Supplementary Material
UniFS: Universal Few-shot Instance Perception with Point Representations

Sheng Jin[1,4⋆], Ruijie Yao[2,4⋆], Lumin Xu[3], Wentao Liu[4] ✉, Chen Qian[4], Ji Wu[2], and Ping Luo[1,5] ✉

[1] The University of Hong Kong [2] Tsinghua University [3] The Chinese University of Hong Kong [4] SenseTime Research and Tetras.AI [5] Shanghai AI Laboratory
js20@connect.hku.hk, yrj21@mails.tsinghua.edu.cn

S1 Effect of Point Number

We evaluate the performance of UniFS with different numbers of points to represent the instance perception tasks, as depicted in Table S1. Initially, we explore the point numbers as 4, 8, and 16 for object detection. It exhibites superior results when applied 16 points, showcasing notably enhanced performance in 5-shot scenario and comparable performance in 1-shot scenario. Therefore, we adopt 16 points for the task of object detection. Subsequently, we conduct experiments on instance segmentation with point numbers as 16, 32, and 64. It demonstrates performance improvement with increased point number on the task of instance segmentation, while introduces a detrimental effect on detection performance. To achieve a trade-off between detection and segmentation, we select to employ 32 points for instance segmentation with decent perforamce on both tasks.

S2 Effect of Transformer Blocks

We also analyze the influence of the number of transformer blocks applied for the point decoder in UniFS. As shown in Table S2, we compare the performance

Table S1: Effect of point number on the COCO-UniFS `val` set. The row with blue color indicates our choice for UniFS.

Det. Point	Seg. Point	Det. AP ↑		Seg. AP ↑	
		K=1	K=5	K=1	K=5
4	-	12.4	17.3	✗	✗
8	-	12.2	16.7	✗	✗
16	-	12.2	17.9	✗	✗
16	16	12.9	16.5	8.6	10.7
16	32	12.6	17.6	8.5	11.2
16	64	12.0	17.0	9.0	11.4

Table S2: Effect of transformer block number on the COCO-UniFS `val` set. The row with blue color indicates our choice for UniFS.

Number	Det. AP ↑		Seg. AP ↑		Kpt. AP ↑		Cnt. MSE ↓	
	K=1	K=5	K=1	K=5	K=1	K=5	K=1	K=5
1	12.1	17.4	8.0	11.0	9.6	20.6	1.38	1.31
2	12.7	18.2	8.6	11.5	12.2	22.1	1.38	1.32
4	11.9	17.2	8.0	11.0	10.9	20.7	1.39	1.29
6	11.8	16.9	8.2	10.9	12.6	21.6	1.38	1.31

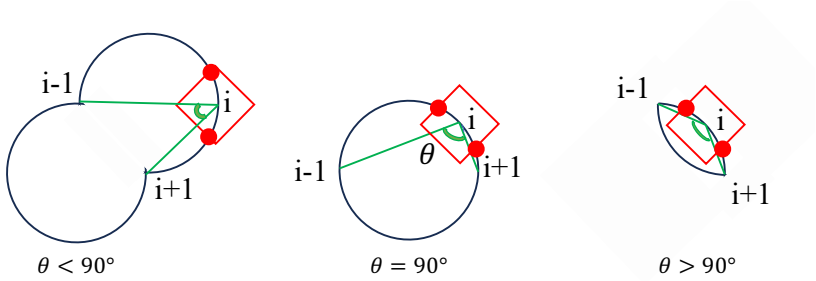

Fig. S1: Analysis on 1-hop SAPL. For 1-hop SAPL, the trajectory of point P, where the angle between a moving point P and two fixed points is a fixed value θ, is a closed curve composed of two symmetrical arcs: the spindle shape ($\theta < 90°$), a circle ($\theta = 90°$), and the lens shape ($\theta > 90°$).

of all the tasks with different block numbers. When the number is set to 2, the best results are achieved across most of the tasks. Considering computational cost and model performance, we opt for 2 blocks for point decoder.

S3 Analysis on Structure-Aware Point Learning

In Fig. 3 of the main text , we show that only using L1/L2 loss will introduce ambiguity, as different model predictions have the same loss values (points on the circle for L2, and the diamond for L1). As shown in Fig S1, for 1-hop SAPL, the trajectory of point P, where the angle between a moving point P and two fixed points is a fixed value θ, is a closed curve composed of two symmetrical arcs: the spindle shape ($\theta < 90°$), a circle ($\theta = 90°$), and the lens shape ($\theta > 90°$). Considering both L1 and SAPL, the same loss value occurs at the intersection of their trajectories (two red points as shown below), significantly mitigating the ambiguity.

Table 5 in the main text demonstrates a significant improvement on the tasks of object detection and instance segmentation by introducing Structure-Aware Point Learning (SAPL). We further provide some visualization results to visually illustrate the effectiveness of our proposed SAPL. As shown in Fig S2, it is obvious that UniFS with SAPL effectively learns the object stuctures and

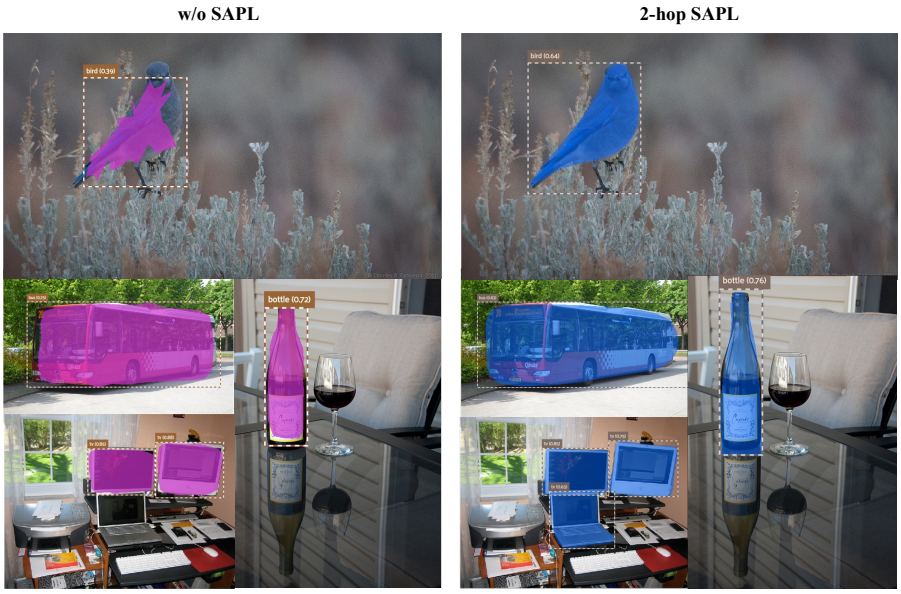

Fig. S2: Qualitative effect of SAPL on the tasks of object detection and instance segmentation. Our proposed UniFS ("2-hop SAPL") significantly outperforms the model only using L1 loss ("w/o SAPL").

produces reliable contours, while the predictions of model trained with L1 loss only ("w/o SAPL") are inaccurate. For example, in the topmost image, the model without SAPL cannot capture the relationship between points when supervising each point individually, leading to unsmooth boundary and incorrect bounding box.