11,12,21,31,50] and efficiency [20,27,30,38,51]. While there has been a flurry of advances in model designs, significantly less effort has been made to the training objectives. The main bottleneck of developing a useful supervision signal is the scarcity of video annotations available. Most video segmentation benchmarks [7,49] provide annotations for only a single frame per video clip, which limits these annotations still to an *image* level.

In line with the growing importance of the field and to meet the high demand for video labels, Kim *et al*. [24] recently presented a new challenging dense video understanding problem called video panoptic segmentation (VPS) and presented a pair of video annotations: Cityscapes-VPS and VIPER datasets. The problem unifies the existing video semantic segmentation [11,38,50] and video instance segmentation [2,48]. It aims at a simultaneous prediction of object classes, masks, instance id associations, and semantic segmentation for *all* pixels in a video. The two VPS benchmarks provide real-world and simulation-based video datasets, opening up the possibilities to probe new video-specific training signals.

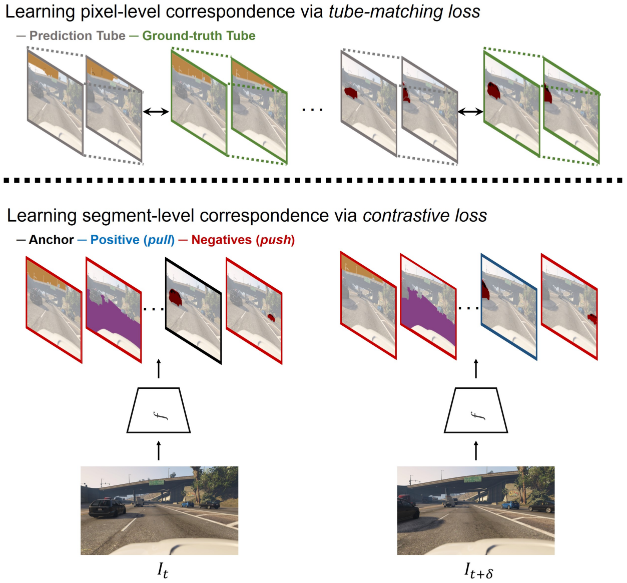

Learning pixel-level correspondence via *tube-matching loss*
— Prediction Tube — Ground-truth Tube
Learning segment-level correspondence via *contrastive loss*
— Anchor — Positive (*pull*) — Negatives (*push*)
f
f
I_t
$I_{t+\delta}$

Figure 1: **Overview of our approach.** Temporal correspondence learning is fundamental for video understanding. In this work, we extend this concept to every segment in a video. Specifically, given a pair of frames, I_t and $I_{t+\delta}$, a siamese model, f, learns to associate every segment at two different levels jointly: *segment-level* via contrastive loss and *pixel-level* via tube-matching loss.

Identifying ***temporal correspondence*** - "what went where" - in a video is a crucial requirement of robust visual reasoning in space and time. Numerous methods for learning temporal correspondences, from pixel-level to object-level, have been developed so far, e.g., optical-flow estimation [18, 29, 39] and object tracking [3, 40, 41, 43, 44]. However, most of the previous methods aim at addressing a single-level of temporal correspondence at a time, and less attempt has been made to solve different levels of temporal correspondences jointly. For the dense panoptic segmentation of a video, we argue that the model representations should support reasoning at various levels of temporal correspondences. At the same time, this should be considered for every individual segment [1] in a video. To this end, we train a video segmenter that simultaneously learns correspondences across frames at both segment-level and pixel-level (see Fig. 1). In the following, we describe these two important views in turn:

1) Segment-level correspondence learning: Let's imagine we assign an id tag for each segment in a video. Naturally, we provide the same id tag for the same segment over time. We formulate this concept as a graph matching problem. In practice, we construct the graph from video frames, where nodes encode segments in each frame and edges are affinities between them. We then aim to learn features such that strong edges represent temporal correspondences. This is achieved through the *contrastive learning* [14], which can encourage pairs of segment embedding to have strong edges if they are temporally associated or weak edges otherwise. Also, as time goes, the segments in a video undergo dynamic appearance changes such as deformation, occlusion, and perspective distortion. Therefore, without composing multiple handcrafted data augmentations [1, 4–6, 13, 15, 32], our formulation of matching the same segment at temporally distant frames naturally leads to learning segment representations invariant in such visual distortions.

2) Pixel-level correspondence learning: The optical flow provides dense pixel-level correspondences where each pixel in the current frame goes in the next frame. The photometric loss is often used for learning this. Recent works [9, 30] generalize the photometric loss to the logit domain, i.e., segmentation output, with the motivation of predicting temporally consistent labels. However, as warping loss often assumes independence between each pixel, higher-level correlations or structures over pixels are hardly modeled. To alleviate this issue, we introduce "tubes" formed by linking the segment masks along the time axis. The model then learns to minimize the mismatch between the prediction tubes and the ground-truth tubes, globally optimizing the entire chain of intermediate mask predictions. This constraint allows the model to capture fine-grained changes in segments, e.g., shape, boundary, and motion tendencies over time. Empirically, we show that the proposed mask tube matching loss performs better than the warping loss in learning an accurate segmentation model.

We aim to learn these two different levels of temporal correspondences jointly. To do so, we propose an efficient framework. Specifically, we use a deep siamese model and train with a pair of frames, where a neighbor reference frame in a pair is randomly selected with a time gap relative to the current target frame. Our model can encode strong temporal consistency into the features during training without using any heavy feature aggregation [50] or fusion [45] operations, resulting in an efficient yet strong video model. Without bells and whistles, our model achieves new state-of-the-art results on Cityscapes-VPS and VIPER datasets while running in a fraction of time compared to the previous state-of-the-art approach [24]. We summarize the contributions of this paper as follows.

1. We generalize the temporal correspondence learning to every segment in a video. We present to learn coarse segment-level matching and fine pixel-level matching together. We achieve this by designing two novel objective functions with an efficient learning framework.
2. We propose a new supervised[2] contrastive learning method to learn the temporal correspondences in a video. In particular, we aim to maximize the mutual information between representations of temporally distant frames of the same segment.
3. We achieve new state-of-the-art on benchmarks, clearly demonstrating the effectiveness of our approach. We additionally provide extensive experimental analysis with ablation studies.

2. Related Works

Video Segmentation Video segmentation aims to assign pixel-wise semantic labels to video frames. As an important task for dynamic scene understanding, it has attracted attention increasingly from the research community. Numerous approaches have been proposed in the literature, either focusing on improving the segmentation quality [9, 11, 12, 21, 31, 50] or accelerating the computation speed [20, 27, 30, 38, 51]. With the growing importance of the field, a more fine-grained dynamic scene understanding task, video panoptic segmentation [24], has been recently presented. It unifies semantic segmentation, instance segmentation, and multi-object tracking into a single coherent task. Kim *et al*. [24] designed a strong baseline model called VPSNet that consists of a shared backbone with multiple task-specific heads. While effective, the model enforces only weak temporal consistency by aggregating neighboring features using optical-flow [18] and tracking foreground objects [48]. Thus, they missed leveraging intrinsic temporal information in a video fully. Our model instead benefits from learning strong temporal consistency, from segment-level to pixel-level, through the proposed learning objectives

[1] Throughout the paper, we use the term ***segment*** to denote the region of both foreground *things* and background *stuffs* in the video.

[2] We effectively leverage the video label information.

during training. With our learning strategy, we show that our per-frame inference model can even generate more accurate video segmentation results than the previous state-of-the-art method.

Temporal Correspondence Learning Temporal continuity provides a strong visual constancy that links disparate percepts into a smoothly-varying entity in the visual world. Learning temporal correspondence serves as a useful training signal in a video and it has been widely explored from pixel-wise to object-level such as optical flow estimation [18, 29, 39] and visual tracking [3, 40, 41, 43, 44]. However, there was less effort in learning different levels of temporal correspondences together, which are inherently related. In this paper, we explore two different temporal stability constraints in a video and show the effectiveness of joint learning. As large-scale video annotations are recently emerging through simulation [10, 17, 36, 37] or automated label generation methods [35], our work aims to develop commensurate video-specific learning strategies. Another orthogonal direction is the self-supervised approach that constructs pretext tasks by exploiting temporal information in a video, e.g., cycle-consistency of time [19, 26, 42, 46]. Since the training data in the video domain is more likely to be a semi-supervised setting, we believe the progress in both directions, supervised and unsupervised learning in the video, should be made in parallel.

Contrastive Learning Recent resurgence in contrastive learning has led to significant advances in a self-supervised learning paradigm [1, 4–6, 13, 15, 32]. A fundamental idea of contrastive learning [14] is that similar examples should be grouped together and far from other clusters of related examples. It is achieved by using three key ingredients, an anchor, positive, and negative(s): pulling together an anchor and a positive sample in embedding space and pushing apart the anchor from many negative samples. Data augmentations of the sample often construct a positive pair, and negative pairs are formed by the anchor and randomly chosen samples from the minibatch. Here, we aim to extend this current self-supervised contrastive learning paradigm from image to video. At the same time, from unsupervised to supervised by effectively leveraging the video label information. Our work shares a similar objective with the recent study, SupCon [22], but we explore a different set of view definitions, architectures, and application settings. Also, we contribute a unique empirical investigation of supervised contrastive learning on the structured output space, i.e., segmentation.

3. Method

Video frames are temporally coherent in nature. A video panoptic segmentation model should faithfully exploit this temporal stability to seamlessly capture the whole segment's panoptic movement in a video. Otherwise, inconsistency in any class label and instance-id will result in low video quality of these panoptic segmentation sequences. This paper suggests learning the temporal correspondences in a video from two different perspectives jointly: *segment* and *pixel*. In the following, we elaborate our approach in detail, including the network design (Sec. 3.1) and objective functions (Sec. 3.2). An overview of our method is shown in Fig. 2.

3.1. Network Architecture

In this paper, we design a deep siamese model, training on a pair of frames, where the neighbor reference frame is randomly sampled from a large range relative to the current target frame [50]. We then force the model to learn representations that can optimally associate the contents of the input frames.

Our model is fully convolutional and mainly consists of three parts: We first forward the target and reference images, I_t and $I_{t+\delta}$, into a siamese network where each sub-network has the same structure with shared parameters. We then match the intermediate features (or predicted panoptic logits) of the two temporally distant images. We finally compute the penalties to learn the temporal correspondences. All components are differentiable, allowing us to train the whole network end-to-end.

Segmentation model. While not being sensitive to any specific design of the segmentation model, we choose the previous state-of-the-art, VPSNet [24] as a baseline. To avoid memory issues during the training, we use an efficient version, VPSNet-Track [3], without their memory-heavy Fuse module. The model consists of a shared ResNet50-Feature Pyramid Network (FPN) backbone [28] with multiple task-specific heads: Mask-RCNN [16], deformable convolutions [8], and MaskTrack [48] for instance segmentation, semantic segmentation, and tracking, respectively.

We propose to learn a projection head to map the feature pyramid representations to a latent embedding space, which will be used for our contrastive feature learning. The projection head is only deployed at training time and abandoned at inference. We provide the details of the projection head in the following.

Projection head. Recent works [5, 6] introduce a small neural network so-called a projection head, e.g., MLP with one hidden layer, to map representations from the base encoder to a latent space where contrastive loss is applied. In this context, we treat our backbone, i.e., ResNet50-FPN, as a base encoder and perform the projection on the FPN features. Our projection head mainly consists of two parts. The first is a gathering operation where we merge multiple levels into a single, highest resolution level by resizing and elementwise adding the feature maps [33]. Second, we then apply two

[3] The heavy version, VPSNet-FuseTrack, cannot be trained with a pair of frames due to the GPU memory limit.

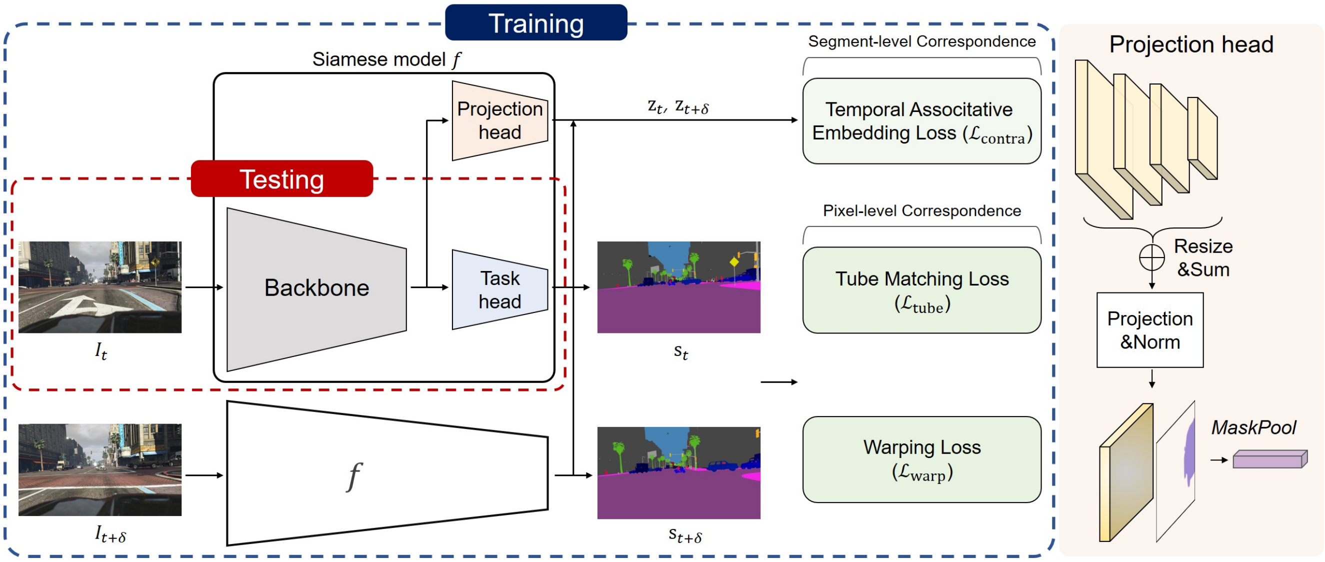

Training
Siamese model f
Projection head
Testing
Backbone
Task head
I_t
s_t
z_t, $z_{t+\delta}$
Segment-level Correspondence
Temporal Associative Embedding Loss ($\mathcal{L}_{\text{contra}}$)
Pixel-level Correspondence
Tube Matching Loss ($\mathcal{L}_{\text{tube}}$)
Warping Loss ($\mathcal{L}_{\text{warp}}$)
f
$I_{t+\delta}$
$s_{t+\delta}$
Projection head
Resize &Sum
Projection &Norm
MaskPool

Figure 2: (**Left**) The overall pipeline of our joint learning framework. We utilize a deep siamese model and train coarse *segment-level* matching and fine *pixel-level* matching together. At inference time, we process video frames in a per-frame manner, without relying on heavy temporal feature aggregation or fusion. (**Right**) The detailed illustration of the projection head, which is required for the contrastive learning at train time only. The forward procedure of the projection head is: First, the FPN features are resized to the highest resolution level, and element-wise summed. Second, it is projected to a latent space via projection and normalization. Finally, the MaskPool operation is applied to obtain a segment-level feature embedding vector. *Best viewed in color.*

3×3 convolutions and a ReLU in between them, followed by the L2 normalization layer. Finally, the resulting output feature is used for contrastive learning. While various projection head implementations are possible, our simple design already achieves great performance.

3.2. Objective Functions

We exploit the temporal continuity in a video as a crucial supervisory signal, which endows strong temporal stability priors to the model. To this end, we define the losses that guide the network to learn temporal correspondences from two different perspectives as

$$\mathcal{L}_{\text{tc}} = \lambda_{\text{segment}}\mathcal{L}_{\text{segment}} + \lambda_{\text{pixel}}\mathcal{L}_{\text{pixel}}, \tag{1}$$

which consists of loss functions for segment-level correspondence learning $\mathcal{L}_{\text{segment}}$ and pixel-level correspondence learning $\mathcal{L}_{\text{pixel}}$. In the following, we describe each term in detail.

3.2.1 Segment-level Correspondence

We formulate the learning of segment-level temporal correspondence as a graph matching problem. The graph is constructed from a video, where nodes indicate segments in each frame and edges are affinities between them. We aim to place large weights on a node pair that is temporally associated, i.e., finding a match. The remaining issues are 1) how we represent the nodes in each frame and 2) how we encode the temporal correspondences by strong edges.

To tackle the first problem, we use a mask pooling operation, which is inspired by Lorenzo *et al*. [35]. They use it to enhance the RoI features for robust Re-ID. By pooling the features under the instance mask, the background information is discarded, and only the necessary foreground information is left. Unlike theirs, our main purpose is to have a representative embedding vector for every segment in a frame. Therefore, we exploit the ground-truth *panoptic* masks. Specifically, we use the *things* and *stuff* masks to spatially pool the projected feature map for all individual segments. We treat the obtained segment-level vectors as the nodes of a graph for each frame.

To address the second problem, our solution is to apply contrastive learning [22]: *"pull" the embeddings if they are temporally associated and "push" otherwise*. Given a pair of temporally distant frames, I_t and $I_{t+\delta}$, we only consider N traceable segments. Then the embeddings of the same segment in two different frames, $z_{t,i}$ and $z_{t+\delta,i}$[4], are regarded as a positive pair and the embeddings of other 2(N-1) segments, e.g., $z_{t,j}$[5], as negatives. For a segment i, the loss function for a positive pair of examples $(t, t+\delta)$ is formulated as

$$l^{i}_{t,t+\delta} = -\log\frac{\exp(\text{sim}(z_{t,i}, z_{t+\delta,i})/\tau)}{\sum_{*}\sum_{k}\mathbb{1}_{[k\neq i\wedge *\neq t]}\exp(\text{sim}(z_{t,i}, z_{*,k})/\tau)} \quad \text{s.t.} \quad * \in \{t, t+\delta\}, \tag{2}$$

[4] $i \in [1, N]$
[5] $j \neq i$

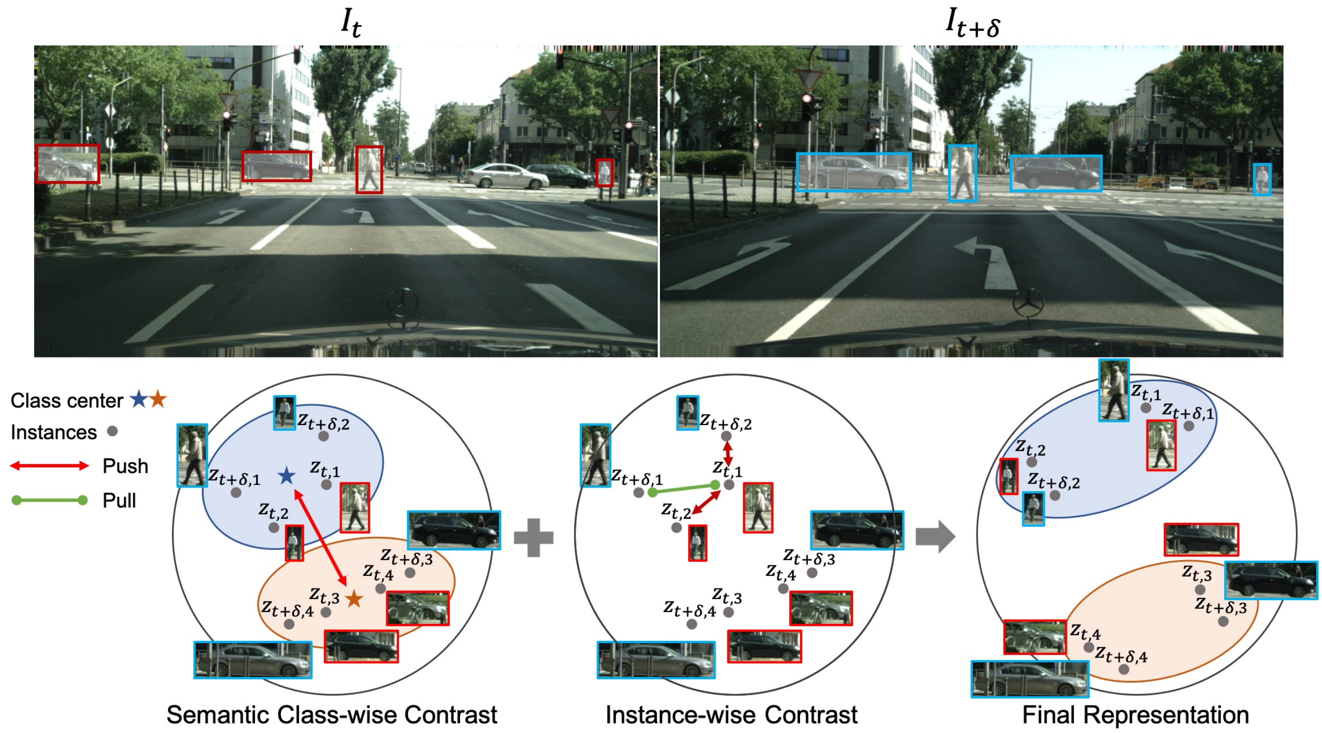


Figure 3: We exploit two different contrastive forces for segment-level temporal correspondence learning: semantic class-wise contrast and instance-wise contrast. The pull-push force at the semantic class-level enforces the model to learn ***grouping***. In the meantime, the pull-push force at the instance-level enforces the model to learn the ***instance discrimination***. These two contrasting forces are applied to link I_t and $I_{t+\delta}$, learning ***temporally invariant features*** as a consequence. As learning proceeds, the feature embeddings desirable for video panoptic segmentation are finally learned. For clarity, only the partial *pull-push* operations are visualized.

where $\mathbb{1}_{[k\neq i \wedge * \neq t]} \in \{0, 1\}$ is an indicator function evaluating to 1 iff $k \neq i$ and $* \neq t$. The τ denotes a temperature parameter, which is set to 0.5. The final contrastive loss is computed across all positive pairs of segments as

$$\mathcal{L}_{contra} = \frac{1}{2N}\sum_{i=1}^{N} l^i_{t,t+\delta} + l^i_{t+\delta,t} \tag{3}$$

Finally, we found that how we contrast the embedding is also important for the performance. For example, if we only contrast the segment embeddings computed from the semantic segmentation label, the different instances in the same semantic class are not distinguished, which degrades the instance tracking accuracy. On the other hand, using only the *thing* instance masks leads to low classification accuracy as different segment embeddings of the same semantic pushed too far apart. Also, the background regions are not considered in learning the temporal association. To resolve these issues, our key idea is to construct two different graphs using both semantic and instance segmentation labels and apply the contrastive learning simultaneously but independently (see Fig. 3). Our formulation enables class-aware instance contrasting. As training proceeds, the features suitable for panoptic segmentation are learned. The loss for the segment-level correspondence learning is defined as $\mathcal{L}_{\text{segment}} = \mathcal{L}_{\text{contra\_sem}} + \mathcal{L}_{\text{contra\_inst}}$, where $\mathcal{L}_{\text{contra\_sem}}$ and $\mathcal{L}_{\text{contra\_inst}}$ denote the contrastive loss (Eq. (3)) computed from the graphs using semantic segmentation and instance segmentation annotations, respectively.

3.2.2 Pixel-level Correspondence

A dense temporal correspondence learning is often achieved by using a photometric loss. The loss is mainly used for flow estimation [18, 39] or temporal stability of output frames [23, 25, 34]. It is calculated by warping the current frame to the next frame through the optical flow and computing the warping error. Recent studies [9, 30] generalize the photometric loss to the logit space, i.e., segmentation output, aiming at learning a segmentation model that predicts temporally consistent labels. However, the photometric loss assumes pixel independence and thus is challenging to model complex correlations over pixels. This may overlook learning important higher-level temporal structures in a video, such as a segment shape changes through time.

To this end, we introduce a concept of *tube* that is constructed by connecting the segment masks along the time-axis. The model attempts to globally minimize the mismatch between the prediction tubes and the ground-truth tubes during training. The proposed tube matching loss is applied pixel-wise while allowing the model to capture important temporal structures over the pixels. We show that it outperforms the warping loss in learning an accurate segmentation model. However, we find that warping loss can complement the tube-matching loss, and the best performance can be achieved when using them together (see Table 4). Therefore, we propose to use them jointly to achieve strong pixel-level temporal correspondence learning. In the following, we detail how we define the final loss function for learning the pixel-level correspondences.

Warping loss. Given a pair of video frames I_t and $I_{t+\delta}$, we first feed them into the flow estimation network, e.g., FlowNet [18], and obtain the optical flow of each pixel from the current frame to the next frame, $F_{t\Rightarrow t+\delta}$. To avoid penalizing the pixels in the invisible regions, we estimate the occlusion map as $O_{t+\delta\Rightarrow t} = \exp(-\alpha||I_t - \hat{I}_{t+\delta}||_2)$, where $\hat{I}_{t+\delta}$ is the frame $I_{t+\delta}$ warped by the flow $F_{t\Rightarrow t+\delta}$ and α is set to 50. By the nature of the convolutional neural network, the output logit is spatially aligned with the input frame. We thus directly use the downsampled version[6] of the same flow (F') and occlusion map (O') to constrain the logits. The panoptic logits s_t and $s_{t+\delta}$ are acquired by feeding the input frames into the segmentation model. The warping loss is defined as

$$\mathcal{L}_{warp} = \sum_{x,y} O'^{x,y}_{t\Rightarrow t+\delta} \left\| s^{x,y}_t - \hat{s}^{x,y}_{t+\delta} \right\|_2, \tag{4}$$

where $s^{x,y}$ is the logit vector at location (x, y) and $\hat{s}_{t+\delta}$ is the logit $s_{t+\delta}$ warped by the flow $F'_{t\Rightarrow t+\delta}$.

Tube-matching loss. Given a pair of panoptic logits s_t and $s_{t+\delta}$, we first apply softmax layer along the the channel-axis. Then, each channel of the panoptic logit will encode specific segment masks, and the values will range between 0 and 1. We temporally concatenate the same segment masks in two different panoptic logits (see Fig. 1). Formally, for an segment i, its tube, T_i, can be constructed as

$$\begin{aligned} \dot{s} &= Softmax(s) \\ T_i &= [\dot{s}_{t,i}, \dot{s}_{t+\delta,i}] \end{aligned} \tag{5}$$

where $\dot{s}_{t,i}$ and $\dot{s}_{t+\delta,i}$ denote the mask of segment i in the frame I_t and $I_{t+\delta}$, respectively. For the tube matching, we adopt the dice coefficient [**?**], D, which is insensitive to the number of foreground and background pixels and can have balanced importance on every segment regardless of their size. It is defined as,

$$D(p, q) = \frac{2\sum(p \cdot q)}{\sum p^2 + \sum q^2}. \tag{6}$$

As the dice coefficient has a range from 0 to 1, the tube matching loss function is calculated as

$$\mathcal{L}_{tube} = \frac{1}{N}\sum_i 1 - D(T_i, \tilde{T}_i), \tag{7}$$

where $\tilde{T}$ is the corresponding binary ground-truth mask tube. N is the total number of the traceable segments.

The loss for the pixel-level correspondence learning is composed of these two losses as $\mathcal{L}_{pixel} = \mathcal{L}_{\text{warp}} + \mathcal{L}_{\text{tube}}$.

4. Experiments

This section presents the experimental results on the two video-level datasets, *VIPER*, and *Cityscapes-VPS*. We adopt the evaluation metrics of VPQ [24], which reflect both image-level prediction and cross-frame association performance.

4.1. Data

We use recently presented video panoptic segmentation datasets: VIPER and Cityscapes-VPS.

- **VIPER**: We follow the public train / val split. For evaluation, we include all the validation videos from *day* scenario and use the first 50 frames of each video: setting a total of 600 frames.
- **Cityscapes-VPS**: We follow the public train / val / test split[7]. Each video consists of 30 consecutive frames, with every 5 frames paired with the ground-truth annotations. All 30 frames are predicted for each video, and only the 6 frames with the ground-truth are evaluated.

4.2. Quantitative Results

We consider VIPER as a primary evaluation dataset, based on its high quantity and quality of the annotation. We mostly conduct experiments with this benchmark.

Baselines. We compare our approach to the state-of-the-art VPSNet [24]. The followings are the detailed information of the baseline models used in experiments.

- **VPSNet-Track**: A basic video panoptic segmentation model, which adds tracking head [48] on top of the image panoptic segmentation model, UPSNet [47].
- **VPSNet-FuseTrack**: An advanced model, which additionally supports temporal feature aggregation [50] and attentional space-time feature fusion [45]. This model holds current state-of-the-art VPQ scores.
- **VPSNet-SiamTrack**: We use VPSNet-Track as a base segmentation model and train it with a pair of temporally distant frames. As there is no difference in model designs, we can see the performance improvement of **SiamTrack** over the **Track** totally comes from the enhanced model representations after the training.

[6] The spatial size of the logit is often smaller than original input resolution, e.g., 1/4 or 1/8.

[7] 400 training, 50 validation, and 50 test videos

VPSNet-variants	Segment	Pixel		Temporal window size				VPQ	FPS
on VIPER	$\mathcal{L}_{\text{segment}}$	$\mathcal{L}_{\text{warp}}$	$\mathcal{L}_{\text{tube}}$	k = 0	k = 5	k = 10	k = 15		
Track				48.1 / 38.0 / 57.1	49.3 / 45.6/ 53.7	45.9 / 37.9 / 52.7	43.2 / 33.6 / 51.6	46.6 / 38.8 / 53.8	5.1
FuseTrack [24]				49.8 / 40.3 / 57.7	51.6 / 49.0 / 53.8	47.2 / 40.4 / 52.8	45.1 / 36.5 / 52.3	48.4 / 41.6 / 53.2	1.6
SiamTrack-*Base*				49.5 / 39.1 / 58.1	50.8 / 46.6 / 54.0	46.7 / 38.8 / 53.0	44.5 / 34.5 / 52.4	47.9 / 39.8 / 54.3	5.1
SiamTrack	✓			50.7 / 41.7 / 58.2	52.6 / 50.8 / 54.2	48.4 / 42.9 / 53.3	46.5 / 39.3 / 52.8	49.6 / 43.7 / 54.6	5.1
SiamTrack		✓		50.3 / 40.4 / 58.6	51.3 / 47.6 / 54.3	47.3 / 40.0 / 53.3	45.4 / 36.4 / 52.9	48.6 / 41.1 / 54.8	5.1
SiamTrack			✓	50.7 / 41.0 / 58.6	52.4 / 49.6 / 54.6	48.5 / 42.5 / 53.5	46.7 / 39.3 / 52.8	49.6 / 43.1 / 54.9	5.1
SiamTrack-*Ours*	✓	✓	✓	**51.1** / 42.3 / 58.5	**53.4** / 51.9 / 54.6	**49.2** / 44.1 / 53.5	**47.2** / 40.3 / 52.9	**50.2** / 44.7 / 55.0	5.1

Table 1: Video panoptic segmentation results on VIPER validation set. Each cell contains VPQ / VPQ$^{\text{Th}}$ / VPQ$^{\text{St}}$ scores.

VPSNet-variants	Temporal window size				VPQ	FPS
on Cityscapes-VPS *val.*	k = 0	k = 5	k = 10	k = 15		
Track	63.1 / 56.4 / 68.0	56.1 / 44.1 / 64.9	53.1 / 39.0 / 63.4	51.3 / 35.4 / 62.9	55.9 / 43.7 / 64.8	4.5
FuseTrack [24]	64.5 / 58.1 / 69.1	57.4 / 45.2 / 66.4	54.1 / 39.5 / 64.7	52.2 / 36.0 / 64.0	57.2 / 44.7 / 66.6	1.3
SiamTrack-*Base*	63.2 / 57.0 / 67.7	56.2 / 42.6 / 66.3	53.2 / 38.3 / 65.2	51.5 / 33.7 / 64.5	56.0 / 42.9 / 65.8	4.5
SiamTrack-*Ours*	**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	4.5

VPSNet-variants	Temporal window size				VPQ	FPS
on Cityscapes-VPS *test*	k = 0	k = 5	k = 10	k = 15		
Track	63.1 / 58.0 / 66.4	56.8 / 45.7 / 63.9	53.6 / 40.3 / 62.0	51.5 / 35.9 / 61.5	56.3 / 45.0 / 63.4	4.5
FuseTrack [24]	**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-*Base*	63.2 / 58.6 / 66.1	57.1 / 46.9 / 63.6	54.0 / 41.8 / 62.6	52.1 / 37.9 / 61.5	56.6 / 46.4 / 63.4	4.5
SiamTrack-*Ours*	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	4.5

Table 2: Video panoptic segmentation results on Cityscapes-VPS validation (top) and test (bottom) set with our VPSNet-variants. Each cell contains VPQ / VPQ$^{\text{Th}}$ / VPQ$^{\text{St}}$ scores.

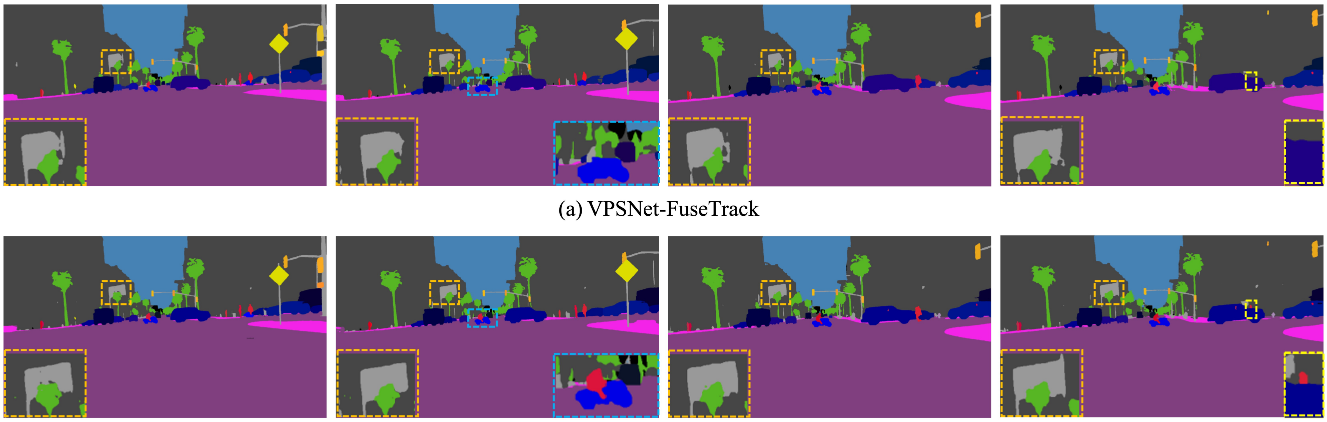
4.2.1 Main Results

VIPER. First, we demonstrate the impact of the proposed learning objectives. The results are summarized in Table 1. The baseline is a siamese model without using any temporal correspondence learning at train time. Upon this baseline, we apply our proposals. As shown in the table, we find that all the loss functions improve the baseline with significant margins, showing their effectiveness. Specifically, the VPQ scores increased by 1.7%, 0.7%, and 1.7% for $\mathcal{L}_{\text{contra}}$, $\mathcal{L}_{\text{warp}}$, and $\mathcal{L}_{\text{tube}}$, respectively. Note that our learning objectives provide the model to perform well on the more challenging regime, i.e., the temporal window size of $k >= 5$. The best VPQ scores were achieved when the two objectives, segment-level and pixel-level temporal correspondence, are enforced together. This result implies that correspondence learning at both segment-level and pixel-level synergize each other to learn more discriminative video features. Compared to the state-of-the-art, VPSNet-FuseTrack, our final model achieves much higher VPQ scores (48.4 VPQ vs. **50.2 VPQ**). Moreover, as our model does not rely on temporal aggregation [50] and fusion [45] strategies, we enjoy faster inference speed (1.6FPS vs. **5.1FPS**) and lower memory usage (5.5G vs. **3.7G**) than the VPSNet-FuseTrack. This shows great potential of our learning framework for practical applications on high-speed and low throughput scenarios.

Cityscapes-VPS. We also benchmark our model on Cityscapes-VPS *val* and *test*. The results are summarized in Tab. 2. We observed a consistent tendency, where our full VPSNet-SiamTrack model achieves +1.3%VPQ and +1.2%VPQ higher than the baseline for *val* and *test*, respectively. Also, it achieves better performance compared to the state-of-the-art, while running in much faster inference speed (3×) and using lower memory usage (5.5G vs. **3.7G**). We see the relatively lower performance improvement than the VIPER mainly comes from the difference in annotation quality [8].

Analysis on $\mathcal{L}_{\text{segment}}$. The model learns coarse segment-level temporal correspondence by matching their embeddings across frames. The temporal association is achieved by contrastive learning, and for successful learning, it is essential to set positive and negatives properly [5, 6, 15]. Here, we consider the same segment's embeddings in the different frames as positive and others as negatives. During learning, our approach exploits both the semantic and instance segmentation labels. The results are summarized in Tab. 3. The instance-wise contrast ($\mathcal{L}_{\text{contra\_inst}}$) enables the learning of visual features suitable for tracking. However, as learning is unaware of the semantic-class, the instance features in the same class may vary significantly (49.6VPQ → 48.4VPQ). Moreover, the background is ignored. On the other hand, the semantic class-wise contrast ($\mathcal{L}_{\text{contra\_sem}}$)

[8] We found several critical annotation errors in the Cityscapes-VPS; incorrect track-ids, wrong class labels, and low-quality masks. They may lead to noisy training signal (e.g., wrong temporal association of the segments or inaccurate boundaries and shapes).



(a) VPSNet-FuseTrack

(b) VPSNet-SiamTrack

Figure 4: The qualitative results on VIPER. Frames are sampled uniformly. The insets show enlarged views of the dotted boxes.

Loss	Segment-lvl matching $\mathcal{L}_{\text{contra\_inst}}$	$\mathcal{L}_{\text{contra\_sem}}$	VPQ	VPQ$^{\text{Th}}$	VPQ$^{\text{St}}$
			47.9	39.8	54.3
	✓		48.4	40.6	54.2
		✓	47.6	39.6	54.1
$\text{L}_{\text{segment}}$	✓	✓	**49.6**	**43.7**	**54.6**

Table 3: Ablation studies on segment-level correspondence loss using VIPER.

Loss	Pixel-lvl matching $\mathcal{L}_{\text{warp}}$	$\mathcal{L}_{\text{tube}}$	VPQ	VPQ$^{\text{Th}}$	VPQ$^{\text{St}}$	TC [30]
			47.9	39.8	54.3	78.3
	✓		48.6	41.1	54.8	79.3
		✓	49.6	43.1	54.9	80.8
L_{pixel}	✓	✓	**49.9**	**43.8**	**55.0**	**82.2**

Table 4: Ablation studies on pixel-level correspondence loss using VIPER.

enforces the model to learn grouping the same semantic features. However, without instance-level contrasting, it is easily trapped into degeneracy (49.6VPQ → 47.6VPQ). To this end, we present to use both the instance-level and semantic class-level contrasting together. The strong empirical results in Table 3 confirms that the features suitable for video panoptic segmentation is learned.

Analysis on $\mathcal{L}_{\text{pixel}}$. The fine pixel-level correspondence is learned with the combination of warping loss ($\mathcal{L}_{\text{warp}}$) and tube matching loss ($\mathcal{L}_{\text{tube}}$). We evaluate their impacts on both segmentation quality and output temporal smoothness using VPQ and TC [30], respectively. The results are summarized in Table 4. As shown in the table, we can clearly see that the tube matching loss outperforms warping loss significantly, both on the segmentation quality (VPQ) and the smoothness of output videos (TC). This backs our claim that modeling higher-level temporal structures over the pixels is more beneficial. Also, we observe that employing both the tube matching loss and the warping loss provides the best VPQ and TC scores. This implies that both are complementary in constructing an accurate and temporally stable VPS model, showing the great advantage of our final loss form.

Analysis on time gap δ. We also study the impact of the time gap, δ, which allows the siamese model to exploit temporally distant frames during training. We conduct experiments with three different values of 5, 10, 15. On VIPER, each provided a final VPQ score of 49.8, 50.2, and 49.9, respectively. We thus use $\delta \in \{-10, 10\}$.

4.3. Qualitative Results

In Fig. 4, we compare our final model VPSNet-SiamTrack with VPSNet-FuseTrack [24] qualitatively. The blue and yellow dotted boxes in the Figure 4 show the segment-level smoothness (i.e., the rider and pedestrian are continuously tracked and segmented over time). Meanwhile, the orange dotted box demonstrates the pixel-level smoothness (i.e., the shape and the boundary of the flag are temporally consistent). These two different temporal stability priors are encoded into the feature representations during the training.

5. Conclusion

In this paper, we generalize the temporal correspondence learning to every segment in a video. To learn their temporal association, we present two novel objective functions with an efficient learning framework. With our learning strategy, a per-frame inference model can outperform previous state-of-the-art while running in a fraction of time. Last but not least, this is the first time to propose a supervised contrastive learning method for learning dense temporal associations. Based on our proposals and their empirical results, we hope more effective video-specific supervisory signals being presented in the future.

Acknowledgements This work was supported in part by Samsung Electronics Co., Ltd (G01200447). Sanghyun Woo and Dahun Kim are supported by Microsoft Research Asia Fellowship.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15535–15545, 2019. 2, 3

[2] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9739–9748, 2020. 1

[3] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 2, 3

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 3, 7

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2, 3, 7

[7] 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. 1

[8] 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

[9] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *AAAI*, 2020. 1, 2, 5

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017. 3

[11] Mohsen Fayyaz, Mohammad Hajizadeh Saffar, Mohammad Sabokrou, Mahmood Fathy, Reinhard Klette, and Fay Huang. Stfcn: spatio-temporal fcn for semantic video segmentation. *arXiv preprint arXiv:1608.05971*, 2016. 1, 2

[12] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video cnns through representation warping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4453–4462, 2017. 1, 2

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2, 3

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2, 3, 7

[16] 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. 3

[17] Juana Valeria Hurtado, Rohit Mohan, and Abhinav Valada. Mopt: Multi-object panoptic tracking. *arXiv preprint arXiv:2004.08189*, 2020. 3

[18] 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. 2, 3, 5, 6

[19] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[20] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8866–8875, 2019. 1, 2

[21] Xiaojie Jin, Xin Li, Huaxin Xiao, Xiaohui Shen, Zhe Lin, Jimei Yang, Yunpeng Chen, Jian Dong, Luoqi Liu, Zequn Jie, et al. Video scene parsing with predictive feature learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5580–5588, 2017. 1, 2

[22] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 3, 4

[23] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019. 5

[24] 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, 6, 7, 8

[25] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 170–185, 2018. 5

[26] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *Advances in Neural Information Processing Systems*, pages 318–328, 2019. 3

[27] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5997–6005, 2018. 1, 2

[28] 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

[29] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010. 2, 3

[30] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *European Conference on Computer Vision*, 2020. 1, 2, 5, 8

[31] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6819–6828, 2018. 1, 2

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 3

[33] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 821–830, 2019. 3

[34] Kwanyong Park, Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Preserving semantic and temporal consistency for unpaired video-to-video translation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1248–1257, 2019. 5

[35] Lorenzo Porzi, Markus Hofinger, Idoia Ruiz, Joan Serrat, Samuel Rota Bulo, and Peter Kontschieder. Learning multi-object tracking and segmentation from automatic annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6846–6855, 2020. 3, 4

[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. 3

[37] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 3

[38] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *European Conference on Computer Vision*, pages 852–868. Springer, 2016. 1, 2

[39] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 2, 3, 5

[40] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1420–1429, 2016. 2, 3

[41] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2805–2813, 2017. 2, 3

[42] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1308–1317, 2019. 3

[43] Qiang Wang, Jin Gao, Junliang Xing, Mengdan Zhang, and Weiming Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017. 2, 3

[44] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4854–4863, 2018. 2, 3

[45] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2, 6, 7

[46] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019. 3

[47] 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. 6

[48] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5188–5197, 2019. 1, 2, 3, 6

[49] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020. 1

[50] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017. 1, 2, 3, 6, 7

[51] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2349–2358, 2017. 1, 2