Supplementary Material – PnPNet: End-to-End Perception and Prediction with Tracking in the Loop

In this supplementary material, we first introduce the backbone network architecture of PnPNet, as well as our implementation details on the nuScenes object detection benchmark. We then showcase more fine-grained evaluation results on motion forecasting by comparing PnPNet and the baseline at all detection recall rates.

1. Backbone Network Architecture

We first show in Figure 1 the architecture of the backbone network. We use bird's eye view (BEV) occupancy map as input representation for the LiDAR, and concatenate the BEV representations of multi-sweep LiDAR point clouds (1 current frame + N previous frames) along the height dimension. We employ $N = 9$ in nuScenes dataset and $N = 4$ in ATG4D dataset. For map representation, we exploit both geometric and semantic priors similar to HDNet [3]. Specifically, we do ground height subtraction on the multi-sweep LiDAR point clouds, and compute two additional binary raster images representing the drivable region and the lane graph respectively (for the current frame only). The map raster images are concatenated with the LiDAR BEV representation along the height dimension.

Given the aforementioned BEV representation of both LiDAR and HD maps as input, we first apply three `Conv2D` layers to down-sample the input BEV images by a factor of 4. We then apply a cross-scale module sequentially three times. This cross-scale module is inspired by the Inception block with residual connections [2]. The difference is that feature maps are spanned at multiple scales (3 in our case), and each scale receives information from all other scales. This leads to a better trade-off between accuracy and speed. After three cross-scale modules, we apply a feature pyramid network [1] to combine multi-scale feature maps, resulting in a $4\times$ down-sampled BEV feature map with 128 channels. For the detection header we simply use 4 `Conv2D` layers each with 128 filters. The detection header outputs $(6 + 1) * C$ channels as dense detection estimations, which correspond to $(x, y, w, l, \sin\theta, \cos\theta, \text{score\_logit})$ for each object category.

2. Implementation Details on nuScenes

We use the same network architecture on both nuScenes and ATG4D datasets. On nuScenes, following the dataset rule imposed by the creators of the dataset, we aggregate 10 sweeps of LiDAR point clouds (1 current and 9 previous) corresponding to 0.5 seconds of past history. We consider the point clouds within a region of $[-50, 50] \times [-50, 50] \times [-3, 5]$ meters around the ego car. We use a voxel size of $0.15625 \times 0.15625 \times 0.25$ meters, leading to a voxel grid size of $640 \times 640 \times 320$ as input.

We apply frame-level data augmentation during training. Specifically, labels at non-key frames are linearly interpolated from labels at adjacent key frames. For each frame, we apply random scaling (0.95 $\sim$ 1.05 for all 3 axes), translation (-1 $\sim$ 1 meters for XY axes and -0.2 $\sim$ 0.2 meter for Z axis), rotation (-45 $\sim$ 45 degrees along Z axis) and flipping (along X axis) to both 3D LIDAR point clouds and 3D object labels.

The model is trained on the car class, and we ignore labels that have 0 LiDAR point inside the box or outside the 50 meters range with respect to the ego car. During training we define positive samples as pixels with IoU (assuming ground-truth size and orientation) larger than 0.9, and define negative samples as smaller than 0.4 IoU. We use Adam optimizer and train with batch size of 8 for 4 epochs. The initial learning rate is 0.001, and is decayed by 0.1 at 2.8 and 3.6 epochs respectively.

3. Fine-Grained Evaluation of Motion Forecasting

While in the submission we evaluate prediction errors (ADE, FDE) of motion forecasting at fixed object recall rates, here we provide more fine-grained evaluation of FDE (at 1s, 2s and 3s respectively) for all recall rates. We show evaluation results in Figure 2, where we observe that the proposed PnPNet not only achieves higher object detection recall, but also outperforms the baseline in prediction consistently at all recall rates.

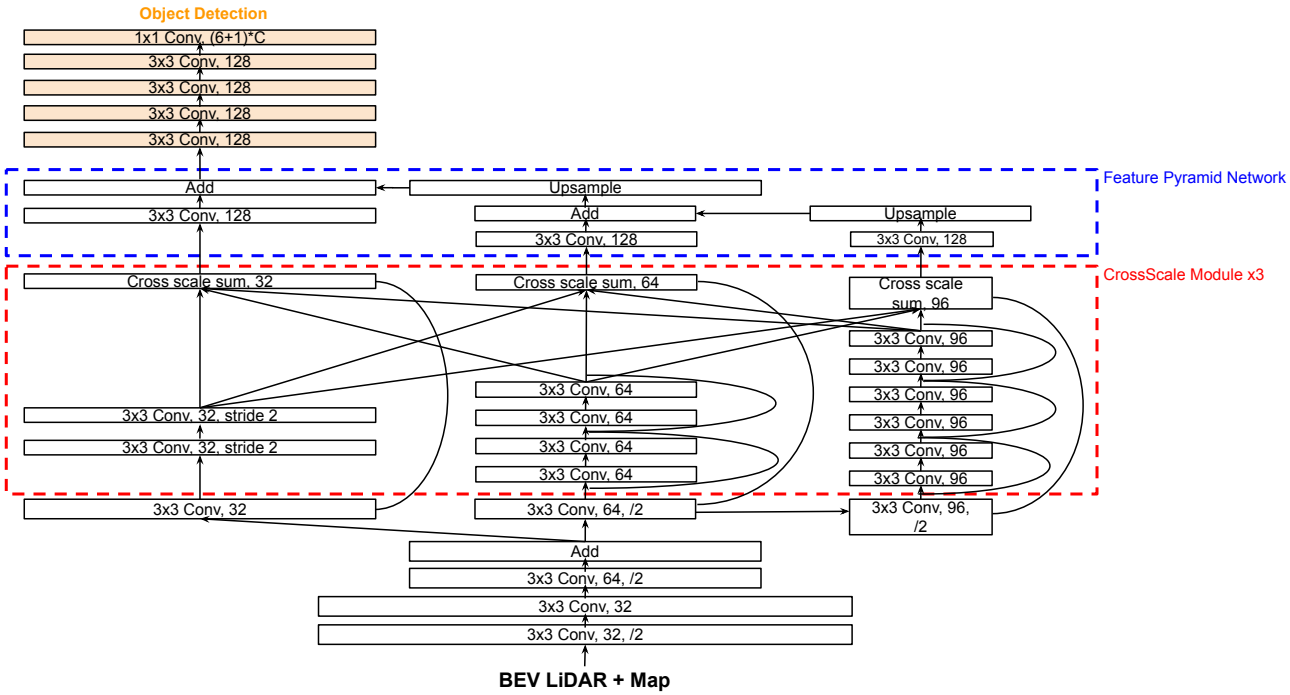

Figure 1. Architecture of the backbone network.

References

[1] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1

[2] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 1

[3] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *CoRL*, 2018. 1

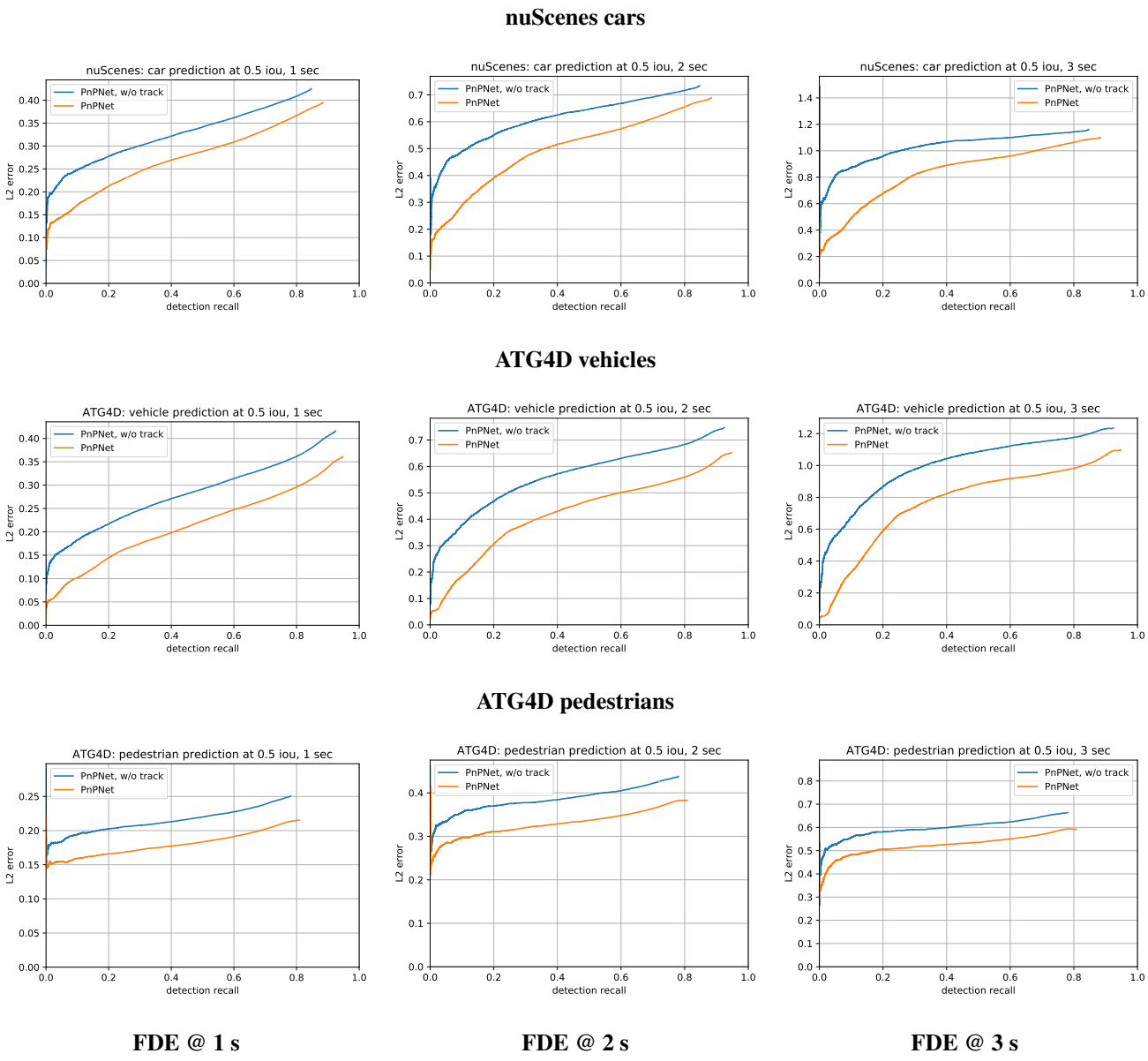

Figure 2. Evaluation of Final Displacement Error (FDE) at 1s, 2s and 3s motion forecasting for all recall rates on nuScenes (1st row) and ATG4D datasets (2nd and 3rd rows).