

Progressive Multi-Modal Fusion for Robust 3D Object Detection

Supplementary Material

Anonymous Author(s)

Affiliation

Address

email

1 In this supplementary material, we provide additional details on various aspects of our work. First, we
2 provide further architecture details for our fusion module and decoder in Sec. 1. Next, we present the
3 loss details of the three pre-training objectives for our proposed LiDAR and Multi-View Camera Mask
4 Modeling in Sec. 2. In Sec. 3 we describe the strategy for selecting the subsets of the dataset used
5 in our data efficiency experiment (Sec. 4.2 of the main manuscript). Lastly, we provide additional
6 qualitative results for ProFusion3D in Sec. 4.

7 1 Additional Architectural Details of ProFusion3D

8 **Inter-Intra Fusion:** For each of the inter-intra base units, the corresponding channel dimension of
9 the query, key, and value embeddings are set to 192. The following Feed-Forward Network (FFN)
10 consists of two fully-connected layers with GeLU activation after the first one. The first layer expands
11 the input channels to 1024 while the second layer condenses it back to the original embedding
12 dimensions (192).

13 The convolutional block of the fusion module uses two consecutive 3×3 depthwise separable
14 convolutions, each with dilation rates of 2 and 4, respectively. The number of channels in these
15 convolutions is twice the embedding dimensions (384).

16 **Decoder:** We employ two DETR-style decoder layers in the BEV decoder, PV decoder, and Joint
17 Decoder. We follow similar parameter settings as described in [1] for each of the decoder layers. The
18 sequence of operations within each layer is as follows: self-attention, normalization, cross-attention,
19 normalization, FFN, and normalization. The embedding dimensions are set to 256, and the FFN
20 channels are set to 2048. Both the attention and FFN dropout rates are configured to 0.1. We
21 set the number of queries to 600 for initializing Q_0 . The initial object queries Q_0 interact with
22 the 3D position-aware BEV features F_{bev}^{3D} in the BEV decoder to update their representations to
23 Q_{bev} . In parallel, Q_0 also interacts with the 3D position-aware PV features F_{pv}^{3D} in the PV decoder
24 to update their representations to Q_{pv} . Following this, $Q_{\text{join}} = [Q_{\text{bev}}; Q_{\text{pv}}]$ interacts with the 3D
25 position-aware joint features $F_{\text{join}}^{3D} = [F_{\text{bev}}^{3D}; F_{\text{pv}}^{3D}]$ in the joint decoder to generate the final updated
26 query representations.

27 We use two FFNs to predict the 3D bounding boxes and the classes using the updated queries in each
28 of the decoder layers. The prediction for each decoder layer is then as follows:

$$\hat{b}_i^d = \phi^{\text{reg}}(Q_i^d), \quad \hat{p}_i^d = \phi^{\text{cls}}(Q_i^d) \quad (1)$$

29 where ϕ^{reg} and ϕ^{cls} represent the FFNs for regression and classification, respectively. Q_i^d are the
30 updated object queries of the i -th decoder layer of the d -th decoder, where $d \in \{\text{bev}, \text{pv}, \text{joint}\}$.

31 We train ProFusion3D through set prediction by using bipartite matching for one-to-one assignment
32 between predictions and ground truths. Specifically, we use the focal loss for classification and L1
33 loss for 3D bounding box regression:

$$L(y, \hat{y}) = \lambda_1 L_{cls}(p, \hat{p}) + \lambda_2 L_{reg}(b, \hat{b}) \quad (2)$$

where λ_1 and λ_2 are the hyperparameters to balance the two loss terms.

2 Loss Functions for Pre-Training Objectives

For the multi-modal masked modeling, we introduce three pre-training objectives: masked token reconstruction, unmasked token denoising, and masked token cross-modal attribute prediction. To train each objective we employ the following losses:

Masked Token Reconstruction: In this pre-training objective, we reconstruct each masked image patch from the PV branch and voxels from the BEV branch. For the image patch reconstruction, we employ L1 loss between the predicted values of the masked pixels and the corresponding RGB values as follows:

$$L_{L1} = \frac{1}{N_{mp}} \sum_{i=1}^{N_{mp}} |\hat{I}_i - I_i| \quad (3)$$

where \hat{I}_i and I_i are the predicted and ground truth RGB value of the i -th masked pixel, respectively, and N_{mp} is the total number of masked pixels.

For the voxel reconstruction, let $P_{gt,i} = \{x_1, x_2, \dots, x_N\}$ be the i -th masked voxel where N is the number of fixed points in voxels and $P_{rec,i} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\}$ be the corresponding reconstruction. Then the Chamfer loss is defined as follows:

$$L_{\text{Chamfer}}(P_{gt,i}, P_{rec,i}) = \frac{1}{|P_{gt,i}|} \sum_{x \in P_{gt,i}} f(x, P_{rec,i}) + \frac{1}{|P_{rec,i}|} \sum_{\tilde{x} \in P_{rec,i}} f(\tilde{x}, P_{gt,i}), \quad (4)$$

$$f(x, P) = \|x - P^j\|_2^2, \text{ with } j = \arg \min_k \|x - P^k\|_2^2$$

where $\|\cdot\|_2$ denotes the L2-norm. This loss function ensures that each point in the ground truth set P_{gt} is close to some point in the reconstructed set P_{rec} and vice versa.

Unmasked Token Denoising: In this pre-training objective, we learn to predict noise for each unmasked image patch and voxel. For the unmasked image patches, we employ L1 loss between the predicted noise and the actual noise added as follows:

$$L_{\text{denoise_image}} = \frac{1}{N_{up}} \sum_{i=1}^{N_{up}} |\hat{n}_i - n_i| \quad (5)$$

where \hat{n}_i is the predicted noise and n_i is the actual noise added to the i -th unmasked pixel, and N_{up} is the total number of unmasked pixels.

For the voxel denoising, let $N_{a,i} = \{n_1, n_2, \dots, n_N\}$ be the noise added to the i -th unmasked voxel and $N_{p,i} = \{\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_N\}$ be the corresponding noise prediction.

$$L_{\text{denoise_voxel}}(N_{a,i}, N_{p,i}) = \frac{1}{|N_{a,i}|} \sum_{n \in N_{p,i}} \min_{\tilde{n} \in N_{p,i}} \|n - \tilde{n}\|_2^2 + \frac{1}{|N_{p,i}|} \sum_{\tilde{n} \in N_{p,i}} \min_{n \in N_{a,i}} \|\tilde{n} - n\|_2^2. \quad (6)$$

Masked Token Cross-Modal Attribute Prediction: In this pre-training objective, we predict pixel intensities for points in masked voxels and depth values for masked image patches. To predict pixel intensities for points in masked voxels, we do this jointly with the masked voxel reconstruction. Hence, while predicting the points in the masked voxel, we also predict the corresponding pixel intensities. For pixel intensity prediction, we add a loss term to the Chamfer distance loss of masked token reconstruction, by replacing f in Eq. (4) with \tilde{f} :

$$\tilde{f}(x, P) = \|x - P^j\|_2^2 + \lambda |x_I - P_I^j|, \text{ with } j = \arg \min_k \|x - P^k\|_2^2 \quad (7)$$

where λ is the loss balancing term, x_I is the ground truth pixel intensity, and P_I^j is the predicted pixel intensity.

65 Following, for depth prediction for masked image patches, let d and \hat{d} denote the ground-truth and
 66 the predicted depth, respectively. Our loss function for depth estimation is then defined by

$$L_{\text{depth}}(d, \hat{d}) = \frac{1}{N_{mp}} \sum_i^{N_{mp}} (\log d_i - \log \hat{d}_i)^2 - \frac{1}{N_{mp}^2} \left(\sum_i^{N_{mp}} (\log d_i - \log \hat{d}_i) \right)^2 \\ + \left(\frac{1}{N_{mp}} \sum_i^{N_{mp}} \frac{d_i - \hat{d}_i}{d_i} \right)^2. \quad (8)$$

67 This loss term is a combination of the scale-invariant logarithmic error and the relative squared error.

68 **3 Strategy for Subset Selection in Data Efficiency Experiments**

69 In Sec. 4.2 of our main manuscript, we perform a data efficiency experiment that utilizes subsets of
 70 20%, 40%, 60%, and 80% from the 100% training annotated data of nuScenes. To select scenes for
 71 each subset of the training dataset, we sorted the dataset based on scene timestamps and then used
 72 a systematic sampling method. Specifically, we divided the scenes into five groups based on their
 73 indices and selected scenes according to these groups:

- 74 • For a 20% subset, we included all scenes from one group (i.e., those with indices where i
 75 $\bmod 5 = 0$).
- 76 • For a 40% subset, we included scenes from two groups (i.e., those with indices where i
 77 $\bmod 5 \in \{0, 2\}$).
- 78 • For a 60% subset, we included scenes from three groups (i.e., those with indices where i
 79 $\bmod 5 \in \{0, 2, 4\}$).
- 80 • For an 80% subset, we included scenes from four groups (i.e., those with indices where i
 81 $\bmod 5 \in \{0, 1, 2, 4\}$).

82 This systematic sampling method helps to minimize temporal dependencies between frames and
 83 ensures that the reduced datasets retain a similar level of diversity as the complete dataset.

84 **4 Visualization**

85 In Fig. 1, we provide additional qualitative results of our proposed ProFusion3D on the nuScenes
 86 dataset. The dataset encompasses urban road scenes with objects ranging from cars, trucks, bicycles,
 87 motorbikes, people, barriers, and cones in a very cluttered environment with occlusions. Despite
 88 these challenging conditions, our ProFusion3D consistently and accurately detects all these objects.

89 **References**

- 90 [1] Y. Liu, T. Wang, X. Zhang, and J. Sun. Petr: Position embedding transformation for multi-view
 91 3d object detection. In *European Conference on Computer Vision*, pages 531–548. Springer,
 92 2022.

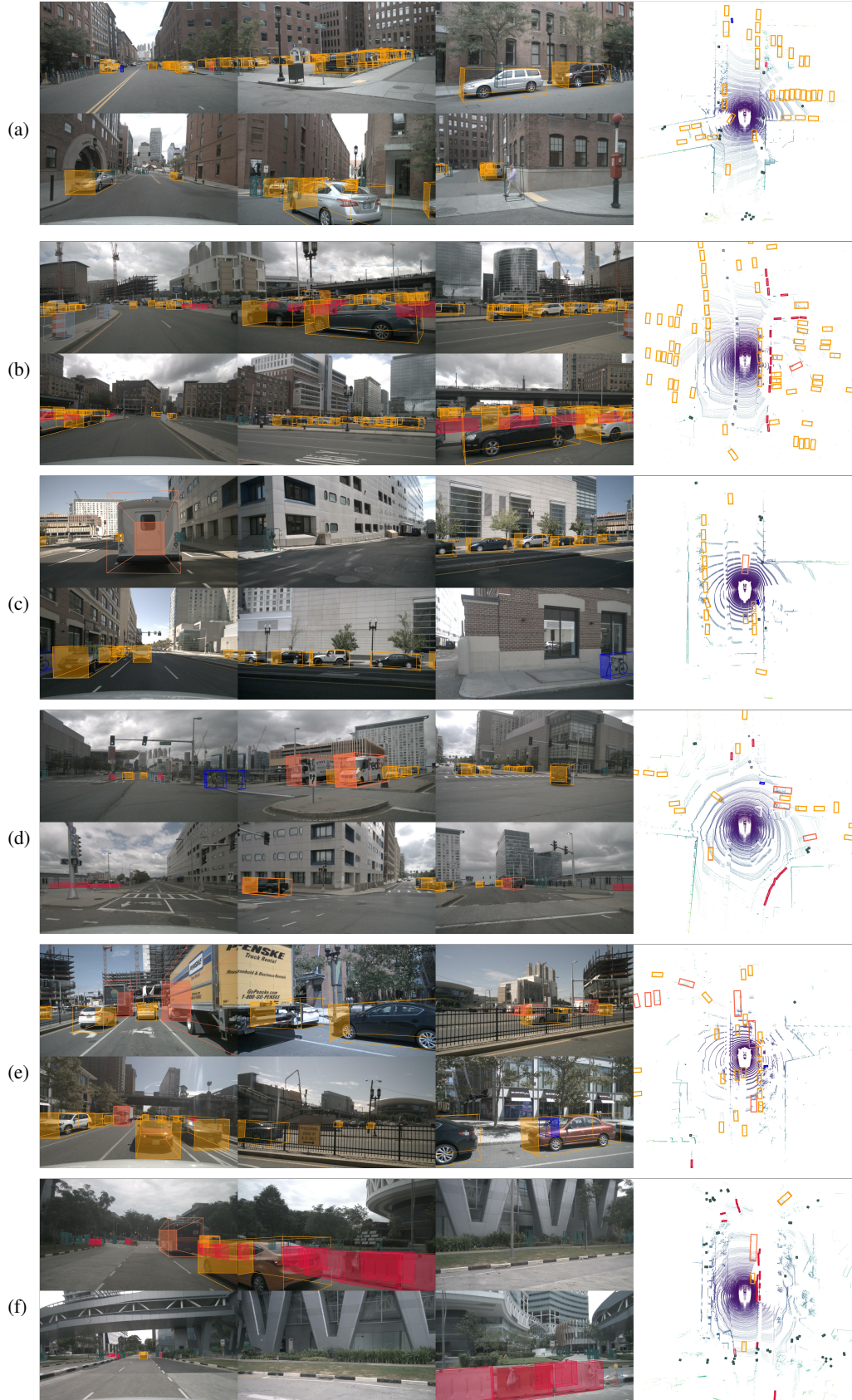


Figure 1: Visualization of 3D object detection prediction of our proposed ProFusion3D on the validation set of nuScenes. Classes are color-coded as follows: ■ car, ■ barrier, ■ truck, ■ cone, ■ bicycle, ■ person.