# Supplementary for MPPNet: Multi-Frame Feature Intertwining with Proxy Points for 3D Temporal Object Detection

Xuesong Chen[1,4⋆], Shaoshuai Shi[2⋆⋆⋆], Benjin Zhu[1], Ka Chun Cheung[3], Hang Xu[4], and Hongsheng Li[1⋆⋆]
[1]MMLab, CUHK [2]MPI-INF [3]HKBU [4]Huawei Noah's Ark Lab
{chenxuesong@link, hsli@ee}.cuhk.edu.hk, shaoshuaics@gmail.com

## 1 Architecture Details of MPPNet

In this section, we show more details about our MPPNet.

**3D Proposal-Trajectory Feature Generation with Proxy Points.** We utilize the proposed proxy points as the medium to generate 3D trajectory features of objects. With the help of inherently aligned proxy points, we can decouple the object feature of each proposal in a 3D trajectory into geometry feature and motion feature, which reduces the difficulty of modeling points with significantly different spatial distributions among multi-frames. Specifically, as shown in the green block of Fig. 1, we employ the Set Abstraction operation to aggregate LiDAR geometry feature to proxy points and use the relative locations of proxy point, projected by MLP, as motion feature. The summation of geometry feature and motion feature serves as the object's feature. Next, the proposed grouping strategy temporally divide the long proposal trajectory into a small number of non-overlapping groups for the following multi-frame feature interaction.

**Multi-Frame Feature Interaction.** As the red block shown in Fig. 1, we illustrate more details of the temporal feature fusion among different frames within a group. To reduce the computational and memory cost of intra-group fusion, we first fuse the multi-frame features within each group by summarizing the temporally compressed multi-frame features as residuals to the feature of each group's first frame, using a weight-sharing MLP. With this per-frame fused feature of each group, the following intra-group feature mixing and inter-group feature attention can further propagate and summarize information in an iterative manner. Consequently, a transformer-based prediction head, where a learnable feature embedding is employed as the query and the fused group feature $\hat{G}^i$ are utilized as key and value, can summarize each group's output feature to a global vector embedding with shape of $1 \times 256$, serving as the output of multi-frame feature interaction block.

**Prediction Head.** We jointly utilize each group's summarized vector embedding and embedding extracted from 3D trajectory parameters (7-dim geometry and 1-dim time encoding) to obtain the combined feature vector (see blue block

⋆ Equal contributions
⋆⋆ Corresponding authors

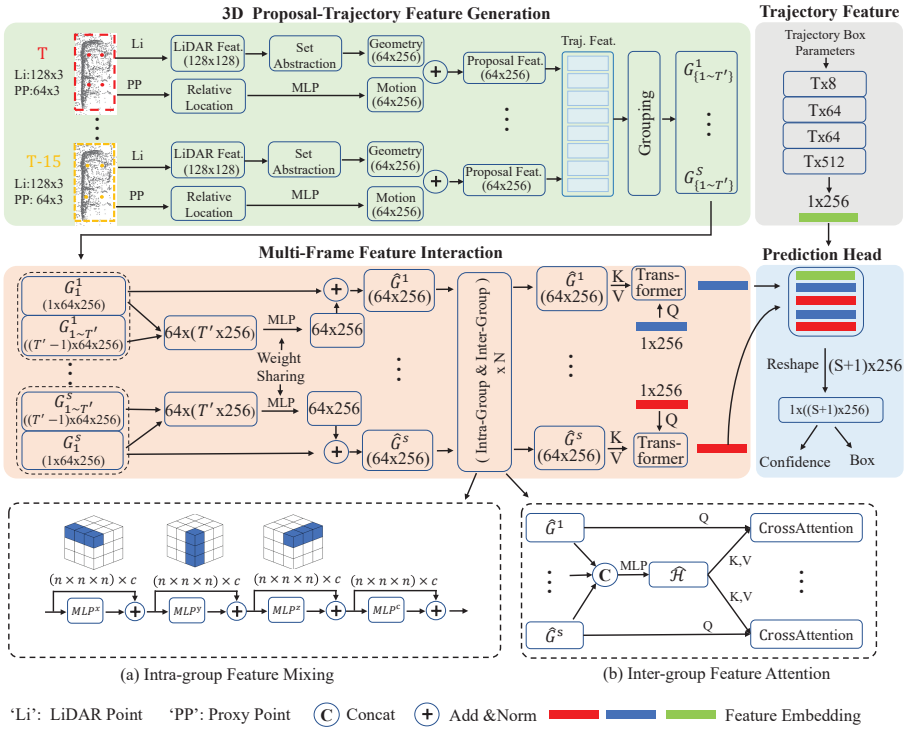


**Fig. 1.** The architecture details of MPPNet with 16-frame input sequence, where $G^i$ indicates the initial feature of each group, $\hat{G}^i$ means the fused group-wise feature and $\hat{\mathcal{H}}$ denotes the global summarized feature for the inter-group feature attention.

in Fig. 1) for getting more accurate 3D detection results. Specifically, for trajectory feature learning, as shown in the gray block of Fig. 1, we adopt a simple PointNet-based architecture [1], where a MLP is adopted to project the parameters to high-dimension feature space and then the max pooling followed with a MLP is utilized to reduce the temporal and channel information. We finally obtain the embedding of a trajectory with shape of $1 \times 256$ for the confidence prediction and box regression.

## 2 More Training Loss Details

As mentioned in Sec. 3.3 of the original paper, MPPNet is trained with a combination loss of confidence loss $\mathcal{L}_{\text{conf}}$, regression loss $\mathcal{L}_{\text{reg}}$, denoted as $\mathcal{L} = \mathcal{L}_{\text{conf}} + 2\mathcal{L}_{\text{reg}}$. For $\mathcal{L}_{\text{conf}}$, we set the training targets as the 3D IoU between the 3D proposals and their corresponding ground-truth boxes. Then, we follow [2] to assign the confidence prediction target:

$$c^t = \min\left(1, \max\left(0, \frac{\text{IoU} - \alpha_B}{\alpha_F - \alpha_B}\right)\right), \tag{1}$$

where $\alpha_F = 0.75$ and $\alpha_B = 0.25$ are the foreground and background thresholds of IoU, respectively. Then the binary cross entropy loss [2] is adopted for the predicted confidence $c^t$ and ground-truth confidence $c$ to compute the IoU-guided confidence loss:

$$\mathcal{L}_{\text{conf}} = -c^t \log(c) - \left(1 - c^t\right) \log(1 - c). \tag{2}$$

Note that only the first group feature $\hat{G}^1$ (including current frame) is supervised by $\mathcal{L}_{\text{conf}}$ because the confidence targets are defined by current frame's prediction boxes and ground-truth boxes.

For regression loss $\mathcal{L}_{\text{reg}}$, a hybrid of shape regression ($\mathcal{L}_{\text{shape}}$) and corner regression ($\mathcal{L}_{\text{corner}}$) formulations is adopted to make the estimation more robust. For $\mathcal{L}_{\text{shape}}$, the targets are encoded by proposals and their corresponding ground-truth boxes, given by:

$$\begin{aligned} &x = \tfrac{x^g - x^c}{d},\ y = \tfrac{y^g - y^c}{d},\ z = \tfrac{z^g - z^c}{h^c}, \\ &l = \log\left(\tfrac{l^g}{l^c}\right),\ w = \log\left(\tfrac{w^g}{w^c}\right),\ h = \log\left(\tfrac{h^g}{h^c}\right),\ \theta^t = \theta^g - \theta^c, \end{aligned} \tag{3}$$

where superscript $c$ and $g$ indicate the parameters of proposals and ground-truth bounding boxes and $d = \sqrt{\left(\frac{l^c}{2}\right)^2 + \left(\frac{w^c}{2}\right)^2}$. Following [2], the $\mathcal{L}_{\text{shape}}$ is formulated as following:

$$\mathcal{L}_{\text{shape}} = \mathbb{I}\left(\text{IoU} \geq \alpha_R\right) \sum_{\mu \in x,y,z,l,w,h,\theta} \mathcal{L}_{\text{smooth-L1}}\left(\mu, \mu^t\right), \tag{4}$$

where $\mathbb{I}\left(\text{IoU} \geq \alpha_R\right)$ means that only proposals with IoU $\geq \alpha_R$ contribute to the regression loss ($\alpha_R = 0.55$). All of the feature embedding of each group, the trajectory embedding as well as the final combined embedding are supervised by the regression loss.

On the other hand, the corner regression loss $\mathcal{L}_{\text{corner}}$ is formulated as follows:

$$\mathcal{L}_{\text{corner}} = \frac{1}{8} \min\left(\sum_{i=1}^{8} |C_i - C_i^*|\right), \tag{5}$$

where $C_i$ and $C_i^*$ are predicted corners and the ground truth corners. The corner loss is only added on the final combined embedding. The $\mathcal{L}_{\text{reg}}$ is presented as:

$$\mathcal{L}_{reg} = \mathcal{L}_{\text{shape}} + \mathcal{L}_{\text{corner}}. \tag{6}$$

## 3 More Ablation Studies

In this section, we provide more ablation experiments for reference. Specifically, the experiments are conducted on Waymo datasets by optimizing on the training set and evaluating on the validation set. We train MPPNet on the vehicle category for 3 epochs by taking four-frame point cloud as input, with the same settings of our original paper. The CenterPoint [3] is adopted as the 1-stage network and we take the mAPH (LEVEL 2) as the default metric for comparison.

**Table 1.** Effects of iteration number of intra-group and inter-group feature interaction block.

| Iteration of intra-inter block | mAPH@L2 |
|---|---|
| 1 | 72.78 |
| 2 | 73.08 |
| 3 | 73.02 |

**Table 2.** Effects of adopting features (BEV) of 1-stage network.

| Feature Source | mAPH@L2 |
|---|---|
| Point feat.+ BEV feat. | 72.99 |
| Point feat. | 73.08 |

**Effects of Iteration Number of Multi-frame Feature Interaction.** We investigate the impact of different iteration numbers of the multi-frame feature interaction block, consisting of the proposed intra-group feature mixing and inter-group feature attention. As shown in Table 1, 2-iteration improves the performance of the 1-iteration by 0.3%, but with more iterations, *i.e.*, 3-iteration, the performance is similar to that of the 3-iteration while increasing the amount of computations. Therefore, we empirically chose 3-iteration as the default setting of MPPNet.

**Effects of Incorporating Features from the First Stage RPN.** MPPNet only uses the proxy point features aggregated from the raw LiDAR points by default, and here we also investigate the impact of additional 1-stage network features, such as bird-eye view (BEV) features. As shown in the table 2, the additional BEV features from CenterPoint achieve similar performance with our default setting, proving that the high-level features from RPN can not further benefit the performance of our multi-frame feature encoding and interaction head. This further verifies our argument that the multi-frame raw points of each proposal box already provide enough and accurate information for the box refinement in the second stage.

## References

1. Li, Z., Wang, F., Wang, N.: Lidar r-cnn: An efficient and universal 3d object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7546–7555 (2021)
2. Sheng, H., Cai, S., Liu, Y., Deng, B., Huang, J., Hua, X.S., Zhao, M.J.: Improving 3d object detection with channel-wise transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2743–2752 (2021)
3. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11784–11793 (2021)