Unifying Visual Perception by Dispersible Points Learning

Jianming Liang[1,2⋆], Guanglu Song[2], Biao Leng[1], and Yu Liu[2✉]

[1] School of Computer Science and Engineering, Beihang University
[2] SenseTime Research
ljmmm1997@gmail.com, songguanglu@sensetime.com,
lengbiao@buaa.edu.cn, liuyuisanai@gmail.com

A Detailed Architecture of UniHead

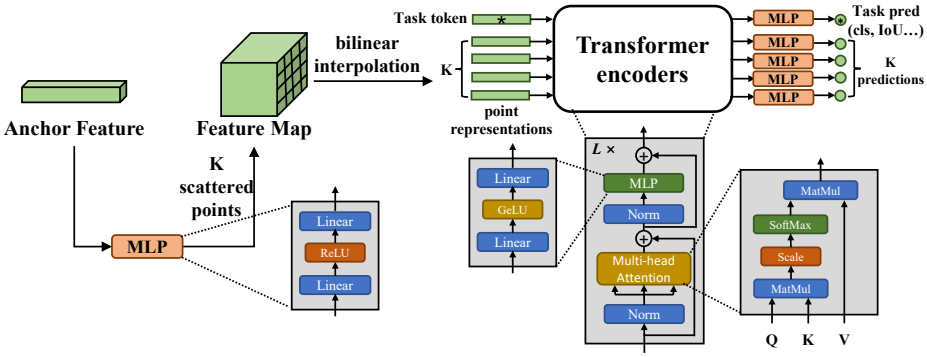

Fig. 1: An illustration of `UniHead` detailed architecture.

A detailed illustration is given in Fig.1. First, K scattered points are generated through an multi-layer perceptron and we use bilinear interpolation to extract point representations. Then, together with a learnable embedding vector (optional), they are fed into transformer encoders. Finally, we use different MLPs to get the final results, where "task pred" means predictions from the task token, like IoU token.

B Discussions on UniHead Unification

The unifying concept in `UniHead` indicates that it can be easily transferred to various visual tasks with different visual frameworks without complex adaptations and adjustments. In Table 1, we also show some methods with generalization abilities. They use point-based [6, 2, 5] or mask-based localization [3] to support

⋆ Work is done during the internship at SenseTime.

Table 1: Comparison with other methods from two perspectives: task generalization and framework generalization.

Method	Task				Framework	
	Cls.	Det.	Segm.	Keyp.	One-stage	Two-stage
Mask R-CNN[3]		✓	✓	✓		✓
Dynamic Head[1]		✓			✓	✓
CenterNet[6]		✓		✓	✓	
PointSet[5]		✓	✓	✓	✓	
LSNet[2]		✓	✓	✓	✓	
UniHead	✓	✓	✓	✓	✓	✓

multiple tasks, or achieve framework generalization by unifying head design [1]. But it is hard to perform various visual tasks with different visual frameworks in a single method.

In addition, the differences with similar methods are also discussed.

Mask R-CNN. Mask R-CNN introduces a strong baseline for mask-based segmentation methods. The mask prediction is made on the upsampled feature map to achieve higher performance. Inevitably, this process introduces extra computational overhead. The dependence on proposal features also makes it hard to be transferred to the one-stage framework. `UniHead` adopts point-based segmentation methods, which greatly decrease the computational cost by predicting the sparse points located in the contour of an instance. For example, Mask R-CNN costs **68 GFLOPs** for mask generation while `UniHead` only needs **5 GFLOPs**. But the modeling error for more complex objects is still challenging in this research community. Besides, in our design, an anchor point is all `UniHead` needs to generate a mask, so it can be easily adopted to the one-stage framework. We aim to conduct a simple but general `UniHead` for different frameworks and visual tasks instead of addressing these task-specific bottlenecks.

PointSet and LSNet. PointSet [5] and LSNet [2] both use different number of points to adapt to different tasks. Following the one-stage detection process, they make dense predictions on different positions of feature maps, which is similar to the FCOS [4] architecture. `UniHead` is not only applicable to the one-stage detectors, but also to the situation where there exists regions of interest, *i.e.*, Faster-RCNN-like frameworks. We also propose dispersible points learning to effectively extract decision-relevant token features, which can automatically adapt to different visual task requirements. Furthermore, we perform token-to-token comparison to enhance global information by transformer encoders. It makes `UniHead` more convenient to transfer to different tasks and architectures directly.

C Discussion on Dispersible Points

Different tasks require different feature representations. As illustrated in Fig.2, dispersible points for classification focus more on the salient part of an instance

Fig. 2: Visualization of dispersible points for classification (left) and segmentation (right).

like knife handles, while for segmentation they are located at the contour of the instance. In `UniHead`, the learning of dispersible points is not supervised in a direct way, so that the extracted features can be decision-relevant and automatically adapt to different visual task requirements.

D Qualitative Results

We show some visualization results of `UniHead` on object detection, instance segmentation, and pose estimation. See Fig.3.

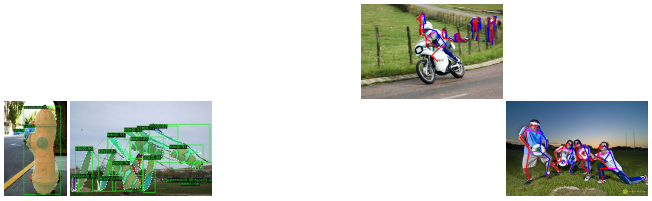

Fig. 3: A visualization of `UniHead` for object detection, instance segmentation and pose estimation on COCO *val* set. Localization points for detection/segmentation are viewed in white dots. Only results with scores higher than 0.4 are shown.

References

1. Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L.: Dynamic head: Unifying object detection heads with attentions. In: CVPR. pp. 7373–7382 (2021)
2. Duan, K., Xie, L., Qi, H., Bai, S., Huang, Q., Tian, Q.: Location-sensitive visual recognition with cross-iou loss. arXiv:2104.04899 (2021)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV. pp. 2961–2969 (2017)
4. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: ICCV. pp. 9627–9636 (2019)
5. Wei, F., Sun, X., Li, H., Wang, J., Lin, S.: Point-set anchors for object detection, instance segmentation and pose estimation. In: ECCV. pp. 527–544 (2020)
6. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv:1904.07850 (2019)