

## Video K-Net: A Simple, Strong, and Unified Baseline for Video Segmentation Appendix

In this appendix, we provide the following information in addition to the main paper: more experimental details and more visualization results on Cityscapes-VPS [3], KITTI-STEP [10] VSPW [6] and YouTubVIS [11]. All the models use AdamW for training.

### 1. More Experimental Details

#### Detailed pretraining setting.

For COCO [4] dataset pretraining, all the models are trained following original K-Net settings [12]. We adopt the multi-scale training as previous work [1] by resizing the input images such that the shortest side is at least 480 and at most 800 pixels while the longest at most 1333. We also apply random crop augmentations during training where the train images are cropped with probability 0.5 to a random rectangular patch which is then resized again to 800-1333. All the models are trained for 36 epochs.

For Mapillary [7] dataset pretraining, we mainly follow the Panoptic-Deeplab setting [2]. We adopt the multi-scale training where the the scale ranges from 1.0 to 2.0 of origin images size, then we apply a random crop of  $1024 \times 2048$  patches. The horizontal flip is applied. The pretraining process takes 240 epochs. The Mapillary pretraining is for fair comparison, since the ViP-Deeplab [8] all use the Mapillary pretraining for better results. Note that, we *only* pretrain our largest models with Swin-base [5] as backbone for fair comparison with previous work.

#### Training and inference on Cityscapes-VPS.

For Cityscapes-VPS training, we follow previous VP-SNet [3] that we randomly sample one frame from near one frames as the reference frame. We adopt the multi-scale training where the scale ranges from 1.0 to 2.0 of origin images size then we apply a random crop of  $1024 \times 2048$  patches. For the Swin-base model, we apply a random crop of  $800 \times 1600$  patches to save memory and computation cost. The total training epoch is set to 8.

During the inference, the previous frame play as the reference frame, the kernel information is directly propagated into the next frame in an online manner.

#### Training and inference on KITTI-STEP.

For KITTI-STEP training, we follow previous Motion-Deeplab [10] that we randomly sample *three* frames from near one frames as the reference frame. We adopt the multi-

scale training where the scale ranges from 1.0 to 2.0 of origin images size then we 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 [10], we also use Cityscapes pretraining before training on STEP which leads to about 3% STQ gain on the K-Net baseline.

#### Training and inference on VSPW.

For VSPW dataset, we adopt the same setting on training K-Net [12] on ADE-datasets [13]. We randomly sample *three* frames from near one frames as the reference frame. The inference procedure is the same as Cityscapes-VPS dataset.

#### Training and inference on Youtube-VIS.

For Youtube-VIS, we use the COCO pretrained K-Net model and train the entire model via randomly sampling 5 frames in a clip-wised manner. During the inference, we pad the entire the clip to 16 [9] and directly output to each instance id via the kernel indexes.

### 2. More Visualization Results

**More Visualization on Cityscapes-VPS.** In Fig. 1, we present more visual examples on Cityscapes VPS dataset. Compared with Ground Truth, our method can segment and track well for each pixel for various scale object inputs. We use the Swin-base model for visualization.

**More Visualization on KITTI-STEP.** In Fig. 2, we give more visual results on the KITTI-STEP validation set. On both clips, Video-KNet shows convincing results.

**Failure Cases Analysis.** In Fig. 3, we present three failure cases of our Video K-Net. The first case is the tracking failure case where the car moves fast in remote scene and then there is an id switch. We believe adding motion cues will improve this case. The last two cases show the segmentation errors. The first is because the color of cars' window is similar to the ground. The second is caused by less training cases: bike on the truck, which makes network hard to predict.

**Border Impact.** Our work pushes the boundary of video segmentation algorithms through simplicity and effectiveness. Since most applications are the video input, this work could also ease and accelerate the model production in real-world applications, such as in autonomous driving. Due to

limited dataset size, we do not evaluate the robustness of the proposed method on corrupted video inputs.

### 3. Demo Video

We also append several video demos on both Cityscapes-VPS and KITTI-STEP. Note that we provide ground truth in Cityscapes-VPS where the black area is the ignored region. For KITTI-STEP, we visualize the network outputs and original video clips.

### References

- [1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1
- [2] 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. 1
- [3] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *CVPR*, 2020. 1
- [4] 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. 1
- [5] 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. 1
- [6] 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. 1
- [7] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Buló, and Peter Kotschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 1
- [8] Siyuan Qiao, Yukun Zhu, H. Adam, A. Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. *CVPR*, 2021. 1
- [9] 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. 1
- [10] 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. *NIPS*, 2021. 1
- [11] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1
- [12] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *CoRR*, abs/2106.14855, 2021. 1
- [13] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *arXiv preprint arXiv:1608.05442*, 2016. 1

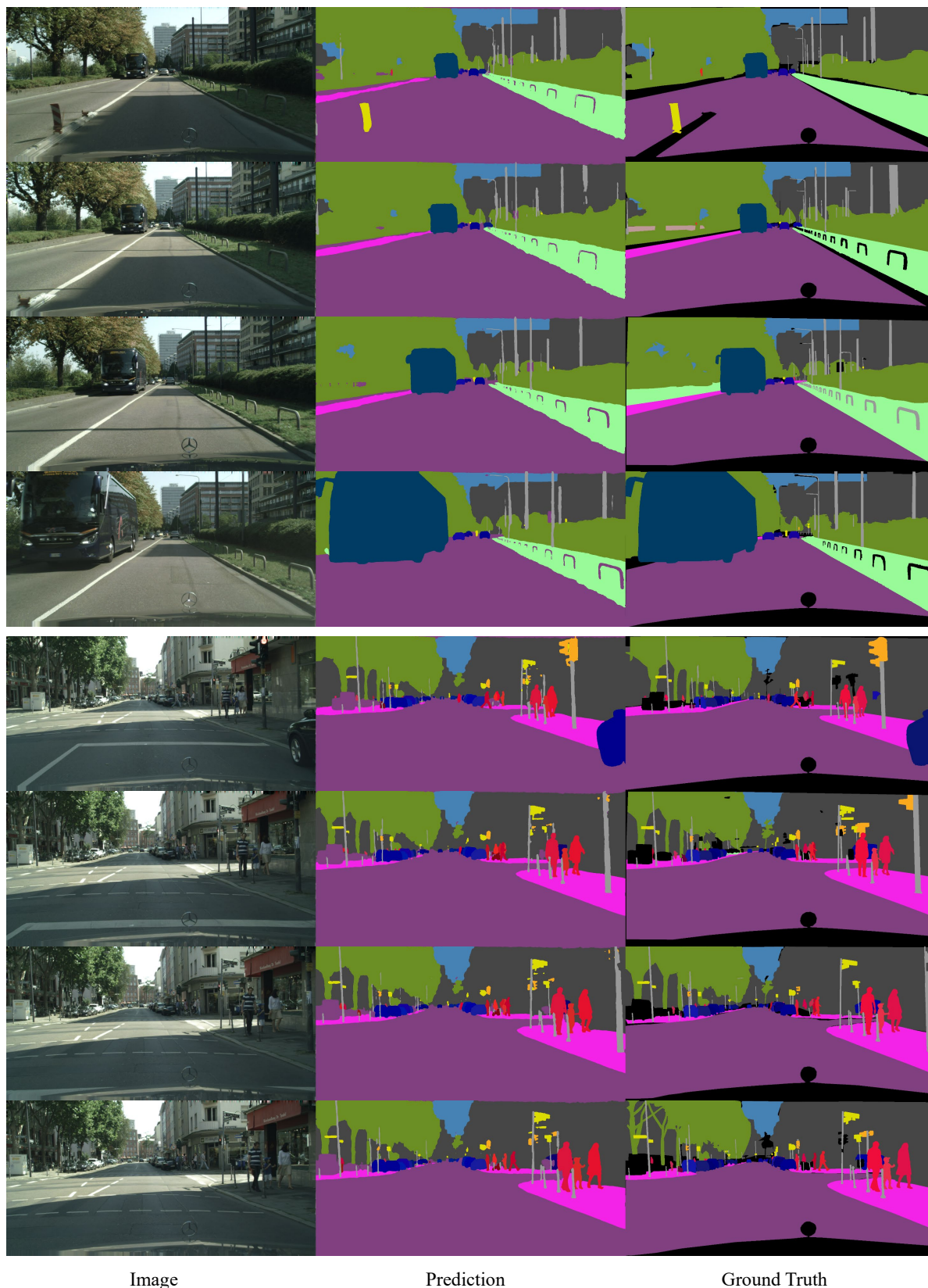


Figure 1. More visual examples of our Video K-Net on CityScapes-VPS dataset. The same instances are shown with the same color. Best view it on screen.



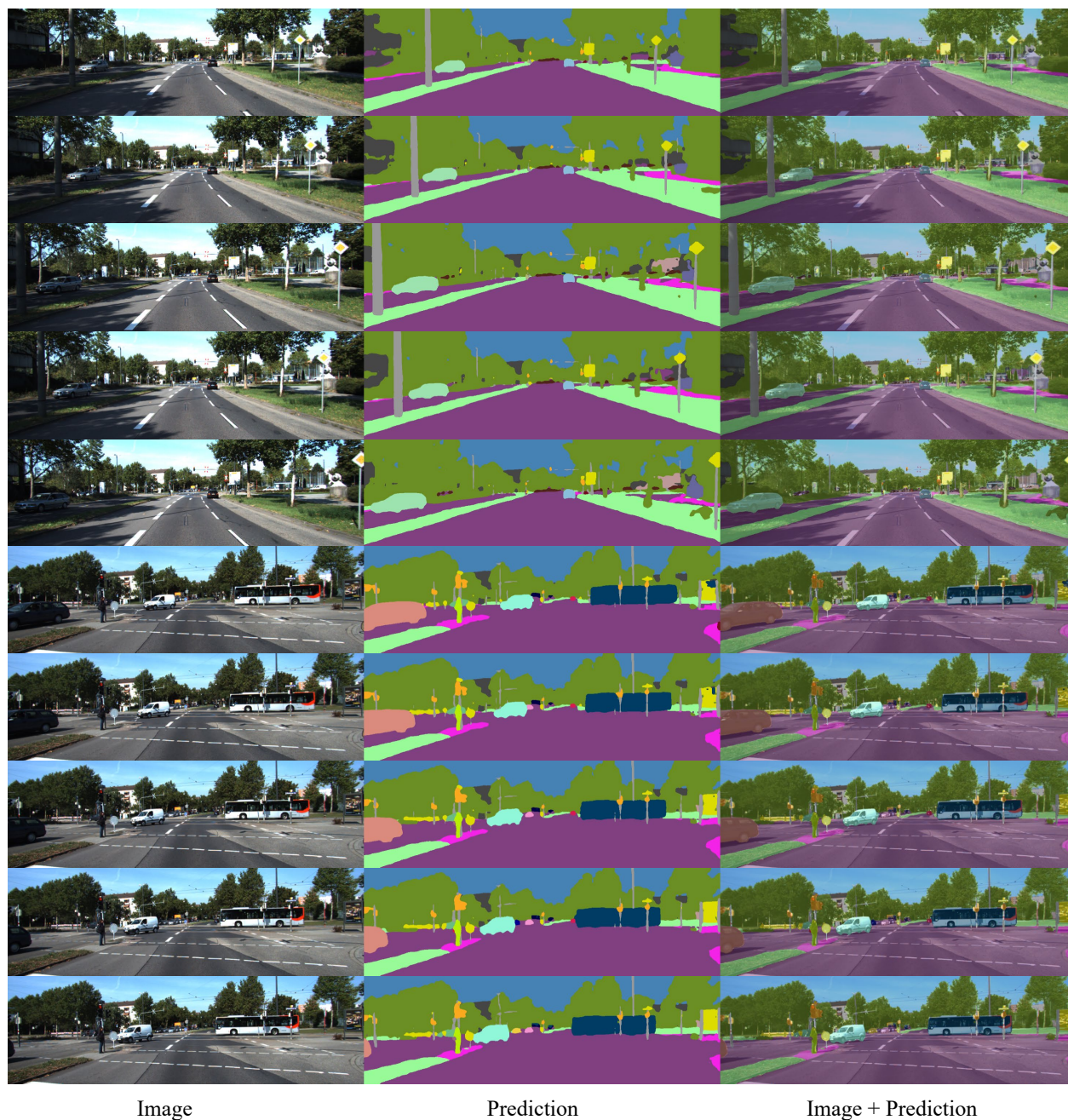


Figure 2. More visual examples of our Video K-Net on STEP dataset. The same instances are shown with the same color. Best view it on screen.

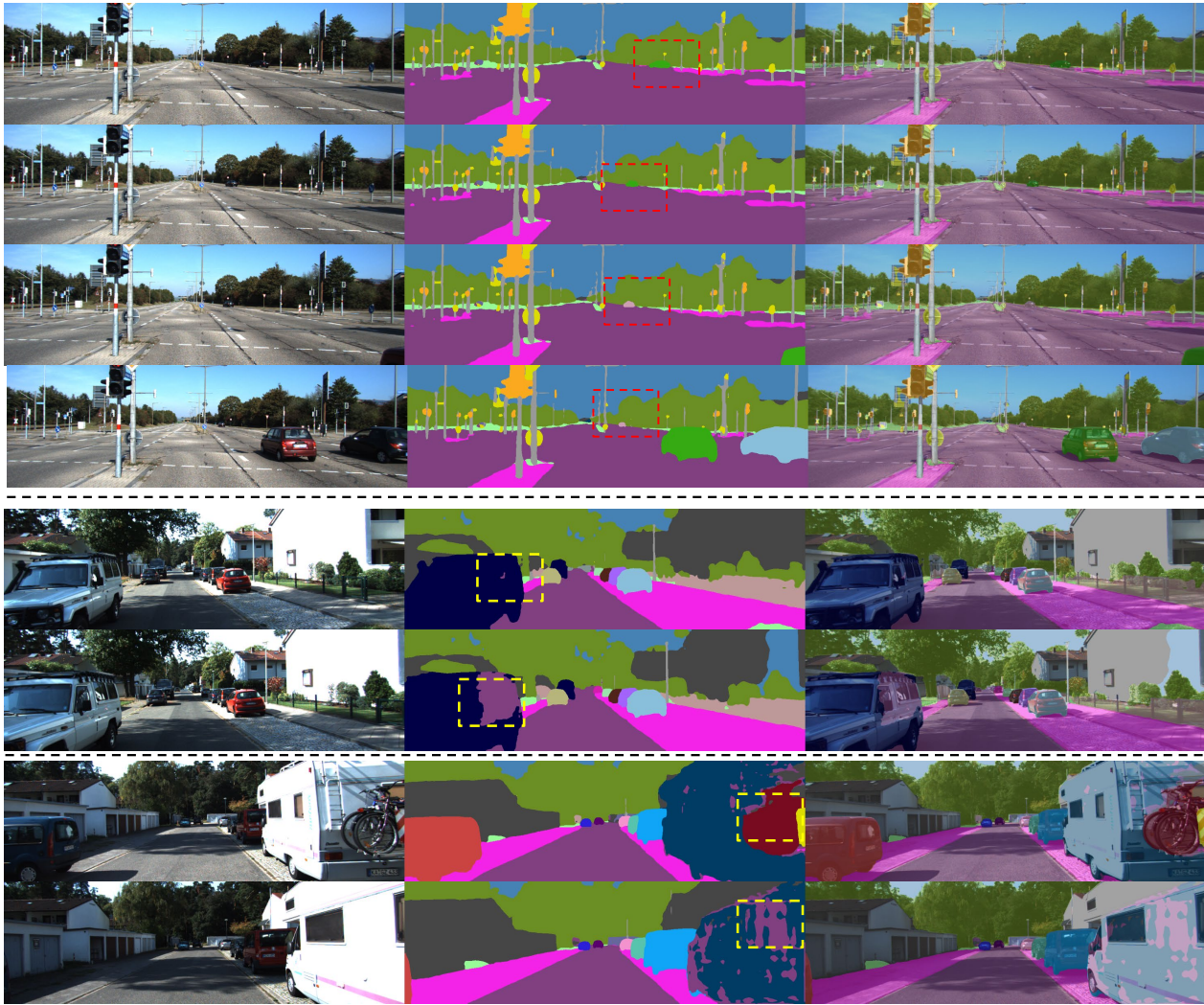


Figure 3. Failure cases of Video K-Net on STEP dataset. The same instances are shown with the same color. The red boxes show the tracking errors while the yellow boxes show the segmentation errors. Best view it on screen.