What is Point Supervision Worth in Video Instance Segmentation?
Supplementary Material

Shuaiyi Huang[1]*, De-An Huang[2], Zhiding Yu[2], Shiyi Lan[2], Subhashree Radhakrishnan[2],
Jose M. Alvarez[2], Abhinav Shrivastava[1], Anima Anandkumar[3]*
[1]University of Maryland, College Park [2]NVIDIA [3]Caltech

In this supplementary material, we provide more detailed quantitative analysis and qualitative results of our method as follows: i) We provide implementation details of distance transform when sampling points in Sec. A; ii) Apart from the ablation studies provided in the main paper, we further provide the ablation analysis of the point selection bias with additional point sampling methods in Sec. B, the analysis of pretrained models in Sec. C, the analysis of generality in Sec. D, the upperbound of more points in Sec. E; iii) Finally, we provide more quantitative and qualitative results on YouTube-VIS 2019 [7], YouTube-VIS 2021 [7], and OVIS [3] validation set in Sec. F and Sec. G. We will release our code upon acceptance.

A. Implementation Details of Sampling Points by Distance Transform

We provide implementation details of distance transform for sampling positive and negative points, respectively. Distance transform is an operator applied to binary images. The resulting distance map is a gray level image that looks similar to the input binary image, except that the gray level intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point.

We first explain how we sample positive points via distance transform in detail. Given a binary ground truth object mask, we generate its distance map by applying distance transform directly to the foreground mask. The resulting distance map has zero values outside the foreground region and positive values inside the foreground region, indicating the euclidean distance to the closest boundary for each point. We then normalize the distance values to obtain the distribution used for sampling points, and randomly sample positive points given the resulting distribution.

We sample negative points via distance transform in a similar way, as shown in Figure 1. We first apply distance transform to the inverse of the foreground mask (*i.e.* background region has value one while the foreground region has value zero). We then threshold the resulting distance map with a fixed pixel values (*e.g.* 50) and only keep the pixels whose distance is within the distance threshold. Finally, we sample negative points randomly inside the kept pixels. By setting different threshold, we can control how far away the negative points are sampled.

*Work done during internship / affiliation with NVIDIA.

B. More Ablation on Point Selection Bias

To further investigate the point selection bias, we report ablation results with additional negative point sampling methods in Table 2. We additionally sample negative points by randomly sampling from the region outside the ground truth mask (Random (Out-mask)) or by negative distance transform (Random (Distance Transform)). We observe that different negative points sampling methods achieve comparable results. This result show that our method is generally robust to the negative point location, thanks to our point-based matcher that incorporates annotation-free negative cues.

C. More Analysis of Pretrained models.

As we focus on reducing video annotations, we therefore follow the existing work [1, 4, 8] that use pretrained instance segmentation models. To further study the impact of pretrained models, we additionally report PointVIS (P1) re-

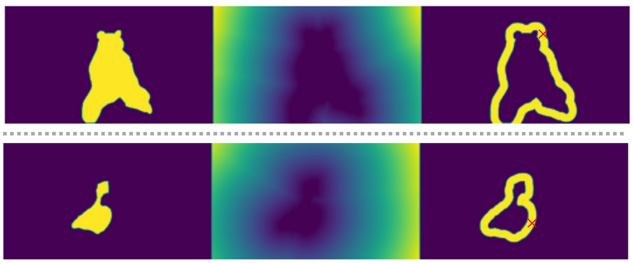

Figure 1. **Visualization of sampling negative points with distance transform.** From left to right is the ground truth foreground mask, heatmap of the distance transform of the ground truth background mask (the lighter the larger distance), region left for negative points sampling after thresholding a distance of 50 pixels. Red cross indicates the position of the sampled negative points.

Methods	person	panda	lizard	parrot	skateboard	sedan	ape	dog	snake	monkey	hand	rabbit	duck	cat	cow	fish	train	horse	turtle	bear	**mAP**
MinVIS [1]	53.3	71.8	83.5	79.6	21.0	61.5	47.2	55.6	47.0	35.4	59.8	93.4	54.9	77.3	66.1	24.0	58.7	47.1	57.3	62.3	-
PointVIS	47.7	56.6	55.6	68.2	11.9	58.5	38.2	50.0	37.3	23.8	44.7	89.3	48.2	58.4	68.4	2.3	66.7	47.5	57.3	63.3	-

Methods	motorbike	giraffe	leopard	fox	deer	owl	surfboard	airplane	truck	zebra	tiger	elephant	snowboard	boat	shark	mouse	frog	eagle	seal	tennis racket	**mAP**
MinVIS [1]	40.9	70.5	59.0	41.9	65.6	57.2	8.4	37.2	72.4	72.2	57.6	70.6	0.2	69.0	58.5	60.4	3.7	84.4	56.4	37.1	55.3
PointVIS	36.7	66.2	35.0	39.2	55.3	48.7	0.0	44.7	64.6	55.9	48.8	58.5	5.5	56.6	23.9	35.7	12.2	72.2	51.4	33.7	46.0

Table 1. **Per-class and overall mAP results for MinVIS [1] and PointVIS (P1, w/o self-training) on Youtube-VIS 2019 val-dev**. All models here use Swin-B as the backbone.

Model ID	Sampling method for Pos	Sampling method for Neg	AP (%)
PointVIS (P1)	Random	-	46.0
PointVIS (P1)	Distance Transform	-	47.1
PointVIS (P1N1)	Random	Random (In-box)	48.6
PointVIS (P1N1)	Random	Random (Out-box-but-in-200%-box)	48.0
PointVIS (P1N1)	Random	Random (Out-mask)	48.5
PointVIS (P1N1)	Random	Random (Distance Transform)	48.8

Table 2. **Analysis of point selection bias on YouTube-VIS 2019 [7] val-dev.**

Image Pretraining	Video Finetuning	mAP %
R101	Swin-B	42.5
Swin-B	Swin-B	46.0

Table 3. **Pretraining with Different models.**

COCO Frames 1%	5%	10%	100%
mAP %	41.2	43.0	45.3	46.0

Table 4. **Pretraining with different number of COCO images.**

Model	Sup.	mAP (seen)	mAP (unseen)	mAP (all)
MinVIS [1]	$\mathcal{M}$	51.6%	58.0%	55.3%
PointVIS (ours)	$\mathcal{P}_1$	47.1%	44.9%	46.0%

Table 5. **Unseen categories evaluation on YouTube-VIS 2019 val-dev.**

Model	Matching	mAP
MinVIS [1]	/	55.3%
PointVIS (ours)	P10N10	49.5%
PointVIS (upperbound)	GT Mask	50.0%

Table 6. **More points oracle on YouTube-VIS 2019 val-dev.**

sults on YouTube-VIS 2019 val-dev by using different pretrained backbones to generate pseudo-labels while using the same backbone to finetune videos as shown in Table 3, and pretraining on COCO with less frames as shown in Table 4. PointVIS achieves competitive results with varying quality of pretrained models.

D. More Analysis of Generality

To validate the generality, we additionally report mAP of 19 seen and 21 unseen categories (Table 5) on YouTube-VIS 2019 val-dev. PointVIS achieves good results on unseen categories with point labels. We also report per-category results in Table 1 for reference.

E. Upperbound of More Points

Treating the ground truth mask as a set of points, we implemented an unpperbound model of our PointVIS (Table 6). The upperbound performance (50.0% mAP) does not match the fully-supervised counterpart (55.3% mAP), as it is bounded by proposals quality. Our PointVIS instead could approach this upperbound with very little point supervision (saturated at 49.5% mAP w/ P10N10).

F. More Quantitative Results

To have a better understanding of our method, we additionally report quantitative performance of our PointVIS w/o self-training on Youtube-VIS 2019 [7], Youtube-VIS 2021 [7] and OVIS [3] validation set for reference, as summarized in Table 7. High retention rate across three benchmarks indicates the effectiveness of our method.

G. More Qualitative Results

More qualitative results from the predictions of our PointVIS on Youtube-VIS 2019 [7], Youtube-VIS 2021 [7] and OVIS [3] validation set, are shown in Figure 2, 3 and 4, respectively.

References

[1] De-An Huang, Zhiding Yu, and Anima Anandkumar. Minvis: A minimal video instance segmentation framework without video-based training. *arXiv preprint arXiv:2208.02245*, 2022. 1, 2, 3

[2] Zhuang Li, Leilei Cao, and Hongbin Wang. Limited sampling reference frame for masktrack r-cnn. In *ICCVW*, 2021. 3

[3] 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. *arXiv preprint arXiv:2102.01558*, 2021. 1, 2, 3, 5

[4] 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

[5] Junfeng Wu, Yi Jiang, Wenqing Zhang, Xiang Bai, and Song Bai. Seqformer: a frustratingly simple model for video instance segmentation. *arXiv preprint arXiv:2112.08275*, 2021. 3

[6] 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. 3

Method	Dataset	Sup.	AP (%)	AP_{50} (%)	AP_{75} (%)	AR_1 (%)	AR_{10} (%)
TeViT [9]	YouTube-VIS-2019	$\mathcal{M}$	56.8	80.6	63.1	52.0	63.3
IDOL [6]	YouTube-VIS-2019	$\mathcal{M}$	64.3	87.5	71.0	55.6	69.1
MinVIS [1]	YouTube-VIS-2019	$\mathcal{M}$	61.6	83.3	68.6	54.8	66.6
PointVIS (P1)*	YouTube-VIS-2019	$\mathcal{P}_1$	52.5 (85.2%)	74.5 (89.4%)	59.2 (86.3%)	47.2 (86.1%)	61.5 (92.3%)
PointVIS (P1)	YouTube-VIS-2019	$\mathcal{P}_1$	53.9 (87.5%)	75.7 (90.9%)	61.8 (90.1%)	47.5 (86.7%)	61.4 (92.2%)
PointVIS (P1N1)*	Swin-L	$\mathcal{P}_2$	57.6 (93.5%)	79.9 (95.9%)	63.9 (93.1%)	52.2 (95.2%)	62.7 (94.1%)
PointVIS (P1N1)	YouTube-VIS-2019	$\mathcal{P}_2$	59.6 (96.7%)	83.3 (100%)	67.1 (97.8%)	52.7 (96.2%)	63.8 (95.8%)
SeqFormer [5]	YouTube-VIS-2021	$\mathcal{M}$	51.8	74.6	58.2	42.8	58.1
IDOL [6]	YouTube-VIS-2021	$\mathcal{M}$	56.1	80.8	63.5	45.0	60.1
MinVIS [1]	YouTube-VIS-2021	$\mathcal{M}$	55.3	76.6	62.0	45.9	60.8
PointVIS (P1)*	YouTube-VIS-2021	$\mathcal{P}_1$	46.0 (83.2%)	70.3 (91.8%)	50.1 (80.8%)	39.2 (85.4%)	52.9 (87.0%)
PointVIS (P1)	YouTube-VIS-2021	$\mathcal{P}_1$	46.3 (83.7%)	70.5 (92.0%)	51.1 (82.4%)	37.7 (82.1%)	52.9 (87.0%)
PointVIS (P1N1)*	YouTube-VIS-2021	$\mathcal{P}_2$	47.6 (86.1%)	72.2 (94.2%)	53.0 (85.5%)	40.7 (88.7%)	53.9 (88.7%)
PointVIS (P1N1)	YouTube-VIS-2021	$\mathcal{P}_2$	48.5 (87.7%)	73.0 (95.3%)	54.4 (87.7%)	41.7 (90.8%)	54.1 (89.0%)
MaskTrack [2]	Occluded VIS	$\mathcal{M}$	28.9	56.3	26.8	13.5	34.0
IDOL [6]	Occluded VIS	$\mathcal{M}$	42.6	65.7	45.2	17.9	49.6
MinVIS [1]	Occluded VIS	$\mathcal{M}$	39.4	61.5	41.3	18.1	43.3
PointVIS (P1)*	Occluded VIS	$\mathcal{P}_1$	27.0 (68.5%)	48.5 (78.9%)	25.2 (61.0%)	13.8 (76.2%)	32.1 (74.1%)
PointVIS (P1)	Occluded VIS	$\mathcal{P}_1$	28.6 (72.6%)	49.6 (80.7%)	27.5 (66.6%)	15.0 (82.9%)	32.8 (75.8%)
PointVIS (P1N1)*	Occluded VIS	$\mathcal{P}_2$	27.4 (69.5%)	48.7 (79.2%)	25.5 (61.7%)	13.9 (76.8%)	31.5 (72.7%)
PointVIS (P1N1)	Occluded VIS	$\mathcal{P}_2$	28.6 (72.6%)	51.2 (83.3%)	27.2 (65.9%)	14.7 (81.2%)	32.2 (74.4%)

Table 7. **Full mask ($\mathcal{M}$) vs. our point supervision ($\mathcal{P}$) on validation set of YouTube-VIS 2019 [7], YouTube-VIS 2021 [7], and OVIS [3].** All results below are based on Swin-L backbone. * denotes our PointVIS results w/o self-training.

Figure 2. Visualization of predictions from our PointVIS on Youtube-VIS 2019 [7] validation set.

[7] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1, 2, 3, 4

[8] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learn-

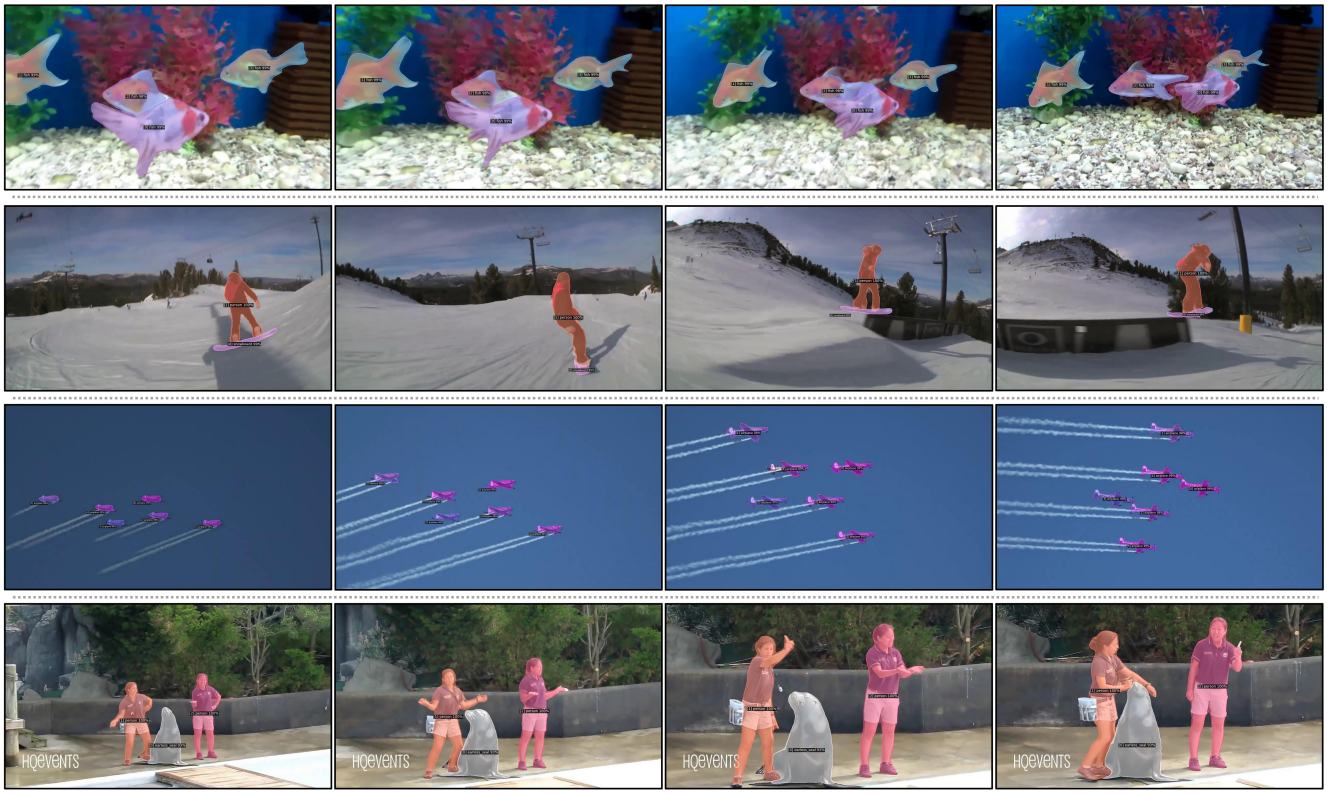

Figure 3. Visualization of predictions from our PointVIS on Youtube-VIS 2021 [7] validation set.

ing for fast online video instance segmentation. In *ICCV*, 2021. 1

[9] Shusheng Yang, Xinggang Wang, Yu Li, Yuxin Fang, Jiemin Fang, Liu, Xun Zhao, and Ying Shan. Temporally efficient vision transformer for video instance segmentation. In *CVPR*, 2022. 3

Figure 4. Visualization of predictions from our PointVIS on OVIS [3] validation set.