

TubeFormer-DeepLab: Video Mask Transformer

Dahun Kim^{1,3} Jun Xie³ Huiyu Wang² Siyuan Qiao³ Qihang Yu² Hong-Seok Kim³
 Hartwig Adam³ In So Kweon¹ Liang-Chieh Chen³
¹KAIST ²Johns Hopkins University ³Google Research

Abstract

We present *TubeFormer-DeepLab*, the first attempt to tackle multiple core video segmentation tasks in a unified manner. Different video segmentation tasks (e.g., video semantic/instance/panoptic segmentation) are usually considered as distinct problems. State-of-the-art models adopted in the separate communities have diverged, and radically different approaches dominate in each task. By contrast, we make a crucial observation that video segmentation tasks could be generally formulated as the problem of assigning different predicted labels to video tubes (where a tube is obtained by linking segmentation masks along the time axis) and the labels may encode different values depending on the target task. The observation motivates us to develop *TubeFormer-DeepLab*, a simple and effective video mask transformer model that is widely applicable to multiple video segmentation tasks. *TubeFormer-DeepLab* directly predicts video tubes with task-specific labels (either pure semantic categories, or both semantic categories and instance identities), which not only significantly simplifies video segmentation models, but also advances state-of-the-art results on multiple video segmentation benchmarks.

1. Introduction

We observe that video segmentation tasks could be formulated as *partitioning video frames into tubes with different predicted labels*, where a tube contains segmentation masks linked along the time axis. Based on the target task, the predicted labels may encode only semantic categories (e.g., Video Semantic Segmentation (VSS) [8, 64]), or both semantic categories and instance identities (e.g., Video Instance Segmentation (VIS) [74, 84] for only foreground ‘things’, or Video Panoptic Segmentation (VPS) [44, 79] for both foreground ‘things’ and background ‘stuff’) (Fig. 1).

However, the underlying similarity of several video segmentation tasks (i.e., assigning tubes with predicted labels) has been long overlooked, and thus models developed for video semantic, instance, and panoptic segmentation have fundamentally diverged. For example, some VSS methods [29, 94] warp features between video frames, while the

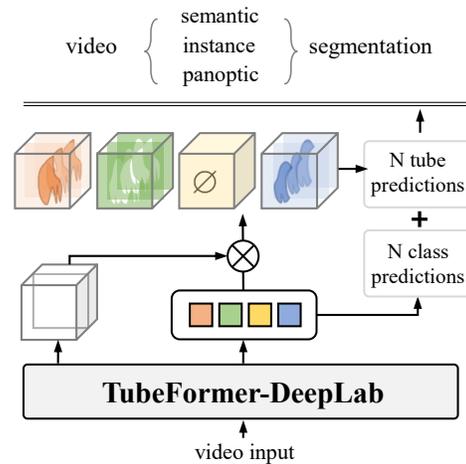


Figure 1. Video segmentation tasks can be formulated as partitioning video frames (e.g., a clip) into tubes (i.e., segmentation masks linked along time) with different labels. TubeFormer-DeepLab directly predicts class-labeled tubes, providing a simple and general solution to Video Semantic Segmentation (VSS), Video Instance Segmentation (VIS), and Video Panoptic Segmentation (VPS).

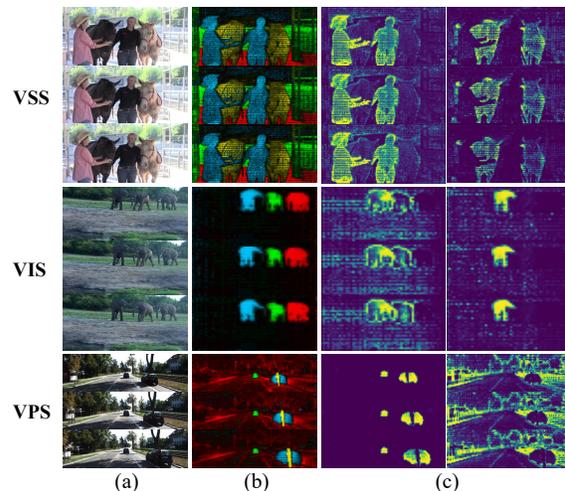


Figure 2. Our proposed hierarchical dual-path transformer performs attention on three consecutive input frames (a) for VSS, VIS, and VPS tasks. While the global memory learns the spatio-temporally clustered attention for individual tube regions (b), our latent memory learns task-specific attention (c).

modern VIS model [5] predicts hundreds of frame-level instance masks [34] and then propagates them to other neighboring frames. To make matters more complicated, state-of-the-art VPS methods [68, 80] adopt separate prediction branches, specific to semantic segmentation, instance segmentation, and object tracking, respectively.

In this work, instead of exacerbating the bifurcation between video segmentation models, we take a step back and rethink the following question: *Can we exploit the similar nature between video segmentation tasks, and develop a single model that is both effective and generally applicable?* To answer this, we propose **TubeFormer-DeepLab** that builds upon mask transformers [75] for video segmentation by directly predicting class-labeled tubes, where the labels encode different values depending on the target task.

Specifically, similar to other Transformer architectures [10, 73], TubeFormer-DeepLab extends the mask transformer [75] to generate a set of pairs, each containing a class prediction and a tube embedding vector. The tube embedding vector, multiplied by the video pixel embedding features obtained by a convolutional network [48], yields the tube prediction. As a result, TubeFormer-DeepLab presents the first attempt to tackle multiple core video segmentation tasks in a general framework without the need to adapt the system for any task-specific design.

Naïvely applying the image-level mask transformer [75] to the video domain does not yield a satisfactory result, mainly due to the difficulty of learning attentions for *video-clip* (i.e., multi-frames) features with large spatial resolutions. To alleviate the issue, we introduce the **latent** dual-path transformer block that is in charge of passing messages between *video-frame* (i.e., single-frame) features and a **latent** memory, followed by the **global** dual-path transformer block that learns the attentions between *video-clip* features and a **global** memory. This hierarchical dual-path transformer framework facilitates the attention learning and significantly improves the video segmentation results. Interestingly, as shown in Fig. 2, our latent memory learns task-specific attention, while the global memory learns the spatio-temporally clustered attention for individual tube regions. Additionally, we split the global memory into two sets, thing-specific and stuff-specific global memory, with the motivation to exploit the different nature of ‘thing’ (countable instances) and ‘stuff’ (amorphous regions).

During inference, practically we could only fit a video clip (i.e., a short video sequence) for video segmentation. The whole video sequence segmentation result is thus obtained by applying the video stitching [69] to merge clip segmentation results. To enforce the consistency between video clips, we additionally propose a Temporal Consistency loss that encourages the model to learn consistent predictions in the overlapping frames between clips.

Finally, we propose a simple and effective data augmen-

tation policy by extending the image-level thing-specific copy-paste [27, 32]. Our method, named clip-paste (clip-level copy-paste), randomly pastes either ‘thing’ or ‘stuff’ (or both) regions from a video clip to the target video clip.

To demonstrate the effectiveness of our proposed TubeFormer-DeepLab, we conduct experiments on multiple core video segmentation datasets, including KITTI-STEP (VPS) [79], VSPW (VSS) [64], YouTube-VIS (VIS) [84], SemKITTI-DVPS (depth-aware VPS) [69], and recent VIPSeg [63] (VPS). Our *single* model not only significantly simplifies video segmentation systems (e.g., the proposed model is end-to-end trained and does not require any task-specific design), but also advances state-of-the-art performance on several benchmarks. In particular, TubeFormer-DeepLab outperforms *published works* Motion-DeepLab [79] by **+13.1** STQ on KITTI-STEP *test* set, TCB [64] by **+21** mIoU on VSPW *test* set, IFC [40] by **+2.9** track-mAP on YouTube-VIS-2019 *val* set, ViP-DeepLab [69] by **+3.6** DSTQ on SemKITTI-DVPS *test* set, Clip-PanoFCN [63] by **+13.6** STQ and **+3.9** VPQ on VIPSeg *test* set. Our experimental results validate TubeFormer-DeepLab’s general efficacy for video segmentation tasks.

2. Related Works

Video Semantic Segmentation (VSS). Extending image semantic segmentation [13, 17, 22, 26, 37, 39, 60, 81, 88, 92, 96] to the video domain requires predicting all pixels in a video with different semantic classes [8, 64]. Prior methods [29, 42, 52, 65, 94, 95] exploit the temporal information via a warping module [23, 38, 41]. Recently, Mao *et al.* [64] introduced a large-scale VSS benchmark, called VSPW (Video Scene Parsing in the Wild), along with a solid baseline that effectively aggregates video context information by extending [88] and [92] to the temporal dimension.

Video Instance Segmentation (VIS). Combining multi-object tracking [4, 7, 24, 30, 67] and instance segmentation [11, 33, 34, 58, 68, 71], video instance segmentation [74, 84] aims to track instance masks across video frames. Most state-of-the-art VIS methods [5, 9, 28, 40, 54, 57, 77, 84] are detection-based approaches, allowing overlapping mask predictions (e.g., based on Mask R-CNN [34], FCOS [72], or DETR [10, 93]). Our work is similar to the concurrent work IFC [40], which uses memory features for video instance segmentation. However, our work does not exploit memory features for inter-frame communication, and thus does not require extra modules to perform such a task. Instead, the latent memory features are deployed in the proposed latent dual-path transformer block to facilitate *per-frame* segmentation. Finally, LatentGNN [90] also explored the latent features in the graphical neural networks [70].

Video Panoptic Segmentation (VPS). Recently, panoptic segmentation [19, 45, 46, 51, 53, 75, 76, 83, 86] has

also been extended to the video domain. Video Panoptic Segmentation [44] attempts to unify video semantic and instance segmentation, requiring temporally consistent panoptic segmentation results. Different from VIS, VPS disallows overlapping instance masks and requires labeling each pixel, including both ‘thing’ and ‘stuff’ pixels. Current state-of-the-art approaches [69, 79, 80] adopted complicated pipelines due to the intricate nature of VPS. Specifically, VPSNet [44] contains multiple task-specific heads, including Mask R-CNN [34], deformable convolutions [23], and MaskTrack [84] for instance segmentation, semantic segmentation, and tracking, respectively, while ViP-DeepLab [69] extends Panoptic-DeepLab [20] (which employs dual-ASPP [15] and dual-decoder structures specific to semantic and instance segmentation, respectively) by adding another next-frame instance segmentation branch. On the other hand, our approach significantly simplifies the current pipeline by employing mask transformers [75] to directly predict clip-level mask segmentation results. Finally, our proposed model could also be easily extended to the recent task of Depth-aware Video Panoptic Segmentation (DVPS) [69], which further requires per-pixel depth estimation on top of VPS results. We note that current with our work, Video K-Net [50], extending K-Net [91], also develops a unified framework for video panoptic segmentation.

3. Method

In this section, we introduce the formulation of several video segmentation tasks, followed by a general formulation that inspires our TubeFormer-DeepLab. We then present its model design, training and inference strategies.

3.1. Video Segmentation Formulation

Let us denote with $v \in \mathbb{R}^{T \times H \times W \times 3}$ an input video clip containing T video frames of spatial size $H \times W$ (T could be equal to the video sequence length if memory allows). The video clip is annotated with a set of class-labeled tubes (a tube is defined as segmentation masks linked along the time axis): $\{y_i\}_{i=1}^K = \{(m_i, c_i)\}_{i=1}^K$, where the K ground truth tubes $m_i \in \{0, 1\}^{T \times H \times W}$ do not overlap with each other, and c_i denotes the ground truth class label of tube m_i . Below, we briefly introduce several tasks.

Video Semantic Segmentation (VSS) is typically formulated as per-video pixel classification, where the pixel features for classification are enriched by warping [94] or aggregating [64] features from neighboring frames. Formally, the model predicts the probability distribution over a predefined set of categories $\mathbb{C} = \{1, \dots, D\}$ for every video pixel: $\{\hat{p}_i | \hat{p}_i \in \Delta^D\}_{i=1}^{T \times H \times W}$, where Δ^D is the D -dimensional probability simplex. The final segmentation output \hat{y} is then obtained by taking its argmax (*i.e.*, $\hat{y}_i = \arg \max_c \hat{p}_i(c), \forall i \in \{1, 2, \dots, T \times H \times W\}$).

Video Instance Segmentation (VIS) requires to segment and temporally link object instances in the video. For each detected foreground ‘thing’ i in the video, the model predicts a video tube (*i.e.*, video-level instance mask track) $\hat{m}_i \in [0, 1]^{T \times H \times W}$ with a probability distribution \hat{p}_i over \mathbb{C} defined for *only* thing classes. Depending on the target dataset or evaluation metric, the model may generate *overlapping* video tubes (*e.g.*, Youtbue-VIS [84] adopts track-mAP, allowing overlapping predicted tubes, while KITTI-MOTS [74] adopts HOTA [62], disallowing so).

Video Panoptic Segmentation (VPS) requires temporally consistent semantic and instance segmentation results for both ‘thing’ and ‘stuff’ classes. Specifically, the model predicts a set of *non-overlapping* video tubes $\{\hat{y}_i\}_{i=1}^N = \{(\hat{m}_i, \hat{p}_i(c))\}_{i=1}^N$, where $\hat{m}_i \in [0, 1]^{T \times H \times W}$ denotes the predicted tube, and $\hat{p}_i(c)$ denotes the probability of assigning class c to tube \hat{m}_i belonging to a predefined category set \mathbb{C} that contains both ‘thing’ and ‘stuff’ classes.

Depth-aware Video Panoptic Segmentation (DVPS) builds on top of VPS by additionally requiring a model to estimate the depth value of each pixel. Similar to VPS output, the prediction has the following format: $\{\hat{y}_i\}_{i=1}^N = \{(\hat{m}_i, \hat{p}_i(c), \hat{d}_i)\}_{i=1}^N$, where $\hat{d}_i \in [0, d_{max}]^{T \times H \times W}$ denotes the estimated depth value and d_{max} is the maximum depth value specified in the target dataset. Accordingly, the dataset contains ground truth depth.

General task formulation. Despite the superficial differences between tasks, we discover the underlying similarity that video segmentation tasks could be generally formulated as the problem of assigning different predicted labels to video tubes and the labels may encode different values depending on the target task. For example, if only semantic categories are predicted, it becomes video semantic segmentation. Similarly, if both semantic categories and instance identities are required (*i.e.*, one predicted tube for each category-identity pair), it then becomes either video instance segmentation (if only foreground ‘thing’ classes are considered) or video panoptic segmentation. This motivates us to develop a general video segmentation model that directly predicts class-labeled tubes $\{\hat{y}_i\}_{i=1}^N = \{(\hat{m}_i, \hat{p}_i(c))\}_{i=1}^N$ (and optionally depth, if required).

3.2. TubeFormer-DeepLab Architecture

We first introduce TubeFormer-DeepLab-Simple, our video-level baseline, which will be improved by our proposed latent dual-path transformer, resulting in the final TubeFormer-DeepLab.

TubeFormer-DeepLab-Simple. We adopt the per-clip pipeline which takes a video clip and outputs clip-level results. Inspired by [75], our TubeFormer-DeepLab-Simple integrates a CNN backbone and a global memory feature in a dual-path architecture, *i.e.*, global dual-path transformer.

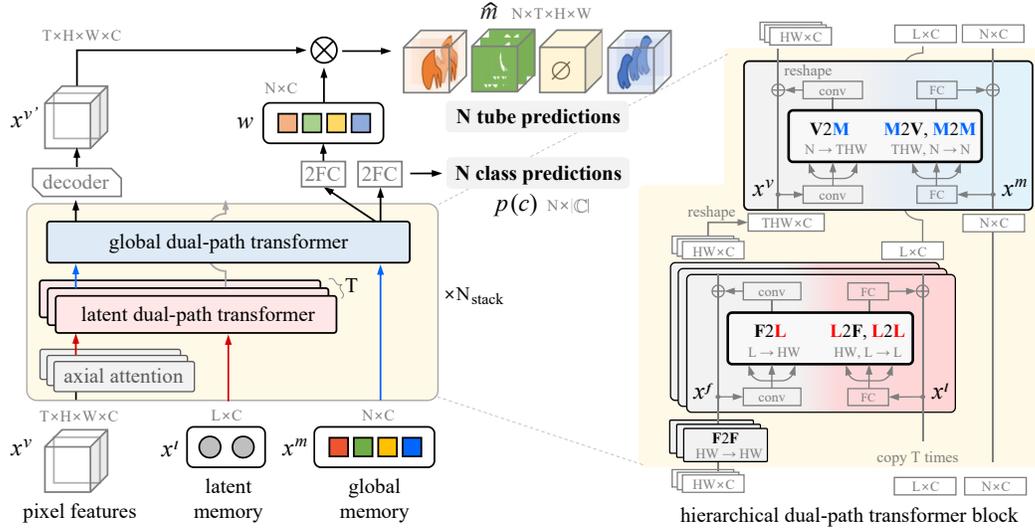


Figure 3. **TubeFormer-DeepLab architecture overview.** TubeFormer-DeepLab extends the mask transformer [75] to generate a set of pairs, each containing a class prediction $p(c)$ and a tube embedding vector w . The tube embedding vector, multiplied by the video pixel embedding features $x^{v'}$ obtained by a convolutional network, yields the tube prediction \hat{m} . We introduce a hierarchical structure with the latent dual-path transformer block that is in charge of passing messages between frame-level features x^f and a latent memory x^l , followed by the global dual-path transformer block that learns the attentions between video-clip features x^v and a global memory x^m .

Given an input video clip v , the CNN backbone processes the input frames independently, and generates pixel features $x^v \in \mathbb{R}^{T \times H \times W \times C}$, where C is channels. The pixel self-attention is performed at the frame level (frame-to-frame, F2F) via an axial-attention block [76].

Afterwards, the global dual-path transformer operates in a *per-clip* manner, taking the flattened video pixel features $x^v \in \mathbb{R}^{T \times H \times W \times C}$ and a 1D global memory $x^m \in \mathbb{R}^{N \times C}$ of length N (*i.e.* the size of the prediction set). Passing through the global dual-path transformer, we expect three attentions: (1) memory-to-video (M2V) attention (in which the video features encode per-clip information to the memory feature), (2) memory-to-memory (M2M) self-attention, and (3) video-to-memory (V2M) attention (in which the video pixel features refine themselves by receiving tube-level information gathered in the global memory). The global dual-path transformer blocks can be stacked multiple times at any layers of the network.

On top of the global memory, there are two output heads: a segmentation head and a class head, each composed of two Fully-Connected (FC) layers. The global memory of size N is independently passed to the two heads, resulting in N unique tube embeddings $w \in \mathbb{R}^{N \times C}$ and N corresponding class predictions $p(c) \in \mathbb{R}^{N \times |\mathcal{C}|}$. Note that the possible classes $\mathcal{C} \ni c$ include “none” category \emptyset in case the embedding does not correspond to any region in a clip. Our video tube prediction \hat{m} is computed in one shot as a dot-product between the decoded video pixel features $x^{v'}$ and the tube embeddings w :

$$\hat{m} = \text{softmax}_N(x^{v'} \cdot w) \in \mathbb{R}^{N \times T \times H \times W}. \quad (1)$$

The final video-clip segmentation $\{\hat{y}_i\}_{i=1}^N = \{(\hat{m}_i, \hat{p}_i(c))\}_{i=1}^N$ can be obtained by combining N binary video tubes with their corresponding class predictions.

TubeFormer-DeepLab with Latent Dual-Path Transformer. Modeling long-range interactions in *video-clip* (*i.e.*, multi-frames) features is especially difficult, when dealing with high-resolution inputs or a large number of input frames. To both alleviate the issue and facilitate the attention learning, we propose a hierarchical structure, which allows two levels of attention mechanisms: frame-level, followed by video-level. Note the video-level attention is performed by the aforementioned global dual-path transformer.

Prior to the global dual-path transformer, we introduce a new **latent** dual-path transformer block in charge of passing messages between *frame-level* features and a **latent** memory. It processes individual video frames in parallel (batch-wise). Our latent memory is inspired by the graphical models with latent representations [40, 47, 90], allowing a low-rank representation for the graph affinity of high complexity. Concurrent with IFC [40], we discovered that latent features facilitate attention learning. However, we deployed them in a different framework (*e.g.*, dual-path transformer and no cross-frame communication).

Specifically, the initial latent memory $x^l \in \mathbb{R}^{L \times C}$ is copied per frame and paired with each frame’s features $x^f \in \mathbb{R}^{HW \times C}$ (flattened) to construct the input. Passing through the latent dual-path transformer, the latent memory first collects messages from frame features via latent-to-frame (L2F) attention, and perform latent-to-latent (L2L) self-attention among themselves. Afterwards, the per-frame

knowledge from the latent memory is propagated back to the frame features via frame-to-latent (F2L) attention. Note the latent memory features are trainable parameters like the global memory features. However, they are only deployed in the latent space (*i.e.*, intermediate layers) and will not be used in the final output layers.

As shown in Fig. 3, our hierarchical dual-path transformer blocks consist of a series of one axial-attention block, the latent dual-path transformer, and the global dual-path transformer. The stacking of multiple blocks will alternate the latent and the global communications, allowing the pixel features to refine themselves by attending to both frame-level and video-level memory, and vice versa. This in turn enriches the features of all three paths: pixel-, latent-memory and global-memory paths, and enables learning more comprehensive representations of the given video clip.

Global memory with split thing and stuff. To further improve the segmentation quality, we propose to split the global memory into two sets: thing-specific and stuff-specific global memory. Originally, the global memory in [75] deals with thing masks and stuff masks in a unified manner. However, the design ignores the natural difference between them — There could be multiple instances of the same thing class in an image, but at most one mask is allowed for each stuff class. We thus allocate the last $|\mathcal{C}^{\text{stuff}}|$ out of N elements in the global memory specifically for predicting stuff classes. The ordering is enforced by assigning the stuff-specific global memory to the ground truth stuff classes, instead of including them in the bipartite matching.

3.3. Training Strategy

VPQ-style loss. To train TubeFormer-DeepLab for various video segmentation tasks in a unified manner, we adopt a VPQ-style loss that directly optimizes the set of class-labeled tubes. Similar to the image-level PQ-style loss [75], we draw inspiration from *video panoptic quality* (VPQ) [44] and approximately optimize VPQ within a video clip.

To start with, a VPQ-style similarity metric between a class-labeled ground truth tube $y_i = (m_i, c_i)$ and a predicted tube $\hat{y}_j = (\hat{m}_j, \hat{p}_j(c))$ can be defined as: $\text{sim}(y_i, \hat{y}_j) = \hat{p}_j(c_i) \times \text{Dice}(m_i, \hat{m}_j)$, where $\hat{p}_j(c_i) \in [0, 1]$ denotes the probability of predicting the correct tube class c_i and $\text{Dice}(m_i, \hat{m}_j) \in [0, 1]$ measures the Dice coefficient between a predicted tube \hat{m}_j and a ground truth tube m_i .

We match the predicted tubes to the ground truth tubes, and optimize the predictions by maximizing the total VPQ-style similarity. The implementation details follow the PQ-style loss in [75]. In addition, we generalize the auxiliary losses used in [75] to video clips, resulting in a tube-ID cross entropy loss, a video semantic segmentation loss, and a video instance discrimination loss.

Shared semantic and panoptic prediction. Originally,

the auxiliary semantic segmentation loss in [75] is applied to the backbone feature with a *separate* semantic decoder. Instead, we propose to apply the loss directly to the decoded video pixel features $x^{v'}$ (*cf.* Eq. (1)) with a linear layer, which learns better features for segmentation.

Temporal consistency loss. The VPQ-style loss benefits the learning of spatial-temporal consistency within an input clip. To further achieve the *clip-to-clip* consistency over a longer video, we propose to use a temporal consistency loss applied between clips. Specifically, we minimize the distance between the N tube logits predicted from the overlapping frames of two clips. We use $L1$ loss for the consistency metric. The loss is back-propagated through the dot-product of the pixel features and N global memory features, affecting both pixel and global memory paths. TubeFormer-DeepLab thereby achieves implicit multi-clip consistency, which makes our training objective symmetrical to the whole-video inference pipeline (Sec. 3.4).

Clip-level copy-paste. Additionally, we propose a simple and effective data augmentation policy by extending the image-level thing-specific copy-paste [27, 32]. Our augmentation method, named clip-paste (clip-level copy-paste), randomly pastes either ‘thing’ or ‘stuff’ (or both) region tubes from a video clip to the target video clip. We use clip-paste with a probability of 0.5.

Depth prediction branch. To grant TubeFormer-DeepLab the ability to perform monocular depth estimation, we add a small depth prediction module (*i.e.*, ASPP [14] and DeepLabv3+ lightweight decoder [17]) on top of the CNN *backbone* features x^v . Note that we found the performance slightly degrades if we add the depth prediction to the decoded video pixel features $x^{v'}$, indicating that it is not beneficial to share depth estimation with segmentation prediction in our case. We apply Sigmoid to constrain the depth prediction to the range (0, 1), and then multiply it by the maximum depth. Following [69], we use the combination of scale invariant logarithmic error [25] and relative squared error [31] as the training loss. The depth loss weight is set to 100 when jointly trained with the other losses.

3.4. Inference Strategy

Clip-level inference. The clip-level segmentation is inferred by simply performing argmax twice. Specifically, a class label is predicted for each tube: $\hat{c}_i = \text{arg max}_c \hat{p}_i(c)$. And then, a tube-ID $\hat{z}_{t,h,w}$ is assigned **per-pixel**: $\hat{z}_{t,h,w} = \text{arg max}_i \hat{m}_{i,t,h,w}$. In practice, our inference sets tube-IDs with class confidence below 0.7 to void.

For video instance segmentation, we also explore **per-mask** assignment scheme [21, 87], which treats the prediction of each object query as one object mask proposal.

Video-level inference. At the clip level, TubeFormer-DeepLab outputs temporally consistent results for T video

frames. To obtain the video-level prediction, we perform clip-level inference for every T consecutive frames with $T - 1$ overlapping frames (*i.e.*, we move along the temporal axis by only one frame at each inference step). The clip-level results are then stitched together by matching tubes in the overlapping frames based on their IoUs, similar to [69].

4. Experimental Results

Our proposed TubeFormer-DeepLab is a general video segmentation model. To demonstrate its effectiveness, we conduct experiments on KITTI-STEP [79], VIPSeg [63], VSPW [64], YouTube-VIS [84], SemKITTI-DVPS [69] for Video Panoptic Segmentation (VPS), Video Semantic Segmentation (VSS), Video Instance Segmentation (VIS), and Depth-aware Video Panoptic Segmentation (DVPS), respectively.

4.1. Datasets

KITTI-STEP [79] is a new video panoptic segmentation dataset that additionally annotates semantic segmentation for KITTI-MOTS [74]. It contains 19 semantic classes (similar to Cityscapes [22]), among which two classes (‘pedestrians’ and ‘cars’) come with tracking IDs. For evaluation, KITTI-STEP adopts STQ [79] (segmentation and tracking quality), which is the geometric mean of SQ (segmentation quality) and AQ (association quality).

VIPSeg [63] is also a new video panoptic segmentation dataset for diverse in-the-wild scenarios. It contains 124 semantic classes (58 ‘thing’ and 66 ‘stuff’ classes) with 3536 videos, where each video spans 3 to 10 seconds.

VSPW [64] is a recent large-scale video semantic segmentation dataset, containing 124 semantic classes. VSPW adopts mIoU as the evaluation metric.

YouTube-VIS [84] contains two versions for video instance segmentation; The YouTube-VIS-2019 contains 40 semantic classes and the YouTube-VIS-2021 is an improved version with higher number of instances and videos. Youtube-VIS adopts track mAP for evaluation.

SemKITTI-DVPS [69] is a new dataset for depth-aware video panoptic segmentation, which is obtained by projecting the 3D point cloud panoptic annotations of SemanticKITTI [3] to 2D image planes. It contains 19 classes, among which 8 are annotated with tracking IDs. For evaluation, SemKITTI-DVPS uses DSTQ (depth-aware STQ), which considers depth inlier metric [25] in addition to STQ.

4.2. Implementation Details

TubeFormer-DeepLab builds upon MaX-DeepLab [75] with the official codebase [78]. The hyper-parameters mostly follow the settings of [75]. Unless specified, we use their small model MaX-DeepLab-S, which augments ResNet-50 [35] with axial-attention blocks [76] in the last

method	rank	STQ	SQ	AQ
Motion-DeepLab [79]	7	52.19	59.81	45.55
ICCV 2021 challenge entries				
HybridTracker	6	54.99	55.54	55.54
slain	5	57.87	60.71	55.16
EffPs-MM	4	62.93	64.41	61.49
REPEAT [61]	2	67.13	68.49	65.81
UW_IPL/ETRI_AIRL [89]	1	67.55	64.04	71.26
TF-DL-B3	3	65.25	70.27	60.59

Table 1. [VPS] KITTI-STEP *test* set results. Ranking includes unpublished methods. The challenge winning entries [61, 89] adopt separate and ensemble methods for tracking and segmentation.

two stages (*i.e.*, stage-4 and stage-5). We also experiment with scaling up the backbone [16] by stacking the axial-attention blocks in stage-4 by n times, and refer them as TubeFormer-DeepLab-B n in the experiments. For VPS, we pretrain the models on Cityscapes [22] and COCO [56], while for other experiments, we only pretrain on COCO. The pretraining procedure is similar to prior works [5, 36, 79]. Using the pretrained weights, TubeFormer-DeepLab is trained on the target datasets using a batch size of 16, with $T = 2$ for all datasets except $T = 5$ for YouTube-VIS dataset. We use the global memory size $N = 128$ (*i.e.*, output size), latent memory size $L = 16$, and $C = 128$ channels. We use ‘TF-DL’ to denote TubeFormer-DeepLab in the results.

4.3. Main Results

[VPS] We evaluate TubeFormer-DeepLab on the challenging video panoptic segmentation dataset, KITTI-STEP [79] in Tab. 1. Our model achieves state-of-the-art performance with 65.25 STQ (70.27 SQ and 60.59 AQ). Among *single unified* approaches, our model ranks first, significantly outperforming the published baseline Motion-DeepLab [79] by **+13.1** STQ. Our model performs comparably with the challenge winning methods [61, 89] without exploiting extra 3D object formulation, depth information, or pseudo labels, and even without the employment of separate and ensemble methods for tracking and segmentation. Nevertheless, our model delivers the best segmentation quality (70.27 SQ), showcasing our TubeFormer-DeepLab’s segmentation ability.

We further evaluate TubeFormer-DeepLab on the recent video panoptic segmentation dataset, VIPSeg [63] in Tab. 2. Our method outperforms Clip-PanoFCN [63] (which built on top of Panopitc FCN [53]) by **+13.6** STQ and **+3.9** VPQ on the test set.

[VSS] We assess TubeFormer-DeepLab on the video semantic segmentation dataset, VSPW [64]. We show the single-model single-scale results on *val* set in Tab. 3. In the table, TubeFormer-DeepLab outperforms all competing methods, which are based on state-of-the-art backbones (BEiT [2], Swin-L [59]) and decoders (OCRNet [88], Uper-

method	STQ	VPQ
<i>val set</i>		
Clip-PanoFCN [63]	31.5	22.9
TF-DL-B1	39.8	29.2
TF-DL-B3	41.5	31.2
<i>test set</i>		
Clip-PanoFCN [63]	25.0	22.9
TF-DL-B3	38.6	26.8

Table 2. [VPS] VIPSeg *val* and *test* set results, using the latest test server at <https://codalab.lisn.upsaclay.fr/competitions/9743>

method	mIoU	VC8	VC16
TCB [64]	37.82	87.86	83.99
ICCV 2021 challenge entries			
BetterThing [18]	57.89	-	-
CharlesBLWX [43]	61.44	-	-
jjRain [36]	59.30	90.07	86.87
TF-DL-B4	63.16	92.08	87.95

Table 3. [VSS] VSPW *val* set results. Comparison includes published and unpublished methods.

method	rank	ens.	m.s.	pseudo	mIoU	VC8	VC16
<i>old codalab</i>							
TCB [64]	13				35.62	86.21	81.90
ICCV 2021 challenge entries							
BetterThing [18]	3	✓	✓		57.35	93.28	90.56
CharlesBLWX [43]	2	✓	✓		57.44	91.29	87.70
jjRain [36]	1	✓	✓	✓	58.85	94.77	92.59
TF-DL-B4	4				56.64	90.16	86.38
<i>new codalab</i>							
TCB [64]					32.58	79.46	73.23
TF-DL-B4					52.99	90.16	86.38

Table 4. [VSS] VSPW *test* set results. Ranking includes published and unpublished methods. Some methods use model ensembles, multi-scale inference, or teacher-student pseudo labeling strategy to boost performance on test set. In the bottom rows, we also include the new test set results, using the latest test server at <https://codalab.lisn.upsaclay.fr/competitions/7869>

Net [81]). Tab. 4 shows the *test* set results. Our single-model TubeFormer-DeepLab achieves competitive results (rank 4 out of 17) with the ICCV 2021 challenge winners, while not employing model ensembles, multi-scale inference, and pseudo labels. Finally, we attain a better +21 mIoU than the published work TCB [64] on the *test set*. As shown in the bottom of Tab. 4, we also include the new test set results using the latest test server.

[VIS] We show that TubeFormer-DeepLab is sufficiently general to solve instance-level video segmentation in a unified manner. The same model, loss, and training procedure is seamlessly applied by treating the background region as a single ‘stuff’ class. At testing, we explore both per-pixel and per-mask argmax for tube ID assignment (Sec. 3.4).

method	T	AP	AP ₅₀	AP ₇₅	AR ₁	AR ₁₀
MaskTrack [84]	2	31.8	53.0	33.6	33.2	37.6
SipMask [9]	2	33.7	54.1	35.8	35.4	40.1
STEm-Seg [1]	8	34.6	55.8	37.9	34.4	41.6
CrossVIS [85]	2	36.6	57.3	39.7	36.0	42.0
MaskProp [5]	13	46.6	-	51.2	44.0	52.6
Seq Mask R-CNN [54]	36	47.6	71.6	51.8	46.3	56.0
VisTR [77]	36	40.1	64.0	45.0	38.3	44.9
IFC [40]	36	44.6	69.2	49.5	44.0	52.1
TF-DL-B4 (per-pixel)	5	45.4	66.6	48.8	48.3	56.9
TF-DL-B4 (per-mask)	5	47.5	68.7	52.1	50.2	59.0

Table 5. [VIS] YouTube-VIS-2019 *val* set results.

method	T	AP	AP ₅₀	AP ₇₅	AR ₁	AR ₁₀
MaskTrack [84]	2	28.6	48.9	29.6	-	-
SipMask [9]	2 [†]	31.7	52.5	34.0	-	-
CrossVIS [85]	2 [†]	34.2	54.4	37.9	-	-
IFC [40]	36 [†]	36.8	57.9	39.3	-	-
TF-DL-B4 (per-mask)	5	41.2	60.4	44.7	40.4	54.0

Table 6. [VIS] YouTube-VIS-2021 *val* set results. [†]: T inferred from their Youtube-VIS-2019 settings.

method	rank	DSTQ
ViP-DeepLab [69]	3	63.36
ICCV 2021 challenge entries		
rl.lab	5	54.77
ywang26	4	55.99
HarborY [49]	2	63.63
TF-DL-B4	1	67.00

Table 7. [DVPS] SemKITTI-DVPS *test* set results. Ranking includes published and unpublished methods.

Tab. 5 and 6 show the comparison with the state-of-the-art methods on YouTube-VIS 2019 and 2021 datasets [84]. Note that TubeFormer-DeepLab predicts a single unique mask per object, while other methods often generate multiple overlapping masks, which are favored by the AP metric. Among end-to-end methods, our TubeFormer-DeepLab-B4 outperforms VisTR [77] by +7.4, and IFC [40] by +2.9 AP. Our model with $T = 5$ sets the highest scores among methods that employ a small value of T . Also, our gains in AR₁ are significant, indicating the benefit of TubeFormer-DeepLab in the non-overlapping segmentation scenario.

Our model performs comparably to Seq Mask R-CNN [54]. We point out that TubeFormer-DeepLab is an end-to-end near-online method, while Seq Mask R-CNN relies on STM [66]-like structure to propagate mask proposals through the whole sequence, and thus is offline ($T=36$).

[DVPS] We evaluate TubeFormer-DeepLab on the SemKITTI-DVPS dataset [69] for depth-aware video panoptic segmentation. Tab. 7 shows the *test* set results. Adding a depth prediction branch to the same exact TubeFormer-DeepLab used for KITTI-STEP outperforms ViP-DeepLab [69] by +3.4 DSTQ and achieves the new state-of-the-art of 67.0 DSTQ.

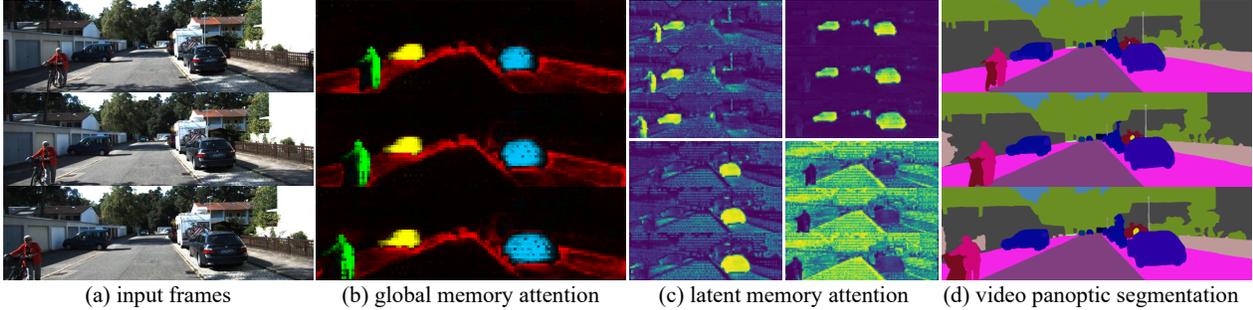


Figure 4. **Visualization** on KITTI-STEP sequence. From left to right: input frames ($T = 3$), global memory attention, latent memory attention, and video panoptic segmentation results. The global memory attention is selected for predicted tube regions of interest: a **pedestrian** and two cars (**left, right**) on the **sidewalk**, and the latent memory attention is selected for 4 (out of $L = 16$) latent memory.

4.4. Ablation Studies

We provide ablation studies on the KITTI-STEP *val* set [79]. To compensate for the training noise, we report the mean of three runs for every ablation study.

Hierarchical dual-path transformer. In Tab. 8a, we verify that the gains demonstrated by TubeFormer-DeepLab come from the proposed hierarchical dual-path transformer. Note that our baseline method (TubeFormer-DeepLab-Simple) already uses the axial attentions and the global memory. Introducing the new latent memory and its communication with the video-frame features (F-L attention: L2F, L2L, and F2L) brings a large improvement of +1.7 STQ. We also ablate adding attentions between the global memory and the latent memory (M2L and L2M), which show no improvements. This suggests the frame-latent (F-L) attention is sufficient to build effective hierarchical attentions between the latent and the global dual-path transformers. We also ablate different latent memory size, and set the default size L to be 16.

Training strategy. In addition, Tab. 8b shows that the proposed temporal consistency loss helps TubeFormer-DeepLab to learn clip-to-clip consistency, and improves the inference on longer videos than the training clip length (T), as demonstrated by +0.5 STQ gain. The proposed clip-level copy-paste (clip-paste) augments more training samples for tube-level segmentation, and further improves by +0.9 STQ.

Scaling. We study the scaling of TubeFormer-DeepLab in Tab. 8c. Pretraining on ImageNet-22k dataset brings +1.6 STQ and adding COCO to the training further gives +1.9 STQ. We also explore scaling up the backbone by stacking the axial-attention blocks in stage-4 by n times (TubeFormer-DeepLab- Bn). The increase of every n will introduce +13M parameters. We notice increasing the stack from $n = 1$ to $n = 3$ improves the STQ from 73.19 to 74.25. Further scaling to $n = 4$ starts to saturate, probably limited by the scale of KITTI-STEP dataset. We observe TubeFormer-DeepLab can further scale to $n = 4$ on larger-

method	L	F-L	M2L	L2M	STQ	SQ	AQ
TF-DL-Simple	0				68.36	74.93	62.38
TF-DL (with hierarchical dual-path transformer)	16	✓			70.03	76.83	63.83
	16	✓	✓		69.63	76.05	63.75
	16	✓		✓	69.64	76.75	63.18
	8	✓			69.39	75.74	63.54
	32	✓			69.57	76.71	63.1

(a) **Varying transformer attention types.** Frame-latent (F-L) attention is introduced in the proposed latent dual-path transformer, and includes latent-to-frame, latent-to-latent, and frame-to-latent attentions. We also ablate memory-to-latent (M2L) and latent-to-memory (L2M) attentions, and different latent memory size L .

method	STQ	SQ	AQ
TF-DL	70.03	76.83	63.83
+ temporal consistency	70.51	77.64	64.04
+ clip-paste	71.40	76.82	66.36

(b) Adding **temporal consistency loss** and **clip-level copy-paste**.

	n	INet	COCO	STQ	SQ	AQ
B1	1	1k		70.03	76.83	63.83
B1	1	1k+22k		72.28	76.27	67.01
B1	1	1k+22k	✓	73.19	78.11	68.58
B3	3	1k+22k	✓	74.25	78.31	70.04
B4	4	1k+22k	✓	73.68	78.16	69.46

(c) **Scaling** by stacking axial blocks in stage-4 of Axial-ResNet-50 by n times and **pretraining** on ImageNet-22K and COCO.

	STQ	SQ	AQ
TF-DL	70.03	76.83	63.83
<i>without</i> sharing semantic and panoptic	68.95	75.83	62.70
<i>without</i> split thing and stuff memory	68.96	75.77	62.76

(d) Ablating **architectural improvements**

Table 8. Ablation studies on KITTI-STEP *val* set.

scale datasets, where TubeFormer-DeepLab-B4 performed better on VSPW and YouTube-VIS datasets.

Architectural improvements. We ablate our new architectural designs: (1) sharing the semantic and panoptic predictions, and (2) splitting the global memory for separate thing and stuff classes. As shown in Tab. 8d, we observe a performance drop of -1.1 STQ by reverting the change of either (1) or (2) from TubeFormer-DeepLab.

4.5. Visualization

In Fig. 4, we visualize how the proposed hierarchical dual-path transformer performs attention onto the input clip of three consecutive frames. We first visualize the global memory attention by selecting four output regions of interest from TubeFormer-DeepLab video panoptic prediction. We probe the attention weights between the four tube-specific global memory embeddings and all the pixels. We see the global memory attention is spatio-temporally well separated for individual thing or stuff tubes.

In addition, we select four latent memory indices and visualize their attention maps in Fig. 4c. We find that some latent memory learns to spatially specialize on certain areas (left vs right side of the scene) or attends to semantically-similar regions (cars or backgrounds) to facilitate per-frame attention. With the hierarchical attentions made by the global and latent dual-path transformers, TubeFormer-DeepLab can be a successful *tube* transformer.

Finally, we provide more visualizations for each video segmentation task in Sec. 6 and *video* prediction results at <https://youtu.be/twoJyHpkTbQ>.

5. More Experimental Results

In this section, we provide more experimental results, comparing our methods with *published* works in detail. We do not include the *unpublished and concurrent* ICCV 2021 challenge entries, which usually adopt complicated pipelines, *e.g.*, model ensembles, separate models for different sub-tasks (*e.g.*, tracking, and segmentation), multi-scale inference, or pseudo labels. In the tables, we explicitly list the adopted backbones and decoders for a detailed comparison. We note that most of the state-of-the-art approaches for different video segmentation tasks have fundamentally diverged, while our proposed TubeFormer-DeepLab is a simple and unified system for general video segmentation tasks.

[VPS] Tab. 9 summarizes our results on KITTI-STEP *val* set. As shown in the table, our TubeFormer-DeepLab-B1, employing ResNet-50 [35] and axial-attention [76], significantly outperforms Motion-DeepLab [79] (w/ ResNet-50, dual-ASPP [15] and dual decoders [17]) and VPSNet [44] (w/ ResNet-50, FPN [55], and Mask R-CNN [34] multi-head predictions) by **+12** and **+14** STQ, respectively. We also report the results in the VPQ metric [44] (another popular video panoptic segmentation metric). Similarly, our model performs better than Motion-DeepLab and VPSNet by **+11.1** and **+8.1** VPQ.

[VSS] In Tab. 10, we report our results on VSPW *val* set. As shown in the table, our TubeFormer-DeepLab-B1, employing ResNet-50 and axial-attention, significantly outperforms TCB [64] (w/ spatial-temporal OCRNet [88] and a novel memory scheme) by **+20.2** mIoU. Our TubeFormer-DeepLab-B1 also shows better results in terms of VC8 and

VC16 (another video semantic segmentation metrics proposed in [64]).

[VIS] Tab. 11 summarizes our results on Youtube-VIS-2019 *val* set, along with several state-of-the-art methods.

Among the methods that predict non-overlapping segmentation, our TubeFormer-DeepLab-B1 (per-pixel), employing ResNet-50 and axial-attention, outperforms STEm-Seg [1] (using ResNet-50, FPN, and their novel 3D convolution-based TSE decoder with multi-head predictions) by **+5.8** AP. Our TubeFormer-DeepLab-B1 (per-pixel) is also better than STEm-Seg with ResNet-101 backbone by **+1.8** AP. If we also increase our backbone capacity, our TubeFormer-DeepLab-B4 (per-pixel) performs better than STEm-Seg w/ ResNet-101 by **+10.8** AP.

Our TubeFormer-DeepLab-B1 (per-pixel) performs worse than other state-of-the-art methods, including MaskProp [5], Seq Mask R-CNN [54], and the concurrent work IFC [40], since our per-pixel inference scheme generates non-overlapping predictions (*i.e.*, only one prediction for each pixel in the final output), which is disfavored by the track AP metric. To bridge the gap, we adopt the mask-wise merging scheme (denoted as per-mask) [21, 87], where each object query generates a mask proposal. The per-mask scheme significantly improves over the per-pixel scheme by more than 2 AP in the TubeFormer-DeepLab framework. Our large model TubeFormer-DeepLab-B4 with per-mask scheme outperforms MaskProp, VisTR, and IFC, and performs comparably with the best model Seq Mask R-CNN, which relies on STM [66]-like structure to propagate mask proposals through the whole sequence.

Notably, our model yields the best AR_1 and AR_{10} (**+3.9** and **+3.0** AR better than the second best Seq Mask-RCNN method, respectively), demonstrating the high segmentation quality in our predictions. Also, TubeFormer-DeepLab employs a smaller clip value ($T = 5$), while other state-of-the-art proposal-based approaches use a large value of clip ($T = 13$ or 36).

6. Visualization

In Fig. 5, 6, and 7, we visualize how the proposed hierarchical dual-path transformer performs attention for video panoptic/semantic/instance segmentation tasks (VPS, VSS, and VIS, respectively). We use input clips of three consecutive frames for visualization. For each sample, we select several output tubes of interest from the TubeFormer-DeepLab prediction. In column-b, we probe the attention weights between the selected tube-specific global memory embeddings and all the pixels. Across all three tasks, we observe the global memory attention is spatio-temporally clustered for individual tube regions, while respecting different requirements among the tasks. That is, one global memory answers for each semantic category in VSS, but

method	backbone	decoder	STQ	SQ	AQ	VPQ
Motion-DeepLab [79]	ResNet-50 + dual ASPP [15]	dual DeepLabv3+ decoder [17] w/ multi-heads	58.0	67.0	51.0	40.0
VPSNet [44]	ResNet-50 + FPN [55]	Mask R-CNN [34] style multi-heads	56.0	61.0	52.0	43.0
TF-DL-B1	ResNet-50 + axial-attention [76]†	tube-transformer	70.0	76.8	63.8	51.1

Table 9. [VPS] KITTI-STEP *val* set results. †: Axial attention blocks [76] are used in the last two stages.

method	backbone	decoder	mIoU	VC8	VC16
TCB [64]	ResNet-101	spatial-temporal OCRNet [88] + memory aggregation	37.8	87.9	84.0
TF-DL-B1	ResNet-50 + axial-attention [76]†	tube-transformer	58.0	90.1	86.8

Table 10. [VSS] VSPW *val* set results. †: Axial attention blocks [76] are used in the last two stages.

method	backbone	decoder	T	AP	AR ₁	AR ₁₀
MaskProp [5]	ResNet-50 + FPN [55] + HTC [11]	Mask R-CNN [34] style	13	40.0	-	-
	ResNet-101 + FPN [55] + HTC [11]	multi-heads	13	42.5	-	-
	ResNeXt-101 [82] + FPN [55] + HTC [11]	w/ mask refinement	13	44.3	-	-
	ResNeXt-101 [82] + FPN [55] + HTC [11] + deform.STSN [6, 23]	postprocessing	13	46.6	-	-
Seq Mask R-CNN [54]	ResNet-50 + FPN [55]	Mask R-CNN [34] style	36	40.4	41.1	49.7
	ResNet-101 + FPN [55]	multi-heads	36	43.8	46.3	52.6
	ResNeXt-101 [82] + FPN [55]	w/ many proposals	36	47.6	46.3	56.0
VisTR [77]	ResNet-50	DETR [10] style transformer	36	36.2	37.2	42.4
	ResNet-101		36	40.1	38.3	44.9
IFC [40]	ResNet-50 + FPN [55]	DETR [10] style transformer	5	41.0	43.5	52.7
	ResNet-50 + FPN [55]		36	42.8	43.8	51.2
	ResNet-101 + FPN [55]		36	44.6	44.0	52.1
STEm-Seg [1]	ResNet-50 + FPN [55]	3D Conv-based TSE [1]	8	30.6	31.6	37.1
	ResNet-101 + FPN [55]	w/ multi-heads	8	34.6	34.4	41.6
TF-DL-B1 (per-pixel)	ResNet-50 + axial-attention [76]†	tube-transformer	5	36.4	40.8	49.5
	(per-mask)		ResNet-50 + axial-attention [76]†	5	38.8	44.0
TF-DL-B4 (per-pixel)	ResNet-50-n4 + axial-attention [76]†		5	45.4	48.3	56.9
	(per-mask)		ResNet-50-n4 + axial-attention [76]†	5	47.5	50.2

Table 11. [VIS] YouTube-VIS-2019 *val* set results. †: Axial attention blocks [76] are used in the last two stages. ResNet-50-n4 scales the number of layers in stage-4 by 4 times (*i.e.*, 24 blocks in total), resulting in a backbone with 104 layers.

for each instance identity in VIS, while both cases appear in VPS task.

In column-c, we select four latent memory indices and visualize their attention maps. Commonly for all tasks, some latent memory learns to spatially specialize on certain areas (left vs right side of the scene) or attends to the tube boundaries. Interestingly, we find that some latent memory focuses on relatively far-away region (Fig. 5c-bottom right), which often requires more attention. Sometimes, it has more interests to the moving object parts or small objects (*e.g.*, *moving arms* and *a road-block cone* in Fig. 6c-bottom left and bottom right, respectively).

The task-specific behavior of the latent memory can be also compared between Fig. 6c and Fig. 7c. The latent memory in VSS does not distinguish instances of a same semantic class. In contrast, the attention is instance-specific in VIS. As shown in Fig. 7c-top left, the *occluded noses of two elephants* are highlighted, which is expected to help the instance discrimination. Also, different latent memory attends to a single, or different multiples of the instances.

Additionally, Fig. 8 visualizes our depth-aware video

panoptic segmentation results on SemKITTI-DVPS dataset, where TubeFormer-DeepLab is able to generate temporally consistent panoptic segmentation and monocular depth estimation results.

7. Discussion

We notice that recently there is some hype in the literature regarding the development of *universal* or *unified* segmentation models for semantic, instance, and panoptic segmentation. We would like to emphasize that the goal of panoptic segmentation is to unify semantic and instance segmentation, and thus a well-designed panoptic segmentation model should naturally demonstrate a fair performance on semantic segmentation and instance segmentation as well. For example, Panoptic-DeepLab [20] and its Naive-Student version [12] already demonstrate that a modern panoptic segmentation model could simultaneously achieve state-of-the-art performance on semantic, instance, and panoptic segmentation. Our work follows the same direction by working on the video segmentation tasks.

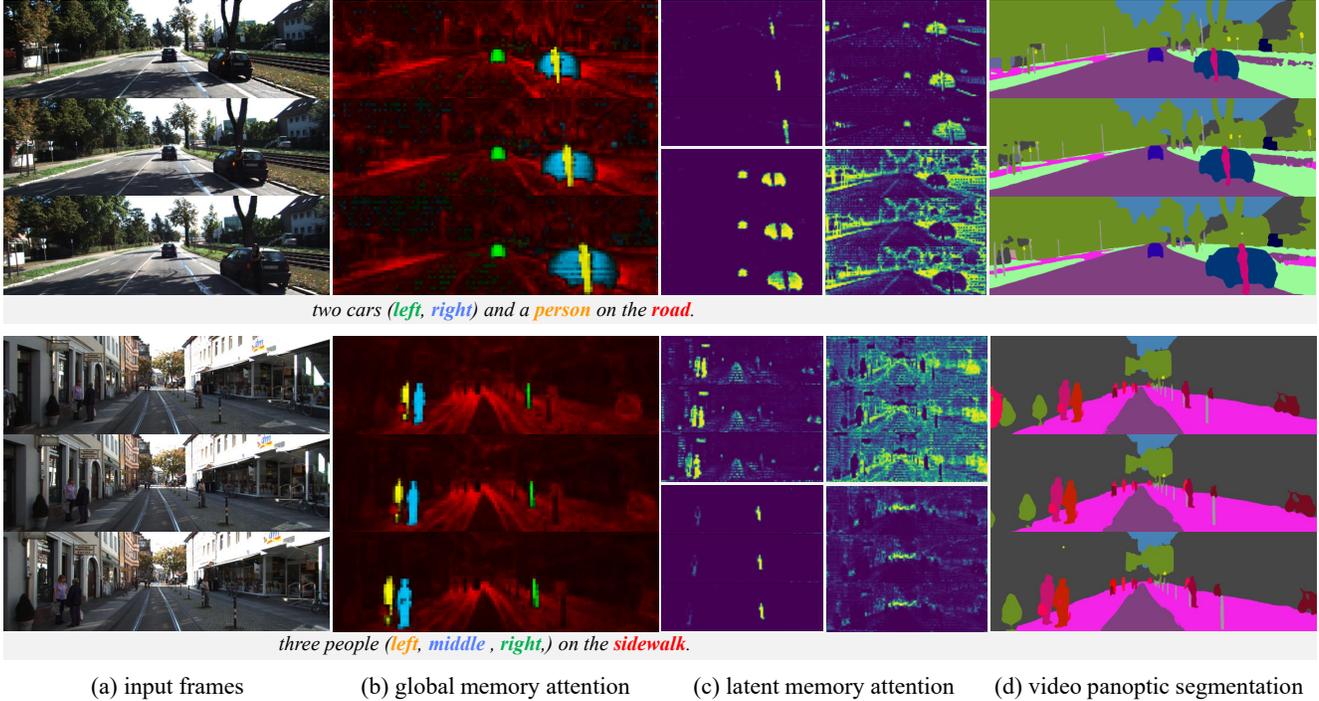


Figure 5. [VPS] Visualization on KITTI-STEP sequence. From left to right: input frames ($T=3$), global memory attention, latent memory attention, and video *panoptic* segmentation results. The global memory attention is selected for predicted tube regions of interest, and the latent memory attention is selected for 4 (out of $L=16$) latent memory.

8. Limitations

Currently, the proposed TubeFormer-DeepLab performs clip-level video segmentation with the clip value $T = 2$ (for VPS and VSS) or $T = 5$ (for VIS). Our model thus performs short-term tracking and may miss objects that have track lengths larger than the used clip value. This limitation is also reflected in the AQ (association quality) reported in Tab. 1 of the main paper (*i.e.*, KITTI-STEP *test* set results). We leave the question about how to efficiently incorporate long-term tracking to TubeFormer-DeepLab for future work.

In any case, our proposed TubeFormer-DeepLab presents the first attempt to tackle multiple video segmentation tasks from a unified approach. We hope our simple and effective model could serve as a solid baseline for future research.

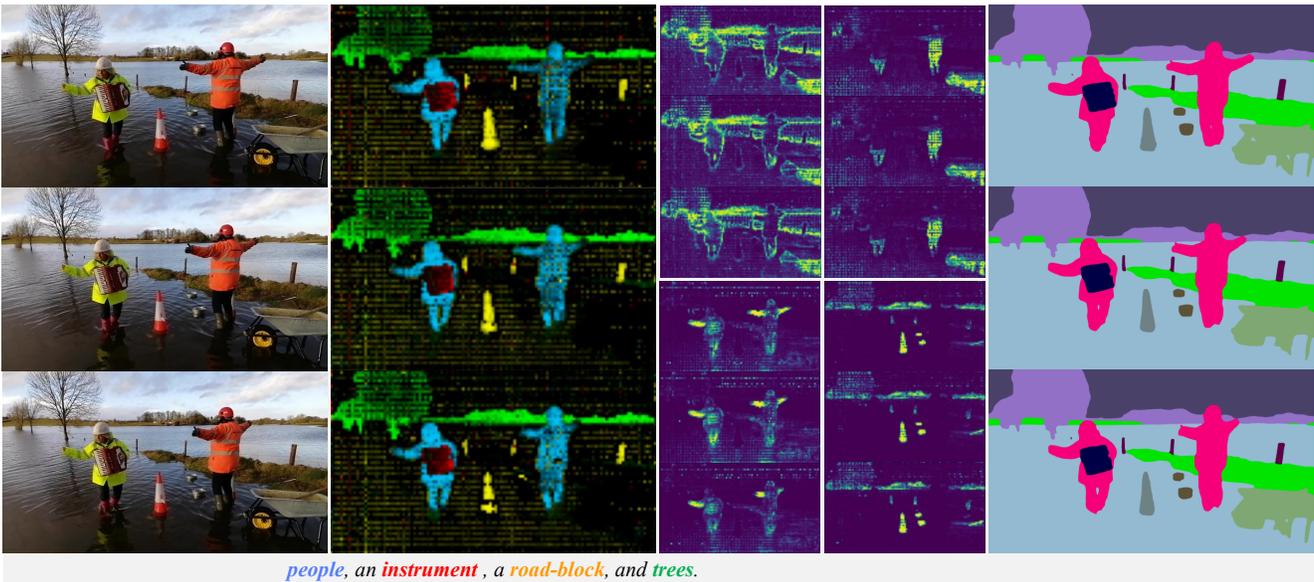
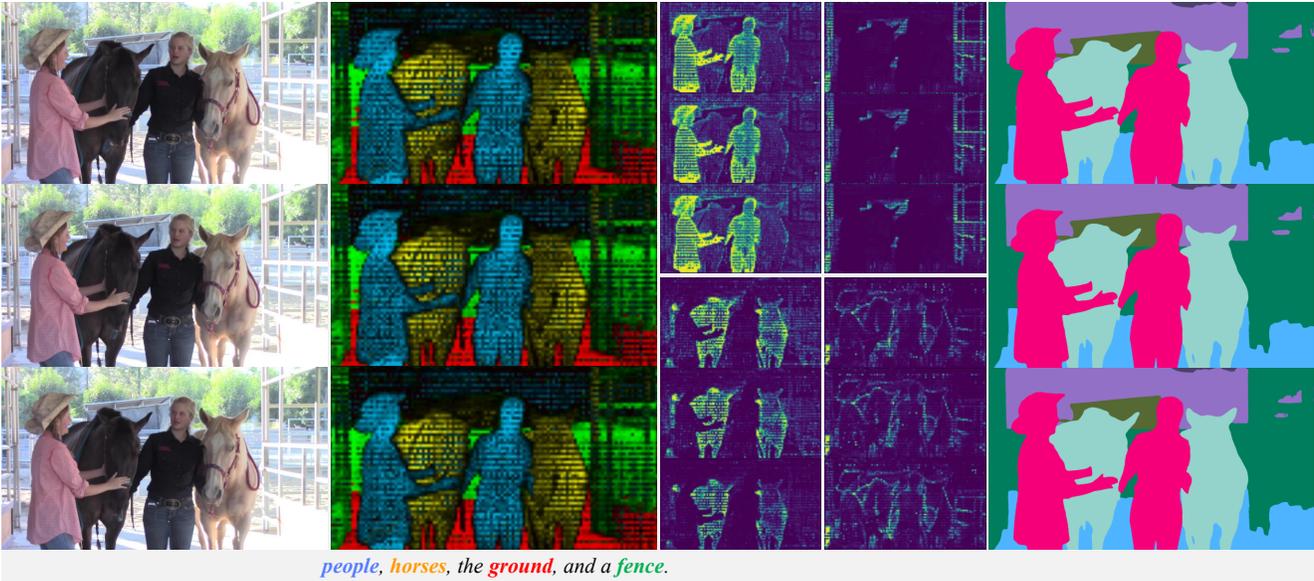
9. Conclusion

We introduced TubeFormer-DeepLab, a novel architecture based on mask transformers for video segmentation. Video segmentation tasks, particularly video semantic/instance/panoptic segmentation, have been tackled by fundamentally divergent models. We proposed a new paradigm that formulates video segmentation tasks as the

problem of partitioning video tubes with different predicted labels. TubeFormer-DeepLab, directly predicting class-labeled tubes, provides a general solution to multiple video segmentation tasks. We hope our approach will inspire future research in the unification of video segmentation tasks.

References

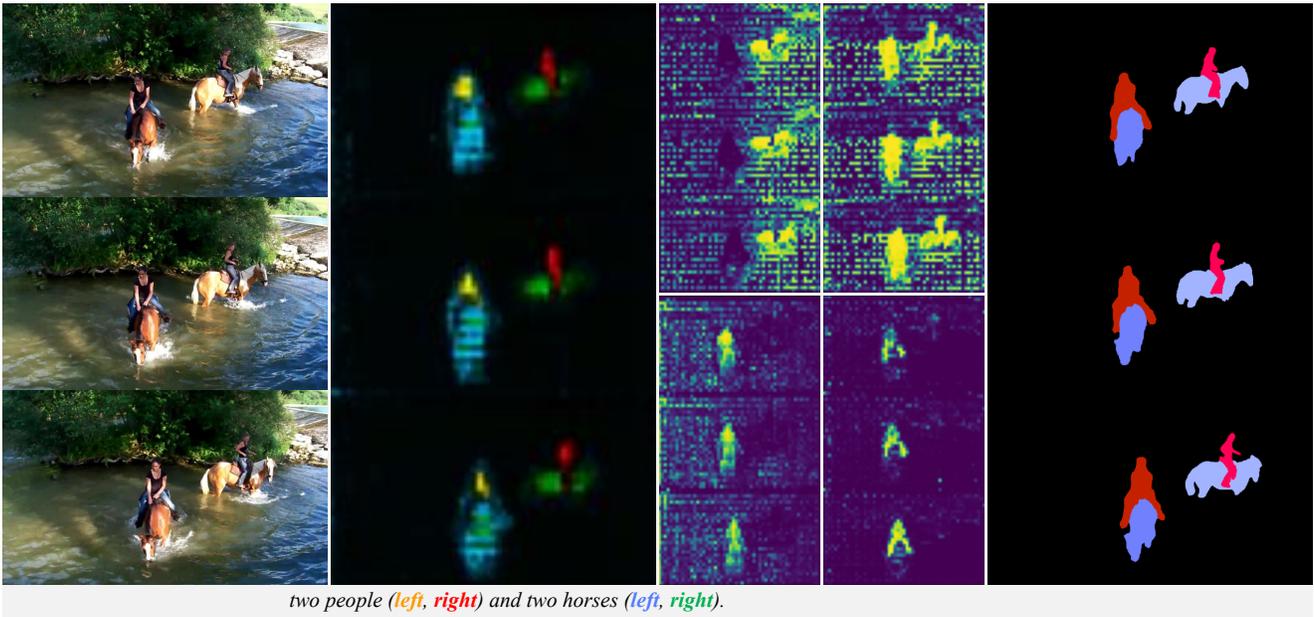
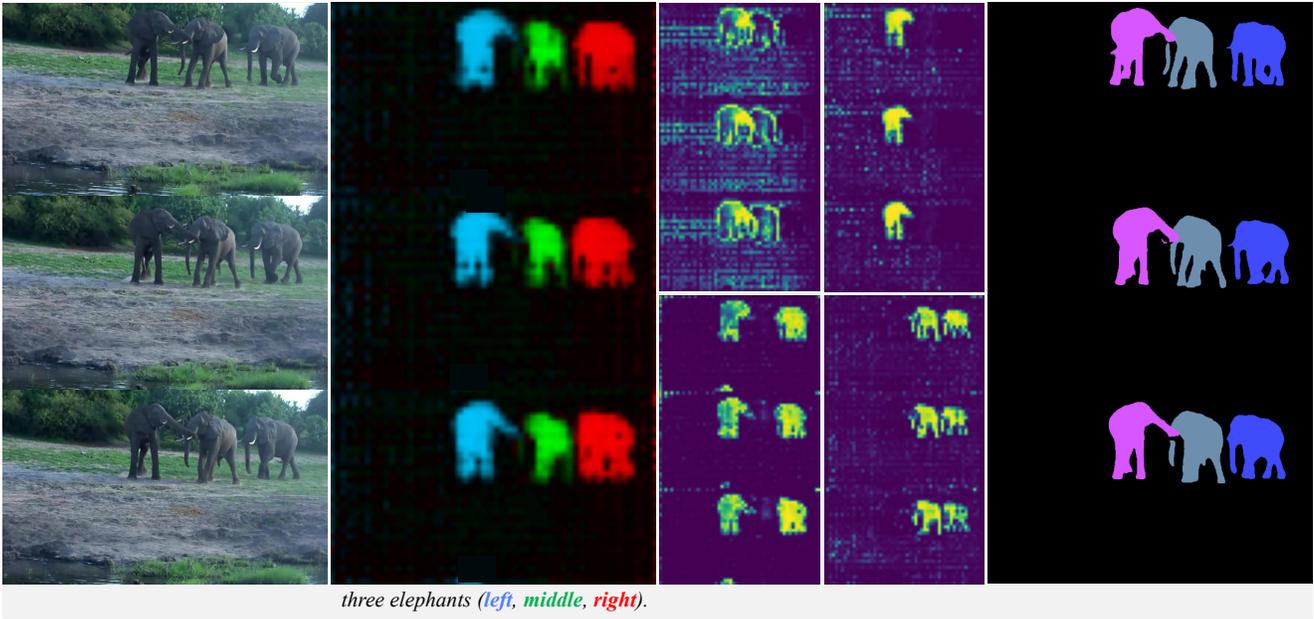
- [1] Ali Athar, Sabarinath Mahadevan, Aljoša Ošep, Laura Leal-Taixé, and Bastian Leibe. STEM-Seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020. 7, 9, 10
- [2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv:2106.08254*, 2021. 6
- [3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 6
- [4] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019. 2
- [5] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*, 2020. 2, 6, 7, 9, 10
- [6] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. Object detection in video with spatiotemporal sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 331–346, 2018. 10



(a) input frames (b) global memory attention (c) latent memory attention (d) video semantic segmentation

Figure 6. **[VSS] Visualization on VSPW sequence.** From left to right: input frames ($T=3$), global memory attention, latent memory attention, and video *semantic* segmentation results. The global memory attention is selected for predicted tube regions of interest, and the latent memory attention is selected for 4 (out of $L=16$) latent memory.

- [7] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009. 2
- [8] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and Recognition Using Structure from Motion Point Clouds. In *ECCV*, 2008. 1, 2
- [9] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*, 2020. 2, 7
- [10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 10
- [11] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiao-xiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 2, 10



(a) input frames

(b) global memory attention

(c) latent memory attention

(d) video instance segmentation

Figure 7. [VIS] Visualization on Youtube-VIS 2019 sequence. From left to right: input frames ($T=3$), global memory attention, latent memory attention, and video *instance* segmentation results. The global memory attention is selected for predicted tube regions of interest, and the latent memory attention is selected for 4 (out of $L=16$) latent memory.

[12] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-Student: Leveraging Semi-Supervised Learning in Video Sequences for Urban Scene Segmentation. In *ECCV*, 2020. 10

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image seg-

mentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015. 2

[14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE TPAMI*, 2017. 5

[15] Liang-Chieh Chen, George Papandreou, Florian Schroff, and

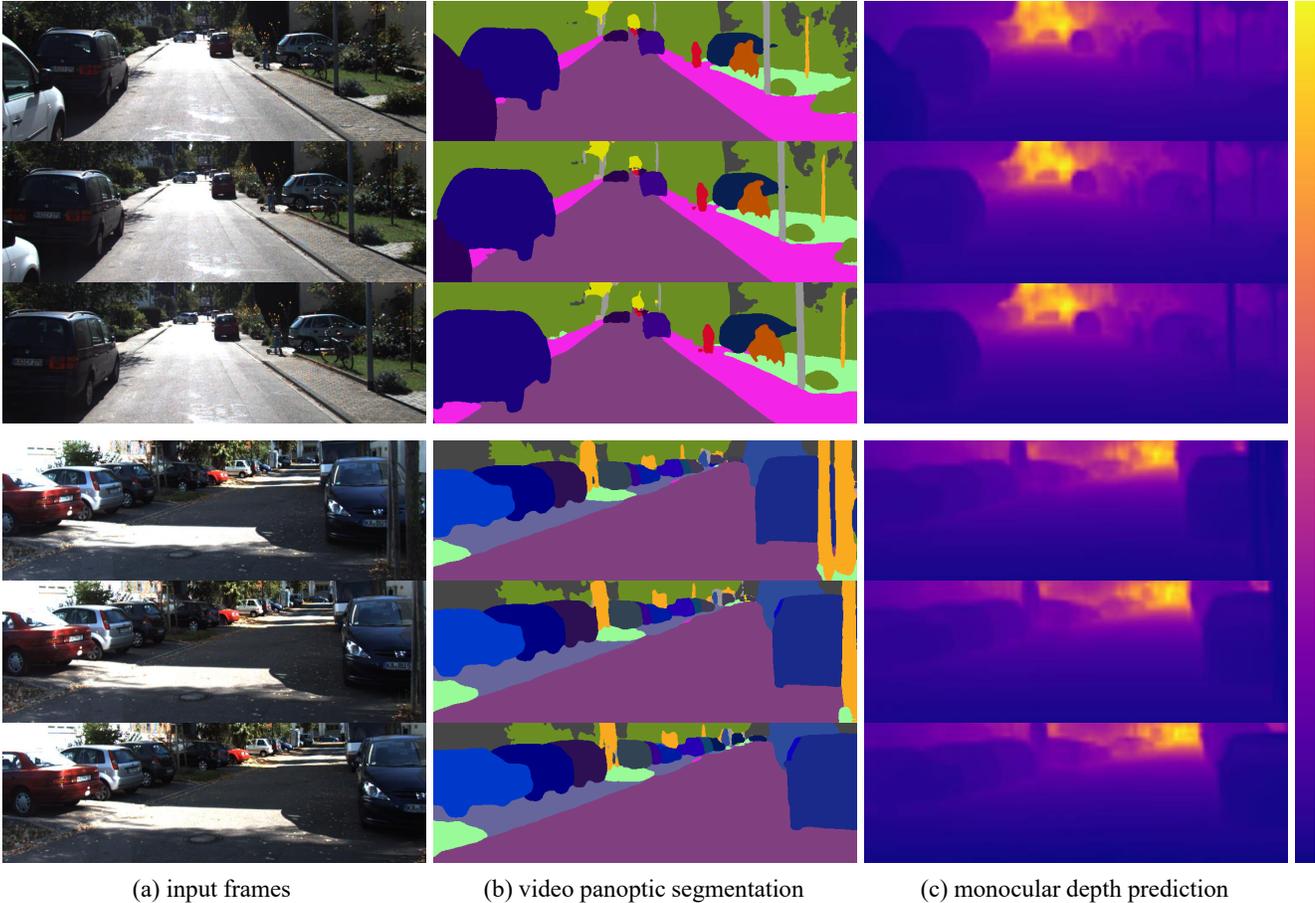


Figure 8. **[DVPS] Visualization on SemKITTI-DVPS sequence.** From left to right: input frames ($T=3$), video *panoptic* segmentation, and monocular depth prediction results. As the attentions are very similar to those in KITTI-STEP (Fig. 5), here we focus on the depth visualization.

- Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017. 3, 9, 10
- [16] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic segmentation. *arXiv:2011.11675*, 2020. 6
- [17] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2, 5, 9, 10
- [18] Zixuan Chen, Junhong Zou, and Xiaotao Wang. Semantic Segmentation on VSPW Dataset through Aggregation of Transformer Models. In *ICCV The 1st Video Scene Parsing in the Wild Challenge Workshop*, 2021. 7
- [19] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab. In *ICCV COCO + Mapillary Joint Recognition Challenge Workshop*, 2019. 2
- [20] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020. 3, 10
- [21] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 5, 9
- [22] 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 *CVPR*, 2016. 2, 6
- [23] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2, 3, 10
- [24] Patrick Dendorfer, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. MOTChallenge: A Benchmark for Single-camera Multiple Target Tracking. *IJCV*, 2020. 2
- [25] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 5, 6
- [26] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 2

- [27] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *ICCV*, 2019. 2, 5
- [28] Yang Fu, Linjie Yang, Ding Liu, Thomas S Huang, and Humphrey Shi. Compfeat: Comprehensive feature aggregation for video instance segmentation. In *AAAI*, 2021. 2
- [29] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video CNNs through representation warping. In *ICCV*, 2017. 1, 2
- [30] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 2013. 2
- [31] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 5
- [32] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 2, 5
- [33] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 2
- [34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 3, 9, 10
- [35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 9
- [36] Xingjian He, Weining Wang, Zhiyong Xu, Hao Wang, Jie Jiang, and Jing Liu. Exploiting Spatial-Temporal Semantic Consistency for Video Scene Parsing. In *ICCV The 1st Video Scene Parsing in the Wild Challenge Workshop*, 2021. 6, 7
- [37] Xuming He, Richard S Zemel, and Miguel Á Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, 2004. 2
- [38] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 2
- [39] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019. 2
- [40] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. Video instance segmentation using inter-frame communication transformers. In *NeurIPS*, 2021. 2, 4, 7, 9, 10
- [41] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, 2015. 2
- [42] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *CVPR*, 2019. 2
- [43] Zhenchao Jin, Dongdong Yu, Kai Su, Zehuan Yuan, and Changhu Wang. Memory Based Video Scene Parsing. In *ICCV The 1st Video Scene Parsing in the Wild Challenge Workshop*, 2021. 7
- [44] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *CVPR*, 2020. 1, 3, 5, 9, 10
- [45] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 2
- [46] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 2
- [47] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 4
- [48] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2
- [49] Xiangtai Li, Haobo Yuan, Yibo Yang, Lefei Zhang, Yunhai Tong, and Dacheng Tao. PolyphonicFormer: Unified Query Learning for Depth-aware Video Panoptic Segmentation. In *ICCV Segmentation and Tracking Every Point and Pixel: 6th Workshop on Benchmarking Multi-Target Tracking*, 2021. 7
- [50] Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. Video k-net: A simple, strong, and unified baseline for video segmentation. In *CVPR*, 2022. 3
- [51] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *CVPR*, 2019. 2
- [52] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *CVPR*, 2018. 2
- [53] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. In *CVPR*, 2021. 2, 6
- [54] Huaijia Lin, Ruizheng Wu, Shu Liu, Jiangbo Lu, and Jiaya Jia. Video instance segmentation with a propose-reduce paradigm. In *ICCV*, 2021. 2, 7, 9, 10
- [55] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 9, 10
- [56] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6
- [57] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *CVPR*, 2021. 2
- [58] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 2
- [59] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 6
- [60] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2
- [61] Jincheng Lu, Yue He, Minyue Jiang, Meng Xia, Wei Zhang, Xiao Tan, Yingying Li, Hao Sun, and Errui Ding. Robust Video Panoptic Segmentation and Tracking. In *ICCV Segmentation and Tracking Every Point and Pixel: 6th Workshop on Benchmarking Multi-Target Tracking*, 2021. 6
- [62] Jonathon Luiten, Aljoša Ošep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe.

- HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking. *IJCV*, 2020. 3
- [63] Jiayu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *CVPR*, 2022. 2, 6, 7
- [64] Jiayu Miao, Yunchao Wei, Yu Wu, Chen Liang, Guangrui Li, and Yi Yang. Vspw: A large-scale dataset for video scene parsing in the wild. In *CVPR*, 2021. 1, 2, 3, 6, 7, 9, 10
- [65] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *CVPR*, 2018. 2
- [66] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 7, 9
- [67] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *ECCV*, 2020. 2
- [68] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *CVPR*, 2021. 2
- [69] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. ViP-DeepLab: Learning Visual Perception with Depth-aware Video Panoptic Segmentation. In *CVPR*, 2021. 2, 3, 5, 6, 7
- [70] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008. 2
- [71] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 2
- [72] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019. 2
- [73] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2
- [74] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019. 1, 2, 3, 6
- [75] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers. In *CVPR*, 2021. 2, 3, 4, 5, 6
- [76] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In *ECCV*, 2020. 2, 4, 6, 9, 10
- [77] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021. 2, 7, 10
- [78] Mark Weber, Huiyu Wang, Siyuan Qiao, Jun Xie, Maxwell D. Collins, Yukun Zhu, Liangzhe Yuan, Dahun Kim, Qihang Yu, Daniel Cremers, Laura Leal-Taixe, Alan L. Yuille, Florian Schroff, Hartwig Adam, and Liang-Chieh Chen. DeepLab2: A TensorFlow Library for Deep Labeling. *arXiv: 2106.09748*, 2021. 6
- [79] Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, Aljosa Osep, Laura Leal-Taixe, and Liang-Chieh Chen. Step: Segmenting and tracking every pixel. In *NeurIPS Track on Datasets and Benchmarks*, 2021. 1, 2, 3, 6, 8, 9, 10
- [80] Sanghyun Woo, Dahun Kim, Joon-Young Lee, and In So Kweon. Learning to associate every segment for video panoptic segmentation. In *CVPR*, 2021. 2, 3
- [81] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 2, 7
- [82] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 10
- [83] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. UPSNet: A unified panoptic segmentation network. In *CVPR*, 2019. 2
- [84] Linjie Yang, Yuchen Fan, and Ning Xu. Video Instance Segmentation. In *ICCV*, 2019. 1, 2, 3, 6, 7
- [85] Shusheng Yang, Yuxin Fang, Xinggong Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learning for fast online video instance segmentation. In *ICCV*, 2021. 7
- [86] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. DeeperLab: Single-shot image parser. *arXiv:1902.05093*, 2019. 2
- [87] Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. CMT-DeepLab: Clustering Mask Transformers for Panoptic Segmentation. In *CVPR*, 2022. 5, 9
- [88] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 2, 6, 9, 10
- [89] Haotian Zhang, Yizhou Wang, Zhongyu Jiang, Cheng-Yen Yang, Jie Mei, Jiarui Cai, Jenq-Neng Hwang, Kwang-Ju Kim, and Pyong-Kun Kim. U3D-MOLTS: Unified 3D Monocular Object Localization, Tracking and Segmentation. In *ICCV Segmenting and Tracking Every Point and Pixel: 6th Workshop on Benchmarking Multi-Target Tracking*, 2021. 6
- [90] Songyang Zhang, Xuming He, and Shipeng Yan. Latentgmn: Learning efficient non-local relations for visual recognition. In *ICML*, 2019. 2, 4
- [91] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *NeurIPS*, 34:10326–10338, 2021. 3
- [92] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2
- [93] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2

- [94] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *CVPR*, 2017. [1](#), [2](#), [3](#)
- [95] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *CVPR*, 2019. [2](#)
- [96] Zhen Zhu, Mengde Xu, Song Bai, Tengting Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *ICCV*, 2019. [2](#)