Solve the Puzzle of Instance Segmentation in Videos: A Weakly Supervised Framework with Spatio-Temporal Collaboration

Liqi Yan, Qifan Wang, Siqi Ma, Jingang Wang, Changbin Yu*

Abstract—**Instance segmentation in videos, which aims to segment and track multiple objects in video frames, has garnered a flurry of research attention in recent years. In this paper, we present a novel weakly supervised framework with Spatio-Temporal Collaboration for instance Segmentation in videos, namely STC-Seg. Concretely, STC-Seg demonstrates four contributions. First, we leverage the complementary representations from unsupervised depth estimation and optical flow to produce effective pseudo-labels for training deep networks and predicting high-quality instance masks. Second, to enhance the mask generation, we devise a puzzle loss, which enables end-to-end training using box-level annotations. Third, our tracking module jointly utilizes bounding-box diagonal points with spatio-temporal discrepancy to model movements, which largely improves the robustness to different object appearances. Finally, our framework is flexible and enables image-level instance segmentation methods to operate the video-level task. We conduct an extensive set of experiments on the KITTI MOTS and YT-VIS datasets. Experimental results demonstrate that our method achieves strong performance and even outperforms fully supervised TrackR-CNN and MaskTrack R-CNN. We believe that STC-Seg can be a valuable addition to the community, as it reflects the tip of an iceberg about the innovative opportunities in the weakly supervised paradigm for instance segmentation in videos.**

Index Terms—**Video instance segmentation, weakly supervised learning, multi-object tracking and segmentation**

I. Introduction

The importance of the weakly supervised paradigm cannot be overstated, as it permeates through every corner of recent advances in computer vision [1]–[3] to reduce the annotation cost [1], [4]. In contrast to object segmentation [5]–[8], for instance segmentation in videos [9]–[11], dense annotations need to depict accurate instance boundaries as well as object temporal consistency across frames, which is extremely labor-intensive to build datasets at scale required to train a deep network. Although a large body of works on weakly supervised image instance segmentation have been discussed in literature [12]–[14], the exploration in video domain remains largely unavailable until fairly recently [5], [15]–[18]. Therefore, understanding and improving the weakly supervised methods of instance segmentation in videos are the key enablers for future advances of this critical task in computer vision.

Developing a weakly supervised framework is a challenging task. One core objective is to devise reliable **pseudo-labels** and **loss function** to perform effective supervision [12], [19]. To date, a popular convention is to use class labels produced by Class Activation Map (CAM) or its variants [20], [21] to supervise image instance segmentation [13], [14], [22]–[24]. However, the CAM-based supervision signal may capture spurious dependencies in training due to two daunting issues: 1) It can only identify the most salient features on object regions, which often lose the overall object structures, resulting in partial instance segmentation [25]–[27]; 2) It cannot separate overlapping instances of the same class and generally lose the capacity to describe individual targets, when dealing with multiple instances present in an image [22], [28], [29]. The challenge is further compounded by instance appearance changes caused by occlusion or truncation [30], [31]. Thus, though CAM is outstanding in semantic segmentation, it does not perform well in instance segmentation. *Under the circumstance, there is a necessity to explore novel weakly supervised approaches with more effective pseudo-labels for video-level instance segmentation.*

Aside from the CAM-based family, a line of research has attempted to tackle image instance segmentation with box-level annotations [32]–[35]. Albeit achieving improvements over CAM-based solutions, they generally have complicated training pipelines, which incur a large computational budget and long supervision schedule. To address this issue, a recent work, BoxInst [36] introduces a simple yet effective mask loss for training, including a projection term and an affinity term. The first term minimizes the discrepancy between the horizontal and vertical projections of the predicted mask and the ground-truth box. The second term is to identify confident pixel pairs with the same label to explore the instance boundary. With the same supervision level, BoxInst achieves significant improvement over the prior efforts using box annotations [37]–[39]. *This successful exploration highlights the importance of loss function to train deep networks in a weakly supervised fashion for the segmentation task.*

On the basis of the preceding lessons, one could argue that box-supervised instance segmentation in videos is feasible. *In*

This work is in part supported by the Shandong Provincial Natural Science Fund (2022HWYQ-081) and the National Science Foundation of China Key Project (U21A20488).

L. Yan is enrolled at Westlake Institute for Advanced Study, Fudan University, China (e-mail: lqyan18@fudan.edu.cn)

Q. Wang is with Meta AI, USA (e-mail: wqfcr@fb.com).

S. Ma and L. Yan are also with School of Engineering, Westlake University, China (e-mail: {yanliqi,masiqi}@westlake.edu.cn).

J. Wang is with Meituan, China (e-mail: wangjingang02@meituan.com).

C. Yu is with College of Artificial Intelligence and Big Data for Medical Science, Shandong First Medical University & Shandong Academy of Medical Sciences, China, and adjunct with the Institute for Intelligent Robots, Fudan University, China (e-mail: Yu_lab@sdfmu.edu.cn).

* is corresponding author.

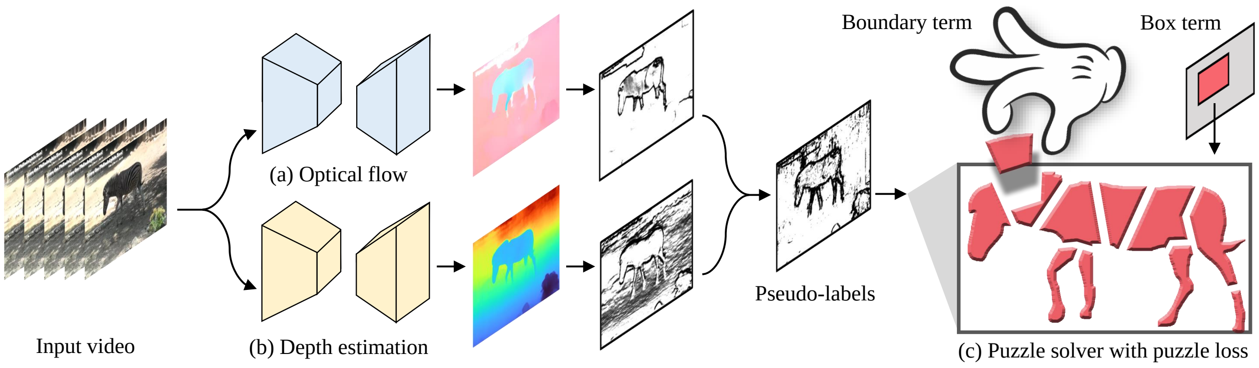


Fig. 1. **Working pipeline of STC-Seg.** The pseudo-labels are generated from spatial and temporal signals, which capture the instance boundary with more accurate edges. Our puzzle solver supervises mask predictions to assemble each sub-region mask together to match the shape of the target with box annotations.

view of the nature of video data, our conjecture is that one can leverage the rich spatio-temporal information in video to develop reliable pseudo-labels for enhancing the box-level supervision. In particular, optical flow captures the temporal motion among instances which ensures the same instances have similar flow vectors (Fig. 1a), while depth estimation provides the spatial relation between instance and background (Fig. 1b). We leverage the complementary representation of ***spatio-temporal signals*** to produce high-quality pseudo-label to supervise instance segmentation in videos. To enable effective training with the proposed pseudo-labels, we propose a novel ***puzzle loss*** that organizes learning in a manner compatible with box annotations, including a boundary term and a box term. The two terms collaboratively resolve the puzzle of how to assemble suitable sub-region masks that match the shape of the instance, facilitating the trained model to be boundary sensitive for fine-grained prediction (Fig. 1c). Furthermore, in contrast to previous efforts [10], [11], which use simple matching algorithms for tracking, we introduce an ***enhanced tracking*** module that tracks diagonal points across frames and ensures spatio-temporal consistency for instance movement. To establish the conjecture, our work essentially delivers the following contributions:

- We develop a Spatio-Temporal Collaboration framework for instance Segmentation (STC-Seg) in videos, which leverages the complementary representations of depth estimation and optical flow to produce high-quality pseudo-labels for training the deep network.
- We design an effective puzzle loss to assemble mask predictions on each sub-region together in a self-supervised manner. A strong tracking module is implemented with spatio-temporal discrepancy for robust object appearance changes.
- The flexibility of our STC-Seg enables weakly supervised instance segmentation and tracking methods to have the capacity to train fully supervised segmentation methods.
- We conduct extensive experiments and demonstrate that our method is competitive with the state-of-the-art system [17] and outperforms fully supervised MaskTrack R-CNN [11] and TrackR-CNN [10].

II. Related Work

Although weakly supervised instance segmentation in videos is relatively under-studied, this section summarizes the recent advances in the related fields regarding weakly supervised instance segmentation, box-supervised methods, and segmenting in videos [14], [23], [24], [33], [36], [40].

A. Fully Supervised Instance Segmentation

In the past decade, various fully supervised image instance segmentation methods have been proposed. These approaches can generally be divided into two categories: two-stage and single-stage approaches. Two-stage methods [41]–[44] typically generate multiple object proposals in the first stage and predict masks in the second stage. While two-stage methods achieve high accuracy with large computational cost, single-stage approaches [45]–[50] employ predictions of bounding-boxes and instance masks at the same time. For example, SipMask [49] proposes a novel light-weight spatial preservation module that preserves the spatial information within a bounding-box. BlendMask [50] is based on the fully convolutional one-stage object detector (FCOS) [51], incorporating rich instance-level information with accurate dense pixel features. However, all these methods are built upon accurate human-labeled mask annotations, which requires far more human annotators than box annotations. In contrast, our method uses only box annotations instead of mask annotations, and thus dramatically reduces labeling efforts.

B. Weakly-supervised Instance Segmentation

Using class labels to extract masks from CAMs or similar attention maps has gained popularity in training weakly supervised instance segmentation models [14], [23], [40], [52]. However, CAM-based supervision is not intrinsically suitable for the instance segmentation task as it cannot provide accurate information regarding individual objects, which potentially causes confusion in prediction [22], [28], [30], [31]. One closely related work is flowIRN [17], which uses the flow fields as the extra supervision signal to operate training. Our technique is conceptually distinct in three folds: 1) flowIRN only uses flow field to generate pseudo-labels and thus fails to fully exploit the spatio-temporal representations.

In contrast, we leverage the collaborative power of spatio-temporal collaboration to produce high-quality pseudo-labels; 2) flowIRN trains different contributing modules ($i.e.$, CAM and optical flow) dis-jointly, resulting in an ineffective and complicated training pipeline. We propose a puzzle solver that organizes learning through the use of our pseudo-labels with box annotations, enable a fully end-to-end fashion; 3) flowIRN directly adopts exiting tracking method [10], while our tracking module builds on a novel diagonal-point-based approach. More comparison results will be provided in the experiments. To address this issue, we aim to explore the more effective pseudo-labels from spatio-temporal collaboration for weak supervision.

C. Box-supervised Instance Segmentation

Our work is also closely related to the box-supervised instance segmentation methods. At the image level, SDI [12] might be the first box-supervised instance segmentation framework, which utilizes candidate proposals generated by MCG [53] to operate segmentation. In the same vein, a line of recent work [33], [37], [54], [55] formulates the box-supervised instance segmentation by sampling the positive and negative proposals based on the ROIs feature maps. However, using proposals for instance segmentation has redundant representations because a mask is repeatedly encoded at each foreground feature around ROIs. In contrast, our method is proposal-free as we remove the need for proposal sampling to supervise the mask learning. BoxInst [36] is one of the works that is similar to ours. It uses a pairwise loss function to operate training on low-level color features. However, their pairwise loss works in an oversimplified manner that encourages confident pixel neighbors to have similar mask predictions, inevitably introducing noisy supervision. Different from BoxInst, our method produces high-quality pseudo-labels derived from high-level spatio-temporal priors for supervision. To organize learning, we devise a novel puzzle loss to supervise our mask generation to capture accurate instance boundaries with box annotations.

D. Video Segmentation

A series of fully supervised approaches have emerged for segmentation in videos [9], [35], [45], [56]–[60]. For instance, VIS [11] and MOTS [10] both extend Mask R-CNN [41] from images to videos and simultaneously segment and track all object instances in videos. To the best of our knowledge, flowIRN [17] and BTRA [61] may be two of the few that explore weakly supervised learning for the video-level instance segmentation task. FlowIRN [17] trains different contributing modules ($i.e.$, CAM and optical flow) dis-jointly and incurs additional dependencies, resulting in a dense training pipeline. BTRA [61] only box to generate pseudo-labels and thus fails to fully exploit the spatio-temporal representations for the boundary supervision. To maximize synergies for instance segmentation in videos, we propose a weakly supervised spatio-temporal collaboration framework in the paper. Unlike the aforementioned methods which overlook the sub-task of tracking in videos, we implement a strong tracking module to model instance movement across frames by using diagonal points with spatio-temporal information. Compared to the prior efforts [10], [11], [17], our tracking module has a more robust tracking capacity.

E. Spatio-Temporal Collaboration

A bunch of previous works [4], [56], [62]–[67] explore spatio-temporal collaboration to assist visual tasks. For example, P3D ResNet [67] mitigated limitations of deep 3D CNN by devising a family of bottleneck building blocks that leverages both spatial and temporal convolutional filters.

SC-RNN [63] simultaneously captures the spatial coherence and the temporal evolution in spatio-temporal space. ESE-FN [64] captures motion trajectory and amplitude in spatio-temporal space using skeleton modality, which is effective in modeling elderly activities. However, these methods embed spatio-temporal analysis into the entire model, where the spatio-temporal modeling process is required during inference. In our method, the spatio-temporal collaboration is only used as the supervision signal during training, but not needed in the segmentation prediction.

III. STC-Seg Approach

A. Overall Framework

The overall framework of STC-Seg is shown in Fig. 2. During training, the pseudo-labels are first generated with spatio-temporal collaboration. The segmentation model is then jointly learned based on the pseudo-labels and the box labels/annotations via a novel puzzle solver. During inference, we directly perform instance segmentation on input video data without using any extra information ($i.e.$, depth estimation or optical flow). Essentially, our STC-Seg consists of three core components: 1) the spatio-temporal pseudo-label generation, which offers a supervision signal for our training; 2) the puzzle solver, which organizes the training of video instance segmentation models; and 3) the tracking module, which enables robust tracking capacity. We present the details of each component in the following sections.

B. Puzzle Solver with Spatio-temporal Collaboration

1) Pseudo-label Generation.: Most existing works [37], [54], [55] rely solely on optical flow to generate pseudo-label. In this work, we leverage both spatial and temporal signals in our pseudo-label generation pipeline to better capture rich boundary information and effectively distinguish the foreground (the instance) from the background. In particular, our method adopts spatial signal $\boldsymbol{S}^s$ obtained from depth estimation [68], and temporal signal $\boldsymbol{S}^t$ obtained from optical flow [69].

As shown in Fig. 2, we directly employ depth estimation $\boldsymbol{x}^s \in \mathbb{R}^{h \times w \times 1}$ and optical flow $\boldsymbol{x}^t \in \mathbb{R}^{h \times w \times 2}$ as the inputs for our pseudo-label generation module. The above two inputs keep the same resolution $w \times h$ with the input frame, in order to build the pixel-to-pixel correspondence. Each signal $\boldsymbol{x} \in \{\boldsymbol{x}^s, \boldsymbol{x}^t\}$ is then fed into a mini network [19] to compute the contextual similarity at each pixel location for obtaining the spatial and temporal signals, respectively. Given a location

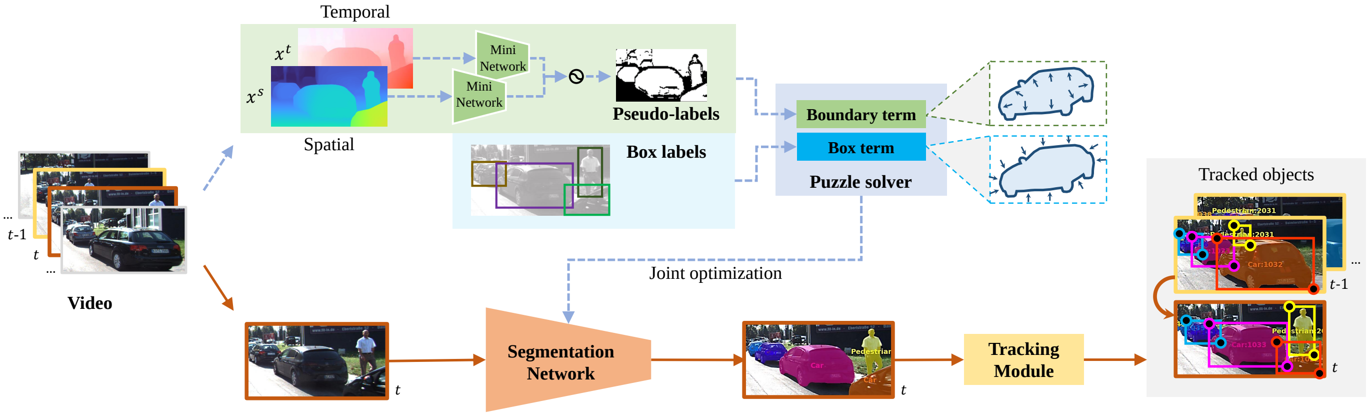
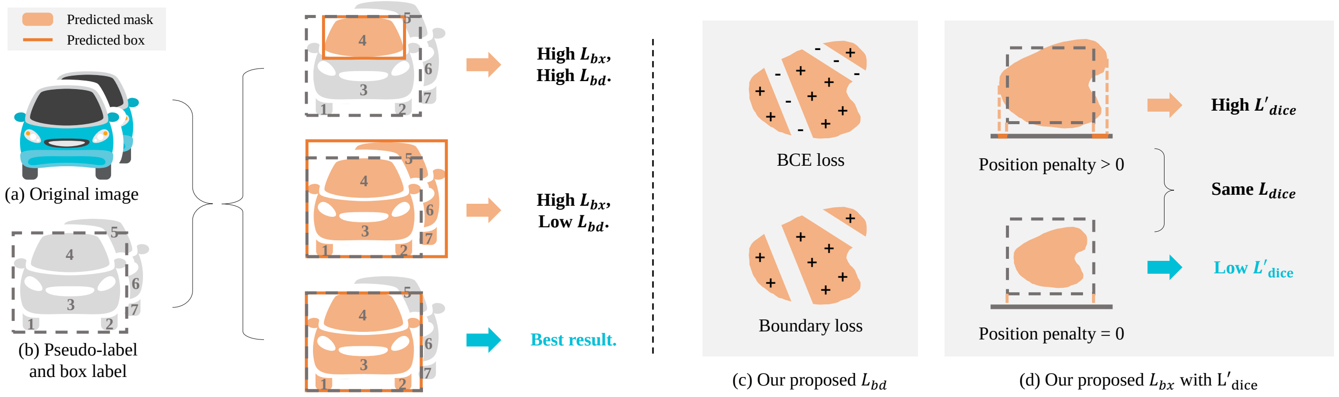


Fig. 2. **The overview of STC-Seg framework**. During training, pseudo-labels from spatio-temporal collaboration and box labels from box annotation are jointly fed into the puzzle solver to learn a unified instance segmentation network. During inference, the learned segmentation network is applied to every frame, followed by a tracking module to perform robust object tracking. Dashed and solid paths are the pipelines for training and inference respectively.



(a) Original image

(b) Pseudo-label and box label

(c) Our proposed L_{bd}

(d) Our proposed L_{bx} with L'_{dice}

Fig. 3. **Demonstration of puzzle solver**. Our puzzle solver performs strong supervision with box annotations and pseudo labels. Albeit the corresponding pseudo labels for one target generally include multiple sub-regions, *i.e.*, sub-region 1-4 in (b), the boundary term and box term in our puzzle loss work collaboratively to supervise the mask prediction for aligning the shape of the instance while being consistent with the ground truth box.

(i,j) on the input $\boldsymbol{x}$, the contextual similarity score $\boldsymbol{S}_{i,j}$ on the corresponding signal $\boldsymbol{S} \in \{\boldsymbol{S}^s, \boldsymbol{S}^t\}$ is computed as:

$$\boldsymbol{S}_{i,j} = \sum_{k_1,k_2} \delta\left(\boldsymbol{w}_{k_1,k_2} \cdot \boldsymbol{x}_{(i+\lambda\cdot k_1),(j+\lambda\cdot k_2)}, \boldsymbol{x}_{i,j}\right) \tag{1}$$

where $k_1, k_2 \in \{-1, 0, 1\}$. $\boldsymbol{w}$ is the dilated kernel, and λ is the dilation rate. $\boldsymbol{\delta}(,)$[1] is the similarity measurement function. For the obtained signals, we have $\boldsymbol{S}^s, \boldsymbol{S}^t \in \mathbb{R}^{h\times w\times 1}$.

To produce the pseudo-label $\boldsymbol{M}$ for training, we leverage the complementary representations of the two signals by fusing them together with a threshold filter:

$$\boldsymbol{M} = (\boldsymbol{S}^s - \phi^s) \wedge (\boldsymbol{S}^t - \phi^t) \tag{2}$$

where ϕ^s, ϕ^t denote the filter factor to determine the salience threshold of each signal on the foreground instances. However, noises may reside on the pseudo-labels and segregate one target instance into multiple sub-regions.

2) Puzzle Solver.: As mentioned above, directly using the pseudo-labels without constraints may result in excessively noisy supervision and suboptimal training outcomes. In comparison to fully supervised information, which can be labeled pixel-by-pixel, solving the puzzle of predicting the imaginary mask is difficult in the weakly supervised fashion. To address this issue, we introduce a novel puzzle solver that organizes learning through the use of our pseudo-labels with box annotations.

Our puzzle solver essentially designs a puzzle loss that operates supervision of mask prediction with two loss terms. The first one is *Boundary term*, which explores all the candidate sub-regions of the target instances to depict their boundaries. The second one is *Box term*, which ensures maximal positions of the predicted mask boundaries can closely stay within the ground truths. *The two terms work collaboratively to solve the puzzle of how to assemble suitable sub-region masks together to match the shape of the instance* (see Fig. 3). Our puzzle solver is to jointly optimize both the boundary term L_{bd} and box term L_{bx} with respect to the network parameters θ:

$$\arg\min_{\theta} L_{pz} = \arg\min_{\theta}(L_{bd} + L_{bx}) \tag{3}$$

Boundary term: With ground truths, fully supervised methods can use binary cross entropy (BCE) loss L_{bce} to supervise the mask generation, which uses both positive samples (the foreground) and negative samples (the background) in training. However, as discussed in Section III-B1, our pseudo-labels are

[1] $\delta\left(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i',j'}\right) = e^{r\cdot\left\|\boldsymbol{x}_{i,j} - \boldsymbol{x}_{i',j'}\right\|_p}$.

noisy references with unwanted inner negative samples inside the object, which would introduce inevitable noises in training. To address this issue, we modify L_{bce} by focusing on learning positive examples to capture the instance boundaries (Fig. 3c). Concretely, given a pixel location (i, j) on the pseudo-labels $\boldsymbol{M}$, its corresponding label $m_{i,j}$ can be $m_{i,j} \in \{0, 1\}$, where 1 denotes the foreground instance and 0 denotes the background. To learn the instance mask generation, our boundary loss only operates learning of the posterior probability $P(\tilde{m}_{i,j}|m_{i,j} = 1)$ from positive samples, where $\tilde{m}_{i,j} \in \{0, 1\}$ is the predicted mask at (i, j). Given the input size $w \times h$, our boundary loss is given by:

$$L_{bd} = -\frac{1}{h \times w} \sum_{j=1}^{w} \sum_{i=1}^{h} m_{i,j} \log P(\tilde{m}_{i,j} = 1) \tag{4}$$

At first glance, only using the positive sampling may not work well in training. However, an important observation is that our pseudo-labels allow the network to effectively learn the dominant representations from the positive examples. With additional box supervision, the boundary loss computation effectively captures the instance boundaries, thus largely eliminating supervision noises.

Box term: To perform box-level supervision, BoxInst [36] adopts dice loss [70], which computes the similarity distance of the predicted bounding boxes and the ground truths. However, a prediction, which is larger or smaller than the ground truth, may have a similar penalty in dice loss computation. Thus, a model supervised by the dice loss tends to generate overly saturated masks that go beyond the boxes. To address this issue, we introduce a position penalty into dice loss L_{dice} to penalize the model for generating a mask that exceeds the box (as shown in Fig. 3d). This penalty term encourages the mask boundary to align with the ground truth box:

$$L^{'}_{dice}(p, g) = \underbrace{\frac{2\sum_{i}^{N} p_i g_i}{\sum_{i}^{N} p_i^2 + g_i^2}}_{\text{Dice loss } L_{dice}} + \underbrace{\frac{\sum_{i}^{N} [\max(p_i - g_i, 0)]^2}{\sum_{i}^{N} g_i^2}}_{\text{Position penalty}} \tag{5}$$

where $p_i \in (0, 1)$ and $g_i \in \{0, 1\}$ are the log-likelihood scores of the prediction and the ground truth respectively. N is the length of the input sequence. The position penalty can be understood as the proportion of the predicted region that exceeds the ground truth region. As shown in Eq. 5, it is clear that there is no position penalty for those points within the ground truth. The final box term can be written as:

$$\begin{aligned} L_{bx}(\tilde{m}, B) = & L^{'}_{dice}(\text{Proj}_x(\tilde{m}), \text{Proj}_x(B)) \\ & + L^{'}_{dice}(\text{Proj}_y(\tilde{m}), \text{Proj}_y(B)) \end{aligned} \tag{6}$$

where $\tilde{m}$ is the predicted instance mask. B is the corresponding box annotations. Proj_x and Proj_y are the projection functions [36], which map $\tilde{m}$ and B onto x-axis and y-axis, respectively. It is worth mentioning that the new L_{bx} effectively rectifies the expanded masks outside the box that are introduced by the L_{bd}. In other words, the L_{bd} allows the model to predict larger masks, while L_{bx} ensures the model predicts precise masks that are consistent with the ground truth boxes. Note that the segmentation generation is independent of pseudo-label generation. The computational cost only increases when calculating the losses, which has the same computational complexity as the MSE loss. Therefore, our method will not introduce additional computation cost.

Our method can also be applied to those tasks with noise supervision, such as target segmentation tasks with inaccurate box labeling or incorrect labels [71]–[74]. For the former cases, we can slightly modify the loss of box term to assign a larger weight to the positive feedback of the intersection area, while assigning the negative feedback outside the intersection area a smaller weight. For the latter cases, we can modify the loss of boundary term by assigning a relatively large weight to the positive item and a small weight to the negative item in the cross entropy. In this way, our loss function is able to deal with more inaccurate box annotations and label predictions. Intuitively, the classification task can be regarded as a regression problem by taking the irrelevant labels as the ”background”, so that the regression boundary can shrink inward on the feature plane until the accurate label boundary is found.

C. Tracking Module

Existing methods [75]–[77] prioritize object position modeling for tracking, which may cause confusion when two objects are extremely occluded or overlapped as shown in Fig. 2. To address this issue, we place a premium on both object size and position modeling in our tracking module. Moreover, the spatio-temporal changes on individual objects should remain within a reasonable range, given the consistency of video object movement across frames. In light of both observations, we introduce a novel tracking module using *diagonal points* with *spatio-temporal discrepancy*.

1) Diagonal Points Tracking: To represent the object position and size, we adopt diagonal points to model the object movement by using the upper-left corner (x_1, y_1) and the lower-right corner (x_2, y_2) of the bounding box. Similar to almost tracking methodology [76], [78], we adopt a recursive *Kalman Filter* and frame-by-frame data association to predict the future location for each tracked object. The movement $\Delta l_o^{t-1 \to t}$ of a tracked object o in the t_{th} frame is used to predict the future location $p^{t+1}(l_o^t) = \Delta l_o^{t-1 \to t} + l_o^t$ of this object at $(t+1)_{th}$, where l_o^t is the location of object o and $\Delta l_o^{t-1 \to t}$ is given by:

$$\Delta l_o^{t-1 \to t} = (x_1^t - x_1^{t-1}, y_1^t - y_1^{t-1}, x_2^t - x_2^{t-1}, y_2^t - y_2^{t-1}) \tag{7}$$

During object tracking, we maintain a dictionary $O_{\leq t} = \{\hat{o}\}^{\hat{K}}$ of $\hat{K}$ tracked objects in former frames. Given K detected objects $O_{t+1} = \{o\}^K$ in $(t+1)_{th}$ frame, our tracking is to build a list of one-to-one matching pairs $\hat{o} = \varphi(o) \in O_{\leq t}$ to minimize the Euclidean distance between ground truth locations l_o^{t+1} of each $o \in O_{t+1}$ and the predicted future locations $p^{t+1}(l^t_{\varphi(o)})$:

$$\arg\min_{\varphi} \sum_{O_{t+1}} < p^{t+1}(l^t_{\varphi(o)}), l_o^{t+1} > \tag{8}$$

2) Bi-greedy Matching: Conventional tracking methods are generally one-directional as they perform a popular greedy search, called *Hungarian Algorithm*, to build the correspondences φ from the previous frame to the current one (*e.g.*, JDE [79], DeepSORT [78], FairMot [80], CenterTrack [77]). However, the position of the same object in the previous frame may not always be the closest one that appeared in the current frame, which causes confusion in tracking. To address this problem, some methods use pre-trained CNN descriptors to distinguish objects [11], [78], but computing features takes too much time and the objects are sometimes very similar. Thus, we consider matching from both directions (i.e., previous-to-current and current-to-previous) and develop a bidirectional greedy matching to output the tracking T_φ as follows (assuming that only DP is used):

Algorithm 1 Bi-greedy matching.

```
Input: $O_{\leq t} = \{\hat{o}\}^G$, $O_{t+1} = \{o\}^K$.
Output: $T_\varphi = \{(o, \varphi(o))\}^K$ for all $o \in O_{t+1}$
$T \leftarrow \emptyset$
$T_\varphi \leftarrow \emptyset$
for all $\hat{o} \in O_{\leq t}$ do
    $o' \leftarrow \arg\min_{o \in O_{t+1}} < p^{t+1}(l^t_{\hat{o}}), l^{t+1}_o >$
    $T \leftarrow T \cup (\hat{o}, o')$
end for
for all $o \in O_{t+1}$ do
    if any $\hat{o}, (\hat{o}, o) \in T$ then
        $\hat{o} \leftarrow \arg\min_{\hat{o}, (\hat{o}, o) \in T} < p^{t+1}(l^t_{\hat{o}}), l^{t+1}_o >$
        $T_\varphi \leftarrow T_\varphi \cup (o, \hat{o})$
    else
        $T_\varphi \leftarrow T_\varphi \cup (o, New\,\hat{o})$
    end if
end for
```

As shown in Algorithm 1, our proposed matching algorithm first finds the nearest instance $o' \in O_{t+1}$ in the current frame for each previous tracked object $\hat{o} \in O_{\leq t}$. There may exist two cases: (a) more than one different previous instance may have the same nearest current instance o, those previous instances are collected as *candidate instances*; (b) it is also possible that some current instances are not marked by any previous instances. In the case (a), for this current instance o, we finally get the matched previous instance $\hat{o}$ by finding the nearest one from those *candidate instances*. In the case (b), those current instances are judged as a new instance. Since $G > K$ in most cases, to run traversal fist on $O_{\leq t}$ is better than O_{t+1} because it focuses on the matched current instances $o' \in O_{t+1}$ rather than previous instances that are not in current frame. In each round of matching, in order to avoid the occluded objects being forgotten, we need to re-match the newly emerged objects. Therefore, we adopt a matching cascade algorithm [78] that gives priority to more frequently appearing objects to ensure those objects that are briefly occluded and disappeared can be re-identified.

3) Occluded Object Multi-stage Matching: The occluded objects often get a low confidence level after passing through the detection algorithm. The existing algorithms only set a single confidence level threshold to divide the correctly detected target and the wrongly detected target. This approach causes the tracking of occluded objects to fail. We introduce a *multi-stage matching* mechanism, that is to set two lower thresholds of confidence scores, and treat the divided high-scoring detection targets and low-scoring detection targets differently, and perform two rounds of matching respectively in turn. In this way, although the confidence obtained by the occluded object position is lower, it can still be successfully matched in the second round of matching.

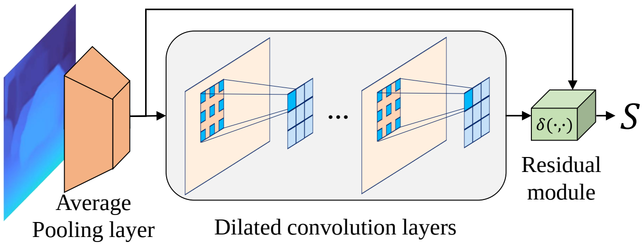


Fig. 4. The architecture of our mini network. We use the dilated convolution layers to capture the spatial or temporal data difference between adjacent pixels. The similarity measurement function $\delta(\cdot, \cdot)$ is implemented by a residual module.

4) Spatio-Temporal Discrepancy: Considering the fact that the spatio-temporal changes on individual objects should retain a reasonable range in videos, we extend the Eq. 8 by adding the spatio-temporal discrepancy for tracking:

$$\begin{aligned} \arg\min_{\tau} \sum_{O_{t+1}} \alpha_1 &< p^{t+1}(l^t_{\varphi(o)}), l^{t+1}_o > \\ +\alpha_2 &< D^t(l^t_{\varphi(o)}), D^{(t+1)}(l^{t+1}_o) > \\ +\alpha_3 &< F^t(l^t_{\varphi(o)}), F^{(t+1)}(l^{t+1}_o) > \end{aligned} \tag{9}$$

where D^t and F^t denotes the depth and optical flow values of the diagonal points for the tracked object o on the t_{th} frame. $\alpha_1, \alpha_2, \alpha_3$ are the trade-off weights that balance these terms. The new objective essentially ensures the tracked objects are aligned with the segmented instances among frames, while at the same time being consistent with their spatio-temporal positions. We demonstrate the improvements of our tracking in Section IV-F.

IV. Experiments

A. Datasets

We evaluate STC-Seg on two benchmarks: KITTI MOTS [10] and YT-VIS [11]. The KITTI MOTS contains 21 videos (12 for training and 9 for validation) focusing on driving scenes. The YT-VIS contains 2,883 YouTube video clips with 131k object instances and 40 categories. On KITTI MOTS, the metrics are HOTA, sMOTSA, MOTSA, and MOTSP from [75]. On YT-VIS, the metrics are: mAP is the mean average precision for IoU between [0.5, 0.9], AP@0.50 and AP@0.75 are average precision with IoU threshold at 0.50 and 0.75, and AR@1 and AR@10 are average recall for top 1 and 10 respectively.

TABLE I

Quantitative results on KITTI MOTS test set. Results for fully supervised methods are retrieved from the MOTS benchmark. For weakly supervised methods WISE and IRN, the results are obtained from their original codes combined with our tracking module. STC-Seg$_{50}$ and STC-Seg$_{101}$ indicate using ResNet-50 and ResNet-101 as backbone respectively. All the baseline methods use ResNet-101 with FPN.

	Methods	Car				Pedestrian			
		HOTA	sMOTSA	MOTSA	MOTSP	HOTA	sMOTSA	MOTSA	MOTSP
Fully	ViP-DeepLab [81]	76.3	81.0	90.7	89.8	64.3	68.7	84.5	82.3
	EagerMOT [82]	74.6	74.5	83.5	89.5	57.6	58.0	72.0	81.5
	MOTSFusion [83]	73.6	74.9	84.1	89.3	54.0	58.7	72.8	81.5
	PointTrack [84]	61.9	78.5	90.8	87.1	54.4	61.4	76.5	80.9
	TrackR-CNN [10]	56.6	66.9	79.6	85.0	41.9	47.3	66.1	74.6
Weakly	WISE [13]	41.8	21.6	39.5	62.9	20.9	18.8	29.1	55.2
	IRN [22]	44.7	25.9	41.1	64.1	22.9	19.1	31.6	56.4
	FlowIRN [17]	50.2	45.1	63.8	71.4	27.5	19.4	35.9	62.7
	PointRend [85]	51.8	49.3	70.6	74.2	28.3	20.0	37.5	64.4
	MOTSNet + Grad-CAM [86]	-	54.6	72.5	76.6	-	20.3	39.7	65.7
	STC-Seg$_{50}$	57.7	66.9	80.7	83.9	46.7	47.4	67.6	73.6
	STC-Seg$_{101}$	**59.6**	**69.2**	**83.3**	**85.1**	**47.5**	**48.6**	**68.3**	**75.8**

TABLE II

Results on YT-VIS validation set. Metrics for SipMask [49] are obtained from its original paper. All other compared results are retrieved from [17]. All methods use ResNet-50 with FPN.

	Methods	mAP	AP@0.5	AP@0.75	AR@1	AR@10
Fully	IoUTracker+ [11]	23.6	39.2	25.5	26.2	30.9
	DeepSORT [78]	26.1	42.9	26.1	27.8	31.3
	MaskTrack R-CNN [11]	30.3	51.1	32.6	31.0	35.5
	SipMask [49]	33.7	54.1	35.8	35.4	40.1
Weakly	WISE [13]	6.3	17.5	3.5	7.1	7.8
	IRN [22]	7.3	18.0	3.0	9.0	10.7
	FlowIRN [17]	10.5	27.2	6.2	12.3	13.6
	STC-Seg$_{50}$	**31.0**	**52.4**	**33.2**	**32.9**	**36.2**

B. Pseudo-Label Generation Network Architecture

In the pseudo-label generation of STC-Seg, unsupervised Monodepth2 [68] and Upflow [69] are adopted for the depth estimation and optical flow respectively. The depth and flow outputs are fed into a mini network. The mini network is composed of a 2D average pooling layer, dilated convolution layers and a residual module, as shown in Fig. 4. In pre-processing, we use a 2D average pooling layer to down-sample the depth and the optical flow data. The kernel size and the stride of the pooling layer are both set to 4 without padding. After the 2D average pooling layer, dilated convolution is applied since it enables networks to have larger receptive fields with just a few layers. The dilation rate λ is set to 2 and the kernel size is set to 3 in our experiments, so the padding size is set to 2 to keep the output size equal to the input size. The weight of the kernel is initialized as $W = [w_{k_1,k_2}]_{3\times3}$, where $w_{k_1,k_2} \in \{0, 1\}$. After the dilated convolution layers, the residual module subtracts the output of the pooling layer from the output of the dilated convolution layers and applies an exponential activation function. The final output of the residual module can be written as $\delta(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i',j'}) = e^{r\cdot||\boldsymbol{x}_{i,j}-\boldsymbol{x}_{i',j'}||_p}$ in Eq. 1, which represents the contextual similarity between locations (i, j) and (i', j') in the frame, where r is the similarity factor. We use the Frobenius norm ($p = 2$) in this contextual similarity calculation. The similarity factor r is set to 0.5 in our experiments.

C. Main Network Architecture

Our segmentation network is crafted on CondInst [87] with a few modifications. Following CondInst, we use the FCOS-based network, which includes ResNet-50/101 backbones [88] with FPN [89], a detection built on FCOS, and dynamic mask heads. For the dynamic mask heads, we use three convolution layers as in CondInst, but we increase the channels from 8 to 16 as in [36], which results in better performance with an affordable computational overhead. Without any network parameter consumption, our tracking module directly performs tracking over the output of the segmentation network.

D. Implementation Details

1) Pseudo-Label Generation: To generate pseudo-labels, unsupervised Monodepth2 [68] and Upflow [69] are adopted for the depth estimation and optical flow respectively. The Monodepth2 [68] is trained on the KITTI stereo dataset [90] when we take experiments on KITTI MOTS. When using the monocular sequences in KITTI stereo dataset for training, we follow Zhou et al.'s [91] pre-processing to remove static frames. This results in 39,810 monocular triplets (three temporally adjacent frames) for training and 4,424 for validation. We use a learning rate of 10^{-4} for the first 15 epochs which is then dropped to 10^{-5} for the remainder. When we take experiments on YT-VIS, the model is pre-trained on NYU Depth dataset [92] with a learning rate 10^{-4}. Following [93], images are flipped horizontally with a 50% chance, and randomly cropped and resized to 384 × 384 to augment the data and maintain the aspect ratio across different input images. Monodepth2 is finetuned on the YT-VIS with a learning rate of 10^{-5} and an exponential decay rate of $\beta_1 = 0.9, \beta_2 = 0.999$. The Upflow [69] is trained on KITTI scene flow dataset [94] when we take experiments on KITTI MOTS. KITTI scene flow dataset [94] consists of 28,058 image pairs (t_{th} frame and $(t-1)_{th}$ frame). Following [95], the learning rate is set to 10^{-4} and the Adam optimizer is used during training. When we take experiments on YT-VIS, the Upflow [69] is pre-trained on FlyingThings [96] for 100k iterations with a batch size of 12, then trained for 100k iterations on FlyingThings3D [96] with a batch size of 6. The learning rate of the above two stages is both set to 1.2×10^{-4}. The model is finetuned on YT-VIS for another 100k iteration with a batch size of 6 and a learning rate of 10^{-4}. Our mini network includes 3 layers of dilated convolutions. For Eq.1, the dilation rate λ is set to be 2 and the kernel size of dilation convolution is set to be 3. The filter factors ϕ^s, ϕ^t are set to be 0.3 and 0.4 respectively.

2) STC-Seg Training and Inference: The STC-Seg is implemented using PyTorch. It is trained with batch size 8 using 4 NVIDIA GeForce GTX 2080 Ti GPUs (2 images per GPU) with 16 workers. During training, the backbone is pre-trained on ImageNet [97]. The newly added layers are initialized as in FCOS [51]. Following CondInst, the input images are resized to have a shorter side [640, 800] and a longer side at a maximum of 1333. The same data augmentation in CondInst [87] is used as well. For KITTI MOTS, we remove 485 frames without any usable annotation so there are 4510 frames left for training. For YTVIS, there are 61341 frames used for training in total. Only left-right flipping is used as the data augmentation during training. Following CondInst [87], the output mask is up-sampled to 1/4 resolution of the input image, and we only compute the loss for top 64 mask proposals per image. For optimization, we use a multi-step learning rate scheduler with a warm-up strategy in the first epoch. In our multi-step learning rate schedule, the base learning rate is set to be 10^{-4}, which starts to decay exponentially after a certain number of iterations up to 2×10^{-5}. In the warm-up epoch, the learning rate is increased linearly from 0 to the base learning rate. The base learning rate is set to be 10^{-4}, which starts to decay exponentially after a certain number of iterations up to 2×10^{-5}. The exact number of iterations varies for each setting as follows: (a) KITTI-MOTS: 10k total iterations, decay begins after 5k iterations; (b) YouTube-VIS: 80k total iterations, decay begins after 30k iterations. The momentum is set to 0.9. The weight decay is set to 10^{-4}, while it is not applied to parameters of normalization layers. In inference, we can directly perform instance segmentation on input video data *without using any extra information*. The hyper-parameters $\alpha_1, \alpha_2, \alpha_3$ are set to be 0.7, 0.2, and 0.1 respectively.

E. Main Results

1) Quantitative Results: On the KITTI MOTS benchmark, we compare our STC-Seg against the state-of-the-art baselines. The results are presented in Table I. It can be seen that our methods achieve competitive results under all evaluation metrics. Our STC-Seg with ResNet-50 significantly outperforms all weakly supervised methods which use a stronger backbone (ResNet-101). In comparison with the fully supervised methods, our method with ResNet-101 can still achieve reasonable results. For example, it outperforms TrackR-CNN [10] by 3.0% on HOTA, 2.3% on sMOTSA, 3.7% on MOTSA and 0.1% on MOTSP for the car class. The results for pedestrian class also are consistent. We further provide comparison results of our STC-Seg with the state-of-the-art baselines on YT-VIS in Table II. It can be seen that our method is competitive with fully supervised MaskTrack R-CNN [11] and SipMask [49]. When comparing with weakly supervised methods, our method outperforms FlowIRN [17], IRN [22] and WISE [13] with significant margins of 20.5%, 23.7%, and 24.7% in terms of mAP metrics respectively.

2) Qualitative Results: We compare qualitative results of our method with those from fully supervised TrackR-CNN [10] and MaskTrack R-CNN [11] on KITTI MOTS and YT-VIS respectively. To demonstrate the advantages of our approach, we select some challenging samples where TrackR-CNN and MaskTrack R-CNN have weaker predictions (see Fig. 5). In the KITTI MOTS examples, the masks generated by Track RCNN have jagged boundaries or leave false negative regions on the borders. In the YT-VIS examples, MaskTrack R-CNN struggles to depict the boundary of instances with irregular shapes (*e.g.*, eagle beak or tail). On the other hand, it is clear that our method captures more accurate instance boundaries.

3) Discussion: The aforementioned results demonstrate the strong performance of STC-Seg in videos. We thus argue that it is effective to use the proposed pseudo-labels and puzzle solver to supervise the mask generation, especially for rigid objects (*e.g.*, vehicles, boats, and planes). However, we encounter notable performance degradation for non-rigid objects (*e.g.*, humans and animals) as the depth and flow estimation become less accurate under the circumstance, which compromises the corresponding pseudo-label generation for supervision. For instances in Fig. 6, there are large false positive regions between pedestrian legs (the top row); our method fails to segment objects in front of the man (the bottom row). The above weak predictions are primarily caused by noisy pseudo-labels incurred by inaccurate depth and flow estimation.

F. Ablation Study

In this section, we investigate the effectiveness of each component in STC-Seg by conducting ablation experiments on KITTI MOTS. For the assessment of our supervision signals and loss terms, we focus on the improvement of mask generation and thus include the average precision (AP) in evaluation. To assess our tracking, we use HOTA, MOTSA, and MOTSP from MOTS [75].

1) Supervision Signals: We show the impact of progressively integrating the depth and flow signals for the pseudo-label generation. As shown in Table III, compared to optical flow, depth has a better performance for car class to produce pseudo-labels when being used alone, while optical flow has a better performance for pedestrian class. In contrast, by

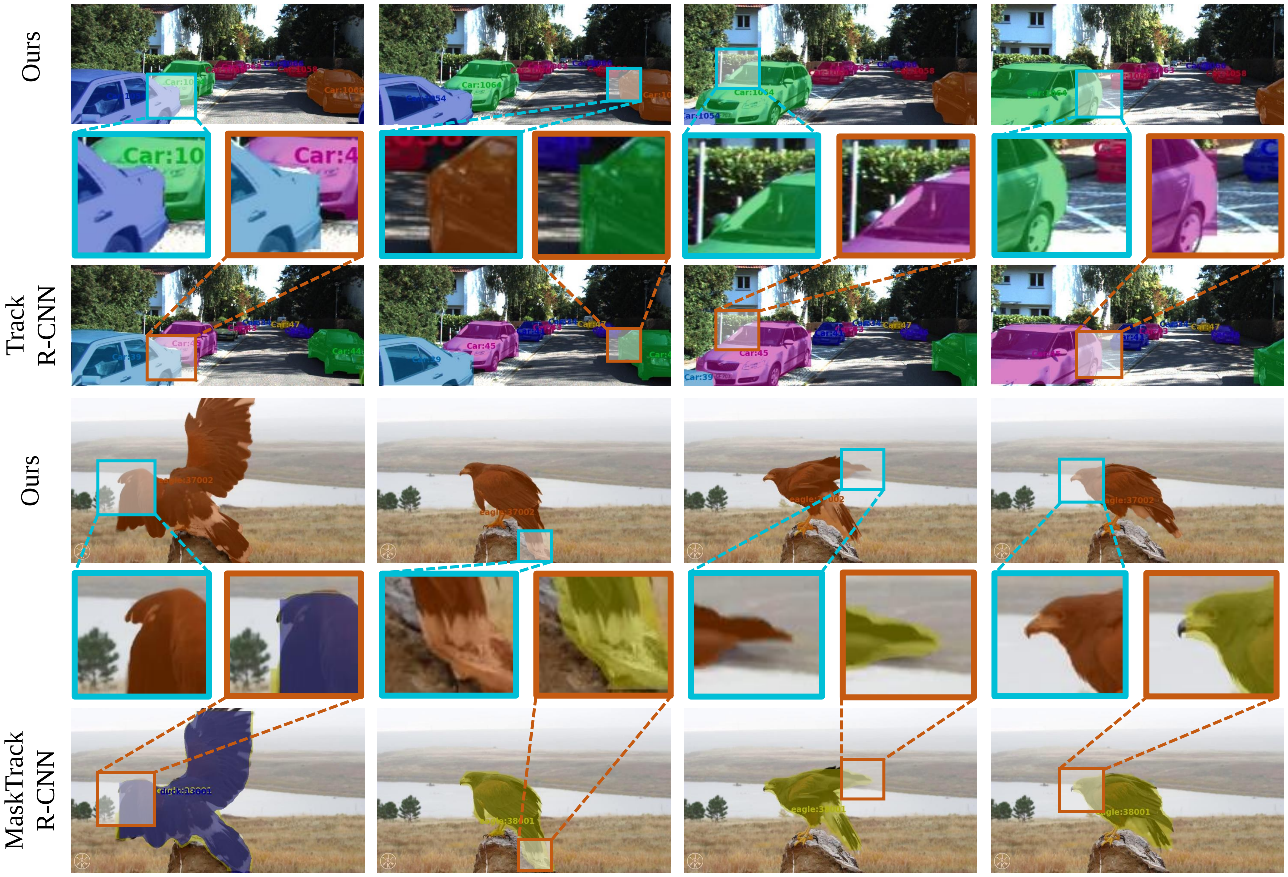


Fig. 5. **Qualitative results** of our STC-Seg in comparison with TrackR-CNN [10] and MaskTrack R-CNN [11] on KITTI MOTS and YT-VIS respectively. All compared methods use ResNet-101 with FPN.

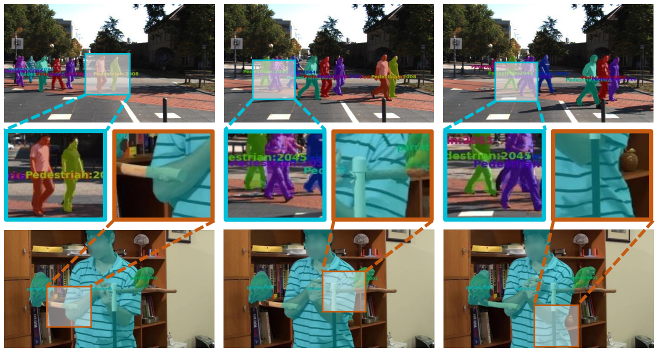


Fig. 6. **Examples of weak predictions from STC-Seg.** The first row is from KITTI MOTS and the second row is from YT-VIS.

leveraging both depth and flow, we develop complementary representations that retain richer and more accurate details of the instance boundary for pseudo-label generation (see Fig. 7a). Therefore, combining the two signals together enables our model to achieve the best performance over the baselines that use them separately.

TABLE III

THE RESULTS OF **USING DIFFERENT SUPERVISION SIGNALS** FROM KITTI MOTS. C AND P DENOTE *car* AND *pedestrian* RESPECTIVELY.

Signal	AP		HOTA		MOTSA	
	C	P	C	P	C	P
Depth	55.6	37.5	57.7	45.0	82.1	66.2
Flow	55.0	37.9	56.8	46.2	80.7	66.9
Depth+Flow	**56.1**	**38.2**	**59.6**	**47.5**	**83.3**	**68.3**

TABLE IV

THE RESULTS OF **USING DIFFERENT LOSS TERMS** FROM KITTI MOTS. C AND P DENOTE *car* AND *pedestrian* RESPECTIVELY.

AP			AP		HOTA		MOTSA	
L_{bx}	L_{bce}	L_{bd}	C	P	C	P	C	P
L_{dice}	×	×	53.3	34.5	53.9	40.3	78.1	61.7
	✓	×	53.8	35.0	54.8	42.2	79.2	63.0
	×	✓	54.7	36.9	56.7	44.1	81.6	65.4
L'_{dice}	×	×	53.6	34.9	54.6	41.9	79.0	62.5
	✓	×	54.2	36.4	55.4	43.2	79.7	64.6
	×	✓	**56.1**	**38.2**	**59.6**	**47.5**	**83.3**	**68.3**

2) Loss Terms: We first only use our box term $L_{bx}(L_{dice})$ without the position penalty to supervise the mask generation as our baseline, followed by the variants supervised by different loss combinations (see Table IV). We achieve immediate improvements of 2.8% (car) and 3.7% (pedestrian) on AP for the model trained only by L_{bce}+$L_{bx}(L_{dice})$ over the baseline. While using BCE loss L_{bce} and our $L_{bx}(L'_{dice})$ for supervision, we can obtain further performance gain over the models trained by L_{bce}+$L_{bx}(L_{dice})$. The best results come from the model trained by our puzzle loss L_{bd}+$L_{bx}(L'_{dice})$,

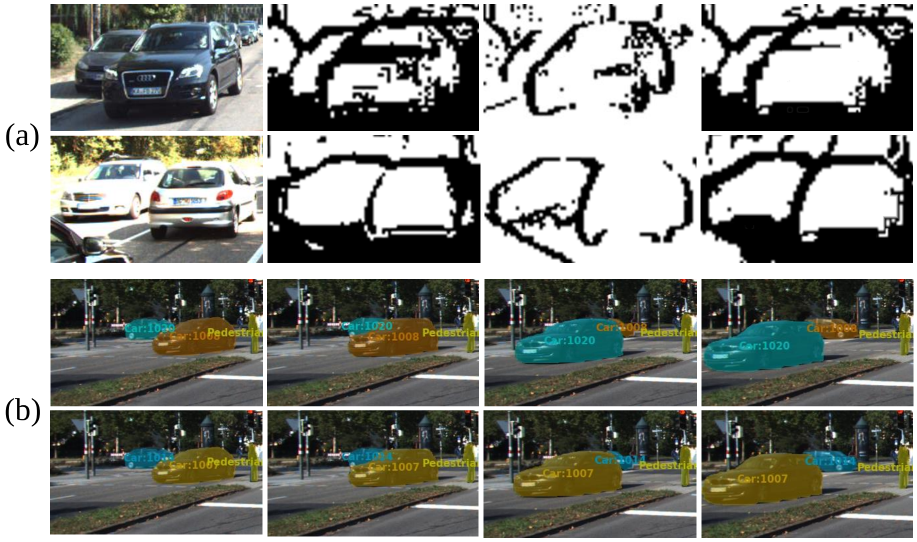


Fig. 7. **Ablation study results**: a). The pseudo-labels generated by depth signals (2^{nd} column), optical flow signals (3^{rd} column), and combination of both (4^{th} column); b). The same mask color indicates the same instance. The first row results are from CP [77], which often encounters the issue of ID switching. The second row results are from ours, which is robust to object appearance changes.

TABLE V
THE IMPACT OF **DIFFERENT MINI NETWORK DEPTH** IN KITTI MOTS. C AND P DENOTE *car* AND *pedestrian* RESPECTIVELY.

Depth	HOTA		MOTSA		MOTSP	
	C	P	C	P	C	P
1	57.4	46.0	82.3	67.4	83.3	73.8
2	58.8	46.6	82.8	67.9	84.2	74.7
3	**59.6**	**47.5**	**83.3**	**68.3**	**85.1**	**75.8**
4	59.2	47.1	83.1	68.2	84.9	75.3

TABLE VI
THE IMPACT OF **DIFFERENT MINI NETWORK DILATION RATE** IN KITTI MOTS. THE DEPTH OF THE MINI NETWORK IS FIXED TO 3. C AND P DENOTE *car* AND *pedestrian* RESPECTIVELY.

λ	HOTA		MOTSA		MOTSP	
	C	P	C	P	C	P
1	56.7	45.4	81.9	66.8	83.0	73.1
2	**59.6**	**47.5**	**83.3**	**68.3**	**85.1**	**75.8**
3	57.9	46.7	82.5	67.2	84.3	74.4

whose margins over the second best results (L_{bd}+$L_{bx}(L_{dice})$) by 1.4% (car) and 1.3% (pedestrian) on AP. The above results confirm our assumption for our puzzle loss design that the proposed box term and boundary term can work collaboratively to generate a high-quality instance mask.

TABLE VII
THE IMPACT OF **DIFFERENT MINI NETWORK FILTER FACTORS** IN KITTI MOTS. THE RESULTS ARE OBTAINED BY HOTA ON CAR AND PEDESTRIAN CATEGORY RESPECTIVELY.

ϕ^t / ϕ^s	0.2	0.3	0.4	0.5
0.2	38.2 / 32.8	43.6 / 37.4	54.1 / 43.2	48.7 / 36.5
0.3	42.6 / 37.7	49.3 / **47.9**	**59.6** / 47.5	52.6 / 40.1
0.4	39.0 / 35.4	46.8 / 41.1	50.6 / 40.4	41.9 / 35.3

TABLE VIII
THE RESULTS OF **USING DIFFERENT TRACKING STRATEGIES** FROM KITTI MOTS. CP, DP, AND SD ARE SHORT FOR CENTER POINT, DIAGONAL POINTS, AND SPATIO-TEMPORAL DISCREPANCY RESPECTIVELY.

Tracking by	HOTA		MOTSA		MOTSP	
	C	P	C	P	C	P
CP	58.9	47.1	83.1	68.0	83.9	74.5
DP	59.3	47.3	83.2	68.2	84.7	74.9
DP+SD	**59.6**	**47.5**	**83.3**	**68.3**	**85.1**	**75.8**

TABLE IX
THE COMPARISON RESULTS ON YT-VIS VALIDATION SET. * AND † INDICATE THE USE OF THE **GROUND TRUTH** AND **PSEUDO LABELS** RESPECTIVELY DURING TRAINING. ALL METHODS USE RESNET-101 WITH FPN.

Methods	mAP	AP@0.75	AR@10
YOLACT*	29.7	32.1	36.5
BlendMask*	32.0	34.1	39.7
HTC*	35.3	36.9	40.8
YOLACT†	28.9	31.2	35.8
BlendMask†	31.3	33.3	38.8
HTC†	34.5	36.3	40.0

3) Mini Network Architecture: We also evaluate the impact of using different configurations for the mini network. Specifically, we vary the mini network depth (number of layers) from the list of $\{1, 2, 3, 4\}$ with the fixed dilation rate λ of 2 and dilation convolution kernel size 3. We also vary the dilation rate λ of the mini network from $\{1, 2, 3\}$, and use the grad search to determine the filter factors ϕ^s, ϕ^t. The results are shown in Table V, Table VI and Table VII respectively. Those results show that a reasonable mini network configuration can account for better supervision, where the mini network includes 3 layers of dilated convolutions with a dilation rate of 2 and a kernel size of 3. To achieve better performances, the filter factors ϕ^s, ϕ^t are set to be 0.3 and 0.4 respectively.

4) Tracking Strategy: We finally evaluate the impact of using different elements for tracking (see Table VIII). For CP, we use the state-of-the-art CenterTrack [77]. For DP, we only use diagonal points in our tracking module. For DP+SD, it uses both diagonal points and spatio-temporal discrepancy. From the results we can see that DP provides immediate improvements in tracking over the baseline that uses CP. DP+SD further improves the tracking capacity compared to DP, demonstrating strong tracking robustness (see Fig. 7b). These results suggest that each element (*i.e.* DP and SD) individually contributes towards improving the tracking performance.

G. Extending Instance Segmentation to Videos

In this section, we evaluate the flexibility and generalization of the proposed STC-Seg framework. In particular, we leverage our STC-Seg framework (*i.e.* pseudo-label generation, puzzle solver, and tracking module) to extend image instance segmentation methods to the video task. We select three widely recognized instance segmentation methods, YOLACT [45], BlendMask [50] and HTC [44]), and integrate with our STC-Seg framework. The results of two set of experiments, *i.e.* training with ground truth labels and pseudo labels, on YT-VIS are shown in Table IX. Each of the selected methods is crafted with our tracking module and uses the same implementation as discussed in Section IV. It can be seen that methods trained using the proposed pseudo-labels achieve comparable results with the models trained on ground truth labels. This observation is consistent among all three selected methods, which demonstrates that our STC-Seg framework can flexibly extend image instance segmentation methods to operate on video tasks.

TABLE X

RESULTS OF USING GROUND TRUTHS OR NOT IN SPATIO-TEMPORAL SIGNALS GENERATION WHEN TRAINING OUR STC-SEG ON KITTI MOTS. "×" DENOTES SIGNAL IS OBTAINED FROM THE PREDICTED DEPTH OR FLOW, WHILE "✓" DENOTES SIGNAL IS OBTAINED FROM THEIR GROUND TRUTH. C AND P DENOTE *car* AND *pedestrian* RESPECTIVELY.

Signal GT		HOTA		MOTSA		MOTSP	
Depth	Flow	C	P	C	P	C	P
×	×	59.6	47.5	83.3	68.3	85.1	75.8
✓	×	63.1	48.3	89.9	69.0	86.6	75.9
×	✓	61.4	49.2	87.5	69.8	86.4	76.1
✓	✓	**64.2**	**51.1**	**92.7**	**70.5**	**86.9**	**76.2**

H. Results Using Ground Truth Depth and Flow

Since depth estimation and optical flow are critical factors to generate our pseudo-label, we also directly employ the ground truth depth and flow for the pseudo-label generation in training to investigate the performance gap between using the predicted spatio-temporal signals and ground truths. Table X demonstrates the results on KITTI MOTS. We can see that using depth and flow ground truths can further improve the performance. Thus, we argue that with strong depth and flow predictions, our method can achieve further performance gain.

V. CONCLUSION AND LIMITATION

Instance segmentation in videos is an important research problem, which has been applied in a wide range of vision applications. In this study, we propose a weakly supervised learning method for instance segmentation in videos with a spatio-temporal collaboration framework, titled STC-Seg. In particular, we introduce a weakly supervised training strategy which successfully combines unsupervised spatio-temporal collaboration and weakly supervised signals, helping networks to jointly achieve completeness and adequacy for instance segmentation in videos without pixel-wised labels. STC-Seg works in a plug-and-play manner and can be nested in any segmentation network method. Extensive experimental results indicate that STC-Seg is competitive with the concurrent methods and outperforms fully supervised MaskTrack R-CNN and TrackR-CNN. Albeit achieving strong performance, our method requires box labels to operate training which limits its applicability to new tasks without any prior knowledge. This challenge remains open for our future research endeavors. There are several ongoing investigations. For example, we are exploring unsupervised or weakly supervised object detection methods to obtain box labels. These predicted box labels can then be used to predict instance segmentation.

REFERENCES

[1] D. Liu, Y. Cui, L. Yan, C. Mousas, B. Yang, and Y. Chen, "Densernet: Weakly supervised visual localization using multi-scale feature aggregation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, 2021, pp. 6101–6109.

[2] Z. Cheng, J. Liang, H. Choi, G. Tao, Z. Cao, D. Liu, and X. Zhang, "Physical attack on monocular depth estimation with optimal adversarial patches," *ECCV*, 2022.

[3] J. Liang, Y. Wang, Y. Chen, B. Yang, and D. Liu, "A triangulation-based visual localization for field robots," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 6, pp. 1083–1086, 2022.

[4] Y. Cui, L. Yan, Z. Cao, and D. Liu, "Tf-blender: Temporal feature blender for video object detection," *ICCV*, 2021.

[5] X. Lu, W. Wang, J. Shen, Y.-W. Tai, D. J. Crandall, and S. C. Hoi, "Learning video object segmentation from unlabeled videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8960–8970.

[6] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 724–732.

[7] F. Porikli, F. Bashir, and H. Sun, "Compressed domain video object segmentation," *IEEE transactions on circuits and systems for video technology (TCSVT)*, vol. 20, no. 1, pp. 2–14, 2009.

[8] L. Zhao, Z. He, W. Cao, and D. Zhao, "Real-time moving object segmentation and classification from hevc compressed surveillance video," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 28, no. 6, pp. 1346–1357, 2016.

[9] D. Liu, Y. Cui, W. Tan, and Y. 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*, 2021, pp. 9816–9825.

[10] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7942–7951.

[11] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5188–5197.

[12] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 876–885.

[13] I. H. Laradji, D. Vazquez, and M. Schmidt, "Where are the masks: Instance segmentation with image-level supervision," *arXiv preprint arXiv:1907.01430*, 2019.

[14] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, "Weakly supervised instance segmentation using class peak response," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3791–3800.

[15] L. Hoyer, D. Dai, Y. Chen, A. Koring, S. Saha, and L. Van Gool, "Three ways to improve semantic segmentation with self-supervised depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 130–11 140.

[16] F. Lin, H. Xie, Y. Li, and Y. Zhang, "Query-memory re-aggregation for weakly-supervised video object segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2038–2046.

[17] Q. Liu, V. Ramanathan, D. Mahajan, A. Yuille, and Z. Yang, "Weakly supervised instance segmentation for videos with temporal mask consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 968–13 978.

[18] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2019, pp. 1328–1338.

[19] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7268–7277.

[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[21] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[22] J. Ahn, S. Cho, and S. Kwak, "Weakly supervised learning of instance segmentation with inter-pixel relations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2209–2218.

[23] H. Cholakkal, G. Sun, F. S. Khan, and L. Shao, "Object counting and instance segmentation with image-level supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 397–12 405.

[24] Y. Zhu, Y. Zhou, H. Xu, Q. Ye, D. Doermann, and J. Jiao, "Learning instance activation maps for weakly supervised instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3116–3125.

[25] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4981–4990.

[26] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, "Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5267–5276.

[27] W. Yang, H. Huang, Z. Zhang, X. Chen, K. Huang, and S. Zhang, "Towards rich feature discovery with class activation maps augmentation for person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1389–1398.

[28] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 275–12 284.

[29] Y. Cui, Z. Cao, Y. Xie, X. Jiang, F. Tao, Y. V. Chen, L. Li, and D. Liu, "Dg-labeler and dgl-mots dataset: Boost the autonomous driving perception," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 58–67.

[30] Y. Chen, G. Lin, S. Li, O. Bourahla, Y. Wu, F. Wang, J. Feng, M. Xu, and X. Li, "Banet: Bidirectional aggregation network with occlusion handling for panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3793–3802.

[31] J. Hur and S. Roth, "Joint optical flow and temporally consistent semantic segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 163–177.

[32] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1635–1643.

[33] C.-C. Hsu, K.-J. Hsu, C.-C. Tsai, Y.-Y. Lin, and Y.-Y. Chuang, "Weakly supervised instance segmentation using the bounding box tightness prior," *Advances in Neural Information Processing Systems*, vol. 32, pp. 6586–6597, 2019.

[34] V. Kulharia, S. Chandra, A. Agrawal, P. Torr, and A. Tyagi, "Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 290–308.

[35] M. Rajchl, M. C. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Damodaram, M. A. Rutherford, J. V. Hajnal, B. Kainz *et al.*, "Deepcut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE transactions on medical imaging*, vol. 36, no. 2, pp. 674–683, 2016.

[36] Z. Tian, C. Shen, X. Wang, and H. Chen, "Boxinst: High-performance instance segmentation with box annotations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5443–5452.

[37] A. Arun, C. Jawahar, and M. P. Kumar, "Weakly supervised instance segmentation by learning annotation consistent instances," in *European Conference on Computer Vision*. Springer, 2020, pp. 254–270.

[38] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1742–1750.

[39] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3136–3145.

[40] W. Ge, S. Guo, W. Huang, and M. R. Scott, "Label-penet: Sequential label propagation and enhancement networks for weakly supervised instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3345–3354.

[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[42] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[43] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6409–6418.

[44] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[45] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9157–9166.

[46] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European conference on computer vision*. Springer, 2016, pp. 75–91.

[47] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *European conference on computer vision*. Springer, 2016, pp. 534–549.

[48] S. Peng, W. Jiang, H. Pi, X. Li, H. Bao, and X. Zhou, "Deep snake for real-time instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8533–8542.

[49] J. Cao, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang, and L. Shao, "Sipmask: Spatial information preservation for fast image and video instance segmentation," in *ECCV*, 2020.

[50] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8573–8581.

[51] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[52] Y. Shen, R. Ji, Y. Wang, Y. Wu, and L. Cao, "Cyclic guidance for weakly supervised joint detection and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 697–707.

[53] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object

proposal generation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 128–140, 2016.

[54] J. Lee, J. Yi, C. Shin, and S. Yoon, "Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2643–2652.

[55] Y. Liu, Y.-H. Wu, P.-S. Wen, Y.-J. Shi, Y. Qiu, and M.-M. Cheng, "Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[56] A. Athar, S. Mahadevan, A. Osep, L. Leal-Taixé, and B. Leibe, "Stemseg: Spatio-temporal embeddings for instance segmentation in videos," in *European Conference on Computer Vision*. Springer, 2020, pp. 158–177.

[57] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, "Polarmask: Single shot instance segmentation with polar representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 193–12 202.

[58] W. Liu, G. Lin, T. Zhang, and Z. Liu, "Guided co-segmentation network for fast video object segmentation," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 31, no. 4, pp. 1607–1617, 2020.

[59] L. Liu and G. Fan, "Combined key-frame extraction and object-based video segmentation," *IEEE transactions on circuits and systems for video technology (TCSVT)*, vol. 15, no. 7, pp. 869–884, 2005.

[60] Y. Gui, Y. Tian, D.-J. Zeng, Z.-F. Xie, and Y.-Y. Cai, "Reliable and dynamic appearance modeling and label consistency enforcing for fast and coherent video object segmentation with the bilateral grid," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 30, no. 12, pp. 4781–4795, 2019.

[61] F. Lin, H. Xie, C. Liu, and Y. Zhang, "Bilateral temporal re-aggregation for weakly-supervised video object segmentation," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2021.

[62] L. Yan, S. Ma, Q. Wang, Y. Chen, X. Zhang, A. Savakis, and D. Liu, "Video captioning using global-local representation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[63] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang, "Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3300–3315, 2021.

[64] X. Shu, J. Yang, R. Yan, and Y. Song, "Expansion-squeeze-excitation fusion network for elderly activity recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[65] Y. Liqi, W. Qifan, C. Yiming, F. Fuli, Q. Xiaojun, Z. Xiangyu, and L. Dongfang, "Gl-rg: Global-local representation granularity for video captioning," in *IJCAI*, 2022.

[66] B. Li, X. Li, Z. Zhang, and F. Wu, "Spatio-temporal graph routing for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8561–8568.

[67] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.

[68] C. Godard, O. M. Aodha, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3827–3837, 2019.

[69] K. Luo, C. Wang, S. Liu, H. Fan, J. Wang, and J. Sun, "Upflow: Upsampling pyramid for unsupervised optical flow learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1045–1054.

[70] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, 2016.

[71] Y. Xu, L. Zhu, Y. Yang, and F. Wu, "Training robust object detectors from noisy category labels and imprecise bounding boxes," *IEEE Transactions on Image Processing (TIP)*, vol. 30, pp. 5782–5792, 2021.

[72] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," *Advances in neural information processing systems (NeurIPS)*, vol. 32, 2019.

[73] K.-H. Lee, X. He, L. Zhang, and L. Yang, "Cleannet: Transfer learning for scalable image classifier training with label noise," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018, pp. 5447–5456.

[74] H. Li, Z. Wu, C. Zhu, C. Xiong, R. Socher, and L. S. Davis, "Learning from noisy anchors for one-stage object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 588–10 597.

[75] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 548–578, 2021.

[76] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 784–11 793.

[77] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *ECCV*, 2020, pp. 474–490.

[78] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017*, 2017, pp. 3645–3649.

[79] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *European Conference on Computer Vision*. Springer, 2020, pp. 107–122.

[80] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021.

[81] S. Qiao, Y. Zhu, H. Adam, A. L. Yuille, and L. Chen, "Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021, pp. 3997–4008.

[82] A. Kim, A. Osep, and L. Leal-Taixé, "Eagermot: 3d multi-object tracking via sensor fusion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[83] J. Luiten, T. Fischer, and B. Leibe, "Track to reconstruct and reconstruct to track," *IEEE Robotics and Automation Letters*, 2020.

[84] Z. Xu, W. Zhang, X. Tan, W. Yang, H. Huang, S. Wen, E. Ding, and L. Huang, "Segment as points for efficient online multi-object tracking and segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[85] B. Cheng, O. Parkhi, and A. Kirillov, "Pointly-supervised instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2617–2626.

[86] I. Ruiz, L. Porzi, S. R. Bulo, P. Kontschieder, and J. Serrat, "Weakly supervised multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 125–133.

[87] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *ECCV*, 2020.

[88] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[89] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[90] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[91] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1851–1858.

[92] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V*, vol. 7576, 2012, pp. 746–760.

[93] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[94] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3061–3070.

[95] L. Liu, J. Zhang, R. He, Y. Liu, Y. Wang, Y. Tai, D. Luo, C. Wang, J. Li, and F. Huang, "Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6489–6498.

[96] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for

disparity, optical flow, and scene flow estimation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[97] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.