Tube-Link: A Flexible Cross Tube Baseline for Universal Video Segmentation Supplementary

A. Appendix

Overview. In addition to the main paper, we further list the following details and more experiment results as supplementary to our work.

1. More detailed description and comparison of Tube-Link. (Sec. A.1)
2. Detailed experiment settings and implementation details for each dataset. (Sec. A.2)
3. More ablation studies and experiment results.(Sec. A.3)
4. More visual results. (Sec. A.4)

A.1. More Detailed Description of Tube-Link

This section presents the method details, including several baselines and Tube-Link inference for different datasets. Then, due to the limited pages in the main paper, we compare several closely related works in VIS and VPS in detail.

Video K-Net+ Baseline. This baseline is based on two previous state-of-the-art methods, including Video K-Net [14] and Mask2Former [4]. In particular, Video K-Net is based on K-Net [31], an image panoptic segmentation model. We replace K-Net with Mask2Former [4], and the remaining parts are the same as the Video K-Net. Since the performance of Mask2Former is better than K-Net on image segmentation datasets, Video K-Net+ serves as the strong online baseline for both VPS and VSS tasks.

Detailed Inference Procedure of Panoptic Matching. During the inference, we perform tube-level panoptic matching according to learned association embeddings only on final panoptic tube masks. In particular, we save the index of each global query from the final panoptic tube results, and then we use these indexed queries via the embedding head Emb.

Detailed Inference Procedure of VSS and VIS. Since VSS does not need tracking, we do not apply the extra tracking embedding during the inference. Instead, the tube mask logits are obtained directly from the dot product between global queries and spatial-temporal decoder features. The final segmentation labels are directly obtained via argmax on predicted logits. For VIS, we follow nearly the same procedure as VPS, except no stuff queries are involved.

More Detailed Comparison with Previous Nearly Online Approaches in VIS and VPS. In addition to the main paper, in Tab. 1, we present a more detailed comparison with previous works on VIS and VPS. From the table, our method uses tube-wised matching and supports all three video segmentation tasks in one architecture.

In particular, both SLOT-VPS [32] and SeqFormer [25] also adopt multiple frames design. However, there are no data association processes involved. Moreover, they are designed for VIS and VPS, individually, and our method outperforms the SeqFormer on two VIS datasets, as shown in Tab. 6. Furthermore, unlike SLOT-VPS and SeqFormer, our method can handle long video inputs.

Gen-VIS [9] also adopts the tube-wised design, which combines the nearly online method and online method in one framework. However, it can not support other video segmentation tasks, including VSS and VPS. Moreover, it is not verified in more complex scenes, including the driving dataset KITTI-STEP [24] and the recent more challenging dataset VIP-Seg [17]. In contrast, our Tube-Link is fully verified by three different video segmentation tasks and five different datasets. In particular, using the same ResNet50 backbone and detector [4], even without COCO video joint training, our method works better than Gen-VIS [9], as shown in Tab. 6.

A.2. Implementation Details

Detailed Training and Inference on VIP-Seg. We use the COCO-pretrained model following [17]. The entire training process takes eight epochs. We adopt multiscale training where the scale ranges from 1.0 to 2.0 of the original image size, and then we apply a random crop of 720 $\times$ 720 patches. In particular, we perform the augmentation for each frame in the sampled subclips. For the inference, the subclip window size is set to six by default. We pad the remaining frames in the last subclip by repeating the last frame. We drop the padded results for evaluation.

Table 1: Different Setting Comparison with previous VIS and VPS methods.

Method	VSS	VIS	VPS	Online	Nearly Online	Joint Mulitple Frames	Frame Matching	Tube Matching	Mask Matching	No Association (use Query Index)
CFFM [20]	✓				✓	✓				✓
MRCFA [21]	✓				✓	✓				✓
Cross-VIS [29]		✓		✓			✓			
IDOL [26]		✓		✓			✓			
SeqFormer [25]		✓			✓	✓				✓
EfficientVIS [27]		✓			✓	✓				✓
VITA [10]		✓			✓	✓				✓
Min-VIS [11]		✓		✓			✓			
IFC [12]		✓			✓	✓		✓		
Gen-VIS [9]		✓		✓	✓	✓		✓		
SLOT-VPS [32]			✓		✓	✓				✓
TubeFormer [13]	✓	✓	✓		✓	✓			✓	
Video K-Net [14]	✓	✓	✓	✓			✓			
Our Tube-Link	✓	✓	✓	✓	✓	✓		✓		

Detailed COCO pretraining setting. For COCO [15] panoptic segmentation dataset pretraining, all the models are trained following original Mask2Former settings [31]. We adopt the multiscale training setting as previous work [2] by resizing the input images such that the shortest side is at least 480 and 800 pixels, while the longest size is at most 1333. For data augmentation, we use the default large-scale jittering (LSJ) augmentation with a random scale sampled from the range 0.1 to 2.0 with the crop size of 1024 $\times$ 1024. For ResNet50 [8] and Swin-base [16] model, we train the model with 50 epochs following the original settings. For STDC model [6], we train the model for 36 epochs.

Detailed Training and Inference on KITTI-STEP dataset. For KITTI-STEP training, we follow previous Motion-Deeplab [24] and Video K-Net [14], we adopt multiscale training where the scale ranges from 1.0 to 2.0 of origin images size. We then apply a random crop of 384 $\times$ 1248 patches. The total training epoch is set to 12. The inference procedure is the same as Cityscapes-VPS dataset. Following the previous works [24, 14], we also use Cityscapes dataset [5] pretraining before training on STEP. Pretraining on Cityscapes STEP datasets further leads to 3% VPQ and 2% STQ improvements. We adopt the same inference pipeline as VIP-Seg, where we set the subclip window size to 2. We **do not** pre-train our model on the COCO dataset for a fair comparison.

Detailed Training and Inference on VSPW dataset. We adopt nearly the same training pipeline for VSPW as VIP-Seg. The main difference is that we adopt longer training epochs, where we set the training epochs to 12, where we find about 1% mIoU gain over different baselines. Moreover, we remove the tracking loss since we only focus on segmentation quality.

Detailed Training and Inference on Youtube-VIS-2019/2021 datasets. We follow the same setting as Mask2Former-VIS [3]. We train our models for 6k iterations, with a batch size of 16 for YouTubeVIS-2019 and 8k iterations for YouTubeVIS-2021. All models are initialized with COCO instance segmentation models of Mask2Former. Different from previous SOTA VIS models [10, 25, 26], we only use YouTubeVIS training data, and *do not* use COCO video images for data augmentation. Moreover, we also do not apply clip-wised copy-paste that is used in TubeFormer [13]. The same training procedure is adopted for the OVIS dataset as well.

Table 2: **More Ablation on Tube-Wised Matching in Youtube-VIS dataset.**

Settings	Youtube-VIS-2019	Youtube-VIS-2021
tube size=1	47.8	44.2
tube size=2	49.8	45.9
tube size=3	51.3	46.2
tube size=4	52.8	47.9
tube size=6	51.2	46.8

Table 3: **Ablation on Inference with Overlapped Frames.** We use the STDC-v1 backbone. The subclip window size is 6.

Settings	STQ	VPQ	SQ	FPS
No Overlapping	32.0	30.6	28.4	16.2
Overlapping=1	31.0	30.5	28.5	14.6
Overlapping=2	32.3	31.2	29.1	10.2
Overlapping=4	33.1	31.6	28.6	8.4

Table 4: **Ablation on Effect of COCO Pretraining.** We use the STDC-v1 backbone.

Settings	Method	STQ	VPQ	SQ
w COCO pretrained	Video K-Net+	26.1	25.8	25.2
w/o COCO pretrained	Video K-Net+	12.4	12.4	18.3
w COCO pretrained	Tube-Link	32.0	30.6	28.4
w/o COCO pretrained	Tube-Link	21.8	16.8	20.3

Table 5: **Ablation on Training Epochs.** We use the STDC-v1 backbone.

Settings	STQ	VPQ	SQ
Epoch=4	29.2	28.1	26.5
Epoch=8	32.0	30.6	28.4
Epoch=12	31.6	30.8	29.1

A.3. More Ablations and Experiment Results

In this section, we first present more detailed ablations for Tube-Link. Then, we present more detailed results on several datasets, including VIS datasets [28], OVIS dataset [19] and VSPW test set [18].

More Ablations on Effectiveness of Tube-Wised Matching. In Tab. 2, we present more detailed ablations on tube size in Youtube-VIS. Note that, for simplicity, the input sub-clip size is the same as the tube size. As we enlarge the tube size, we find a significant improvement in the final performance. After enlarging the size to 4, the performance is the best. Using a tube size of 6, the performance slightly degrades. However, it still performs better than single-frame matching. All the models are trained under the same tube size (default is 2). The findings also verify our motivation for using clip-level matching, which shares similar findings on the VIP-Seg dataset in the main paper.

Inference with Overlapped Frames in VIP-Seg. In Tab. 3, we explore the effect of the overlapping size for nearby windows. As shown in that table, increasing the overlapping size leads to better performance for all three metrics: VPQ, STQ, and SQ. This is because we can use multiple frames twice, which leads to more consistent segmentation results. Moreover, instances in smaller windows are easier to be tracked. However, to save computation costs and increase inference speed, we do not introduce overlapping during inference. All the results in the main paper use non-overlapping inference.

Effect of COCO Pretraining in VIP-Seg. In Tab. 4, we show the effect of COCO pretraining on both Video K-Net+ and our Tube-Link. From the table, we can see that COCO pretraining plays an important role for VIP-Seg datasets, which shares the same conclusion with previous work [14, 17]. Without COCO pretraining, both Video K-Net+ and Tube-Link drop a lot. However, as shown in the gray area, our method *without* COCO pretraining outperforms the Video K-Net+ baseline by a large margin, where we still achieve over 8% STQ gain and 14% VPQ gain. The results suggest the effectiveness of our framework on better usage of temporal information.

Effect of Training Epoch on VIP-Seg. We perform ablation on training epochs as in Tab. 5. With more training epochs, we do not observe performance gain with the COCO pre-trained model due to the overfitting issues. We use eight training epochs by default for all models.

Impact of Quasi-Dense Tracker. We adopt the same quasi-dense tracker for all experiments in the main paper, and we can achieve 3.0% VPQ improvement upon the baseline. In Tab. 9, we perform an extra experiment by replacing our tracker with a naive tracker used in MinVIS, where we only found 0.2% mAP drop. This proves the robustness and generalizability of Tube-Link. In contrast, we add the quasi-dense tracker to MinVIS, and we only find 0.6% mAP improvements. Directly extending a method with tube matching leads to more improvements. The results also indicate that the effect of the tracker is not apparent on the Youtube-VIS dataset, since the instance number is limited and occlusion is not heavy. Thus, we adopt *simple tube-matching* for VIS datasets.

Detailed Results Youtube-VIS. In Tab. 6, we report the detailed results on Youtube-VIS-2019 and Youtube-VIS-2021 datasets. We follow the baseline method, Mask2Former-VIS [3]. As shown in that table, our method achieves all the best metrics on both datasets *without COCO video joint training or clip-wised copy-paste*.

More Results on Test Set.. Moreover, we also report our results on the KITTI-STEP test set. As shown in Tab. 8b, Our method can still achieve better results.

Detailed Results on OVIS. In Tab. 7, we also report our model results on OVIS. Again, without bells and whistles, our method achieves comparable results with IDOL. We use the ResNet50 backbone for a fair comparison.

A.4. Visual Results

Visual Comparison on Youtube-VIS-2019 dataset. In Fig. 1, we compare our Tube-Link with strong baseline Mask2Former-VIS with the same ResNet50 backbone. Our methods achieve more consistent tracking and segmentation results in two examples.

More Visual Results on VIP-Seg Dataset. In Fig. 2, we present more visual examples on our Tube-Link. Compared with the Video K-Net+ baseline, our method achieves better segmentation and tracking consistency.

Visual Results on KITTI-STEP Dataset. In Fig. 3, we present visual results on the KITTI-STEP dataset, where we achieve consistent segmentation and tracking on the driving scene.

Failure Cases Analysis. In Fig. 4, we show several failure cases on the KITTI-STEP and VIP-Seg datasets using our best models. We observe three error sources: (1). remote and small objects. (2). heavy occlusion. (3). segmentation consistency caused by camera motion. We will handle these issues in future work.

Table 6: **Detailed Results on the Youtube-VIS datasets (2019/2021).** We report the mAP metric. † adopts COCO video pseudo labels [10, 9, 10]. Axial means using the extra Axial Attention [22]. Our method does not apply these techniques for simplicity.

Method	Backbone	YTVIS-2019					YTVIS-2021				
-	-	AP	AP_{50}	AP_{75}	AR_1	AR_{10}	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
VISTR [23]	ResNet50	36.2	59.8	36.9	37.2	42.4	-	-	-	-	-
EfficientVIS [27]	ResNet-50	37.9	59.7	43.0	40.3	46.6	34.0	57.5	37.3	33.8	42.5
TubeFormer [13]	ResNet50 + Aixal	47.5	68.7	52.1	50.2	59.0	41.2	60.4	44.7	40.4	54.0
IFC [12]	ResNet50	41.2	65.1	44.6	42.3	49.6	35.2	55.9	37.7	32.6	42.9
Seqformer [25]	ResNet50	47.4	69.8	51.8	45.5	54.8	40.5	62.4	43.7	36.1	48.1
Mask2Former-VIS [3]	ResNet50	46.4	68.0	50.0	-	-	40.6	60.9	41.8	-	-
IDOL [26]	ResNet50	46.4	-	-	-	-	43.9	-	-	-	-
IDOL [26] †	ResNet50	49.5	-	-	-	-	-	-	-	-	-
VITA [10] †	ResNet50	49.8	72.6	54.5	49.4	61.0	45.7	67.4	49.5	40.9	53.6
Min-VIS [11]	ResNet50	47.4	69.0	52.1	45.7	55.7	44.2	66.0	48.1	39.2	51.7
Cross-VIS [29]	ResNet50	36.3	56.8	38.9	35.6	40.7	34.2	54.4	37.9	30.4	38.2
VISOLO [7]	ResNet50	38.6	56.3	43.7	35.7	42.5	36.9	54.7	40.2	30.6	40.9
GenVIS [9]	ResNet50	51.3	72.0	57.8	49.5	60.0	46.3	67.0	50.2	40.6	53.2
Tube-Link	ResNet50	52.8	75.4	56.5	49.3	59.9	47.9	70.0	50.2	42.3	55.2
SeqFormer [25]	Swin-large	59.3	82.1	66.4	51.7	64.4	51.8	74.6	58.2	42.8	58.1
Mask2Former-VIS [3]	Swin-large	60.4	84.4	67.0	-	-	52.6	76.4	57.2	-	-
IDOL [26]	Swin-large	61.5	-	-	-	-	56.1	-	-	-	-
IDOL [26]	Swin-large †	64.3	87.5	71.0	55.6	69.1	56.1	80.8	63.5	45.0	60.1
VITA [10] †	Swin-large	63.0	86.9	67.9	56.3	68.1	57.5	80.6	61.0	47.7	62.6
Min-VIS [11]	Swin-large	61.6	83.3	68.6	54.8	66.6	55.3	76.6	62.0	45.9	60.8
Tube-Link	Swin-large	64.6	86.6	71.3	55.9	69.1	58.4	79.4	64.3	47.5	63.6

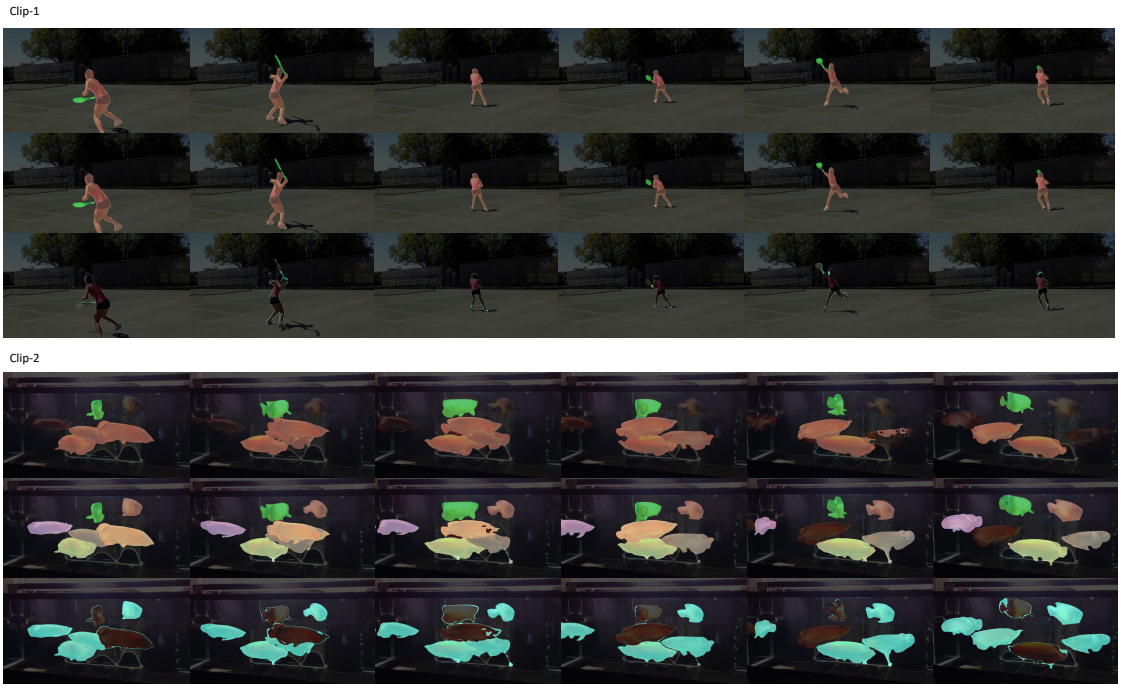

Figure 1: Visual Comparison Results from Tube-Link with ResNet50 backbone. Our method (middle) achieves consistent segmentation and better segmentation/tracking results than the Mask2Former-VIS baseline (top). We also visualize the difference maps (bottom). **Best viewed by zooming in.**

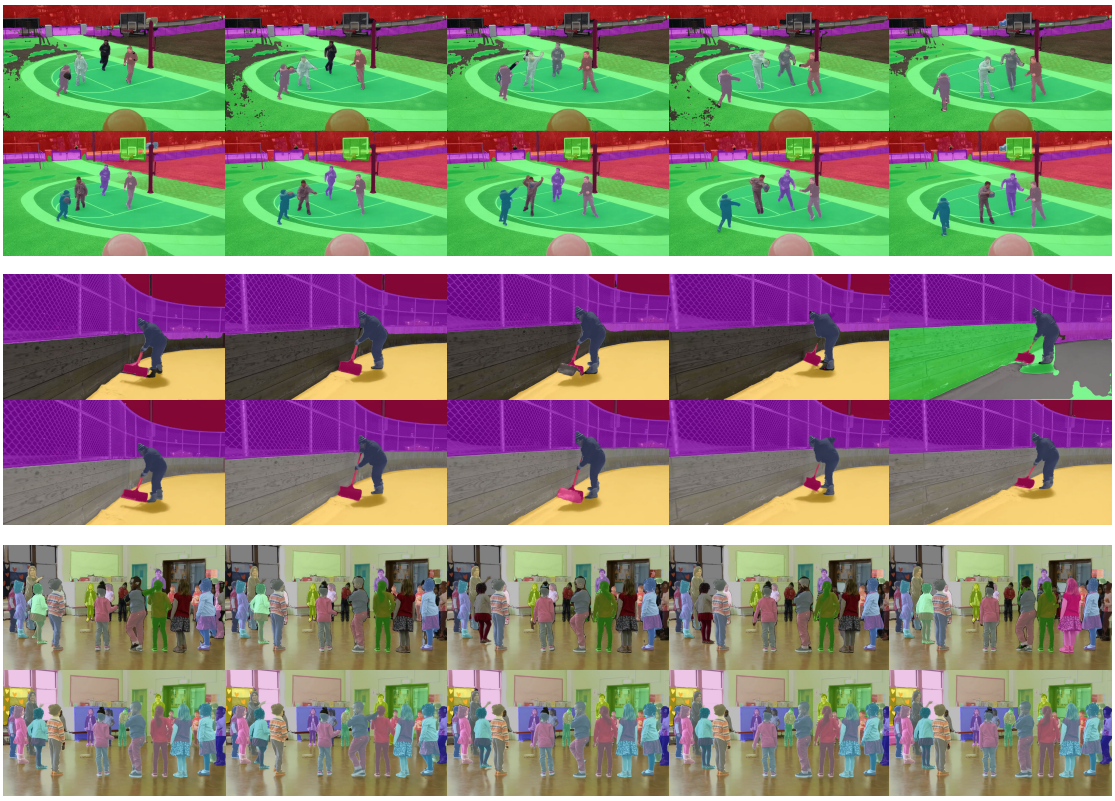

Figure 2: More Visual Results from Tube-Link with ResNet50 backbone. Our method (top) achieves consistent segmentation and better tracking results than the Video K-Net+ baseline (bottom). **Best viewed by zooming in.**

Table 7: **Results on the OVIS datasets.** We report the mAP metric. † adopts COCO video pseudo labels. Axial means using the extra Axial Attention [22]. Our method does not apply these techniques for simplicity.

Method	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
CrossVIS [29]	14.9	32.7	12.1	10.3	19.8
VISOLO [7]	15.3	31.0	13.8	11.1	21.7
TeViT [30]	17.4	34.9	15.0	11.2	21.8
VITA [10]	19.6	41.2	17.4	11.7	26.0
DeVIS [1]	23.8	48.0	20.8	-	-
Min-VIS [11]	25.0	45.5	24.0	13.9	29.7
IDOL [26]	30.2	51.3	30.0	15.0	37.5
VITA [10] †	19.6	41.2	17.4	11.7	26.0
Tube-Link	29.5	51.5	30.2	15.5	34.5

Table 8: More Experiment Results.

(a) More results on ViP-Seg dataset.

Method	STQ	SQ	AQ
Video K-Net	33.1	35.0	29.6
Video K-Net + tube matching (Ours)	34.7	36.8	30.8
Video K-Net + tube matching (IFC)	33.3	35.7	28.4

(b) KITTI-STEP test set results.

Method	Backbone	STQ
Motion-Deeplab	ResNe50	0.52
Video K-Net	ResNet50	0.59
Video K-Net	ResNet50	0.63
Tube-Link	ResNet50	0.60
Tube-Link	Swin-base	0.65

Table 9: Effect Of Quasi-Dense Tracker (Results on Youtube-VIS-2019 validation set).

Method	Naive Tracker	Quani-Dense Tracker	mAP
MinVIS	✓	-	47.4
MinVIS	-	✓	48.0 (+0.6)
MiniVIS + tube matching	✓	-	48.8 (+1.4)
Tube-Link	✓	-	52.6 (-0.2)
Tube-Link	-	✓	52.8

References

[1] Adrià Caelles, Tim Meinhardt, Guillem Brasó, and Laura Leal-Taixé. Devis: Making deformable transformers work for video instance segmentation. *arXiv preprint arXiv:2207.11103*, 2022. 5

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[3] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv pre-*

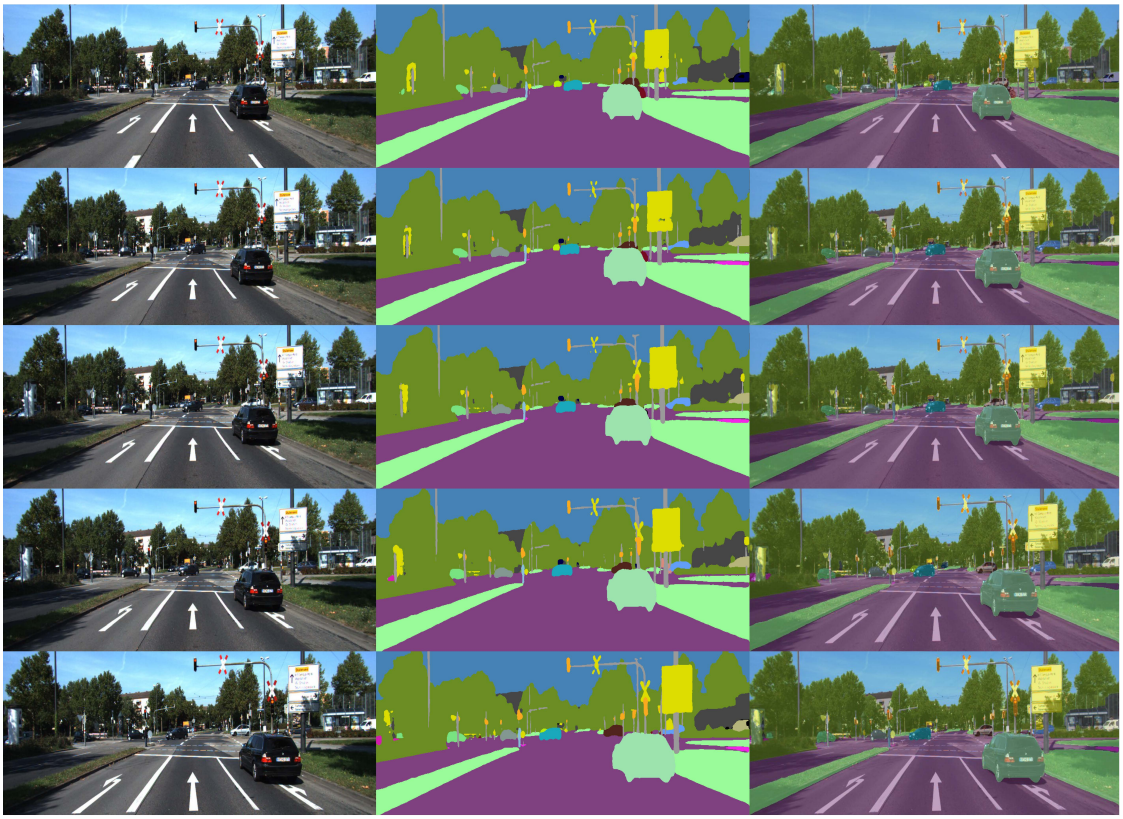

Figure 3: Visual Results from Tube-Link with ResNet50 backbone on the KITTI-STEP dataset. **Best viewed by zooming in.**

Figure 4: Visual Results on Failure Cases of Tube-Link. (a), Remote objects lead to ID switches and inferior segmentation results. (b), Heavy occlusion leads to an ID switch. (c), Segmentation consistency problems caused by camera motion.

print, 2021. 2, 3, 4

[4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1

[5] 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] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *CVPR*, 2021. 2

[7] Su Ho Han, Sukjun Hwang, Seoung Wug Oh, Yeonchool Park, Hyunwoo Kim, Min-Jung Kim, and Seon Joo Kim. Visolo: Grid-based space-time aggregation for efficient online video instance segmentation. In *CVPR*, 2022. 4, 5

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[9] Miran Heo, Sukjun Hwang, Jeongseok Hyun, Hanjung Kim, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. A generalized framework for video instance segmentation. In *CVPR*, 2023. 1, 2, 4

[10] Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. Vita: Video instance segmentation via object token association. In *NeurIPS*, 2022. 2, 4, 5

[11] De-An Huang, Zhiding Yu, and Anima Anandkumar. Minvis: A minimal video instance segmentation framework without video-based training. In *NeurIPS*, 2022. 2, 4, 5

[12] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. Video instance segmentation using inter-frame communication transformers. *NeurIPS*, 2021. 2, 4

[13] Dahun Kim, Jun Xie, Huiyu Wang, Siyuan Qiao, Qihang Yu, Hong-Seok Kim, Hartwig Adam, In So Kweon, and Liang-Chieh Chen. Tubeformer-deeplab: Video mask transformer. In *CVPR*, 2022. 2, 4

[14] 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. 1, 2, 3

[15] 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. 2

[16] 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. *ICCV*, 2021. 2

[17] Jiaxu 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. 1, 3

[18] Jiaxu 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. 3

[19] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *IJCV*, 2022. 3

[20] Guolei Sun, Yun Liu, Henghui Ding, Thomas Probst, and Luc Van Gool. Coarse-to-fine feature mining for video semantic segmentation. In *CVPR*, 2022. 2

[21] Guolei Sun, Yun Liu, Hao Tang, Ajad Chhatkuli, Le Zhang, and Luc Van Gool. Mining relations among cross-frame affinities for video semantic segmentation. *ECCV*, 2022. 2

[22] 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. 4, 5

[23] 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. 4

[24] M. Weber, J. Xie, M. Collins, Yukun Zhu, P. Voigtlaender, H. Adam, B. Green, A. Geiger, B. Leibe, D. Cremers, Aljosa Osep, L. Leal-Taixé, and Liang-Chieh Chen. Step: Segmenting and tracking every pixel. *NeurIPS*, 2021. 1, 2

[25] Junfeng Wu, Yi Jiang, Song Bai, Wenqing Zhang, and Xiang Bai. Seqformer: Sequential transformer for video instance segmentation. In *ECCV*, 2022. 1, 2, 4

[26] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In defense of online models for video instance segmentation. In *ECCV*, 2022. 2, 4, 5

[27] Jialian Wu, Sudhir Yarram, Hui Liang, Tian Lan, Junsong Yuan, Jayan Eledath, and Gerard Medioni. Efficient video instance segmentation via tracklet query and proposal. In *CVPR*, 2022. 2, 4

[28] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 3

[29] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learning for fast online video instance segmentation. In *ICCV*, 2021. 2, 4, 5

[30] Shusheng Yang, Xinggang Wang, Yu Li, Yuxin Fang, Jiemin Fang, Wenyu Liu, Xun Zhao, and Ying Shan. Temporally efficient vision transformer for video instance segmentation. In *CVPR*, 2022. 5

[31] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. In *NeurIPS*, 2021. 1, 2

[32] Yi Zhou, Hui Zhang, Hana Lee, Shuyang Sun, Pingjun Li, Yangguang Zhu, ByungIn Yoo, Xiaojuan Qi, and Jae-Joon Han. Slot-vps: Object-centric representation learning for video panoptic segmentation. In *CVPR*, 2022. 1, 2