Supplementary Material for DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection

Yingwei Li[1,2*] Adams Wei Yu[2*] Tianjian Meng[2] Ben Caine[2] Jiquan Ngiam[2] Daiyi Peng[2]
Junyang Shen[2] Yifeng Lu[2] Denny Zhou[2] Quoc V. Le[2] Alan Yuille[1] Mingxing Tan[2]
[1]Johns Hopkins University [2]Google
{ywli, adamsyuwei, tanmingxing}@google.com

A. Impact of Alignment Quality

In this section, we provide more detailed experimental settings and more results of our preliminary experiments in Section 3.2 in the main paper.

Experimental Settings. We use the 3D-MAN++ Pedestrian model mentioned in Section 4.1 and Section B. To examine the alignment quality, InverseAug and all data augmentations are removed. Then, we apply different magnitude of RandomRotation [12] to both single-modal and multi-modal models. Finally, for the same perturbation magnitude, we compute the performance gap for the best validation results from the single-modal and multi-modal models.

Additional Results. Besides testing with RandomRotation [12], we also test with RandomFlip [12], another commonly used data augmentation strategy for 3D point cloud object detection models. Specifically, RandomFlip flip the 3D scene along the Y axis with a given probability p. Here, we set the probability as 0%, 50%, and 100%, respectively, and the results are shown in Table 1. The observation is similar: when applying large magnitude data augmentation, the benefit from multi-modal fusion diminishes. For example, when applying zero-probability RandomFlip (*i.e.*, not applying this data augmentation), the improvement is the most significant (+2.3 AP); when flip probability is 100% (*i.e.*, flip the 3D scene every time), the improvement is almost zero (+0.03 AP).

B. Implementation Details of 3D Detectors

In the main paper, we mainly focus on providing more details about DeepFusion due to the space limitation. In this section, we will also illustrate other important implementation details to build the strong 3D object detection models.

Point cloud 3D object detection methods. We reimplement three popular point cloud 3D object detection methods, PointPillars [4], CenterPoint [11], and 3D-MAN [10]. As mentioned in Section 2, PointPillars voxelize the point cloud by pillars, a single tall elongated voxel per map location, to construct bird eye view pseudo image; finally, the pseudo image is fed to an anchor-based object detection pipeline. A high-level model pipeline is shown in Figure 1. CenterPoint is also a pillar-based method, but using anchor-free detection head instead. Note that we only implemented the PointPillars-based single-stage version of CenterPoint. 3D-MAN is similar to CenterPoint, and the main difference is when computing the loss, 3D-MAN uses a Hungarian algorithm to associate the prediction and the ground-truth (See Section 3.1 of Yang *et al.* [10] for more details).

Flip Probability	0%	50%	100%
Single-Modal	72.6	76.7	71.8
Multi-Modal	74.9	76.8	71.9
Improvement	+2.3	+0.10	+0.03

Table 1. Performance gain by multi-modal fusion diminishes as the magnitude of RandomFlip [12] goes up, indicating the importance of accurate alignment. InverseAug is not used here. On the Waymo Open Dataset pedestrian detection task, the LEVEL 1 AP improvements from single-modal to multi-modal are reported.

Our improved implementations. We also introduce two simple but effective findings that significantly improve the point cloud 3D object detection baselines. We take the PointPillars framework as an example to introduce them, but these techniques can be naturally applied to other point cloud 3D object detection frameworks, such as CenterPoint and 3D-MAN. As shown in Figure 1, we build upon the PointPillars model and indicate our modifications the red dotted line boxes. The NAS block depicts the voxel feature encoder found using architecture search. We also replace the ReLU [2, 6] activation function in the original frameworks with SILU [1,7]. Our improved models (named as PointPillars++, CenterPoint++, and 3D-MAN++) shows better performance that its baseline method as shown in Table 4 in the main paper. For example, for 3D-MAN, after applying these two techniques, the LEVEL_2 APH is improved from 52.2 to 63.0. This improvement is significant,

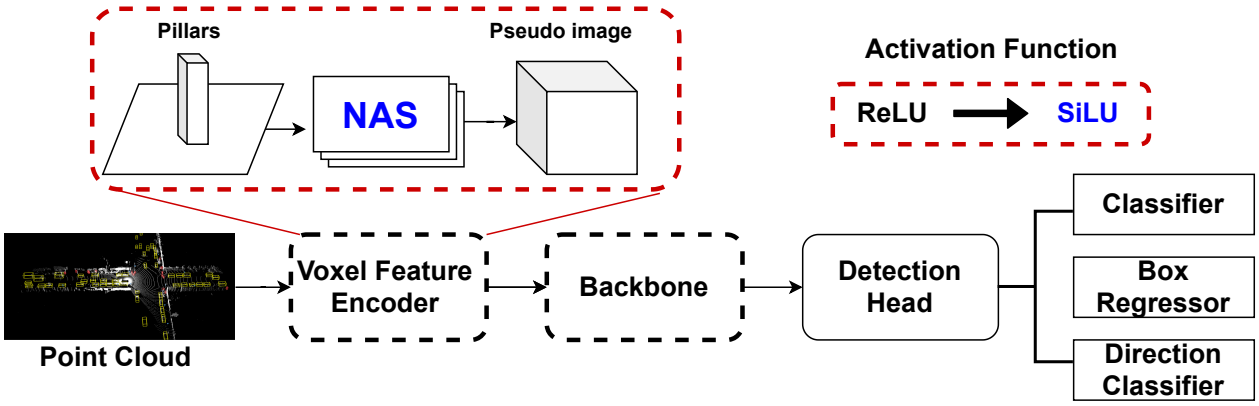

Figure 1. The overview of PointPillars framework and its improved implementation (marked in red dashed boxes). Our improved implementation replace the original Voxel Feature Encoder from one fully-connected layer to a multilayer perceptron, whose hyperparameter (such as the number of layers, and the number of filters) are automatically discovered by Neural Architecture Search [13]. In addition, we change the non-linear activation function from ReLU [2, 6] to SILU [1, 7].

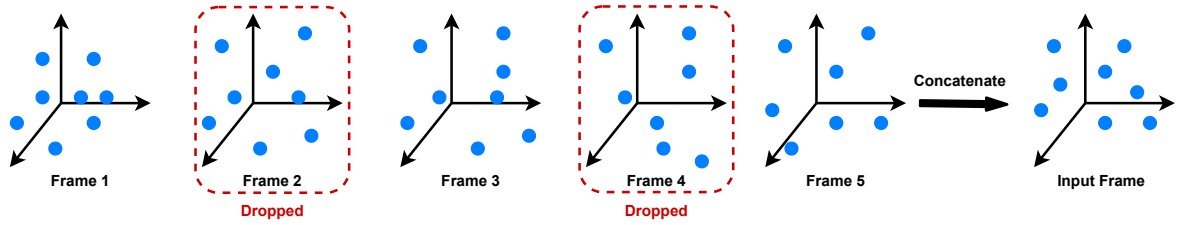

Figure 2. The DropFrame process during training of a 5-frame model. In this example, Frame 2 and 4 are dropped out before collapsing. The frames selected to be dropped are randomly selected in each training step. During inference, all frames are used as input without any frames dropped.

and is consistently observed from other metrics and from other baselines.

Other training details. We use both LEVEL_1 and LEVEL_2 difficulty data for training. Since the LEVEL_2 data is difficult for model to predict, we use the uncertainty loss [5] during training to tolerate the models to detect low-confident objects with low accuracy.

Details of the submitted models. We apply DeepFusion to CenterPoint to prepare our models for submission. We enlarge the Max Rotation for the RandomRotation augmentation to 180° (120° for Pedestrian model) since we see its benefit according to Table **??**. We also enlarge the pseudo-image feature resolution from 512×512 to 704×704. We combine the information from previous frames by simply concatenating the point-clouds across the last N frames together. As shown in Figure 2, to prevent the over-fitting issue under the multi-frame setting, we propose DropFrame, that randomly drop the point cloud from previous frames. Our very best model concatenates 5 frames, and with dropframe probabilities 0.5 during training. Besides, we also apply model ensemble and Test-Time Augmentation (TTA) by weighted box fusion (WBF) [3]. For TTA, we use yaw rotation, and global scaling. Specificly, we use [0°, ±22.5°, ±45°, ±135°, ±157.5°, ±180°] for yaw rotation, and [0.95, 1, 1.05] for global scaling. For model ensemble, we obtain 5 different type of models with different pseudo-image feature resolution and different input modality, *i.e.*, single-modality 512 / 704 / 1024 resolution, and multi-modality 512 / 704 resolution. For each type of model, we train 5 times with different random seed. Then, we rank all 25 models with the performance on validation set and ensemble top-k models, where k is the optimal value to get the best results on the validation set.

C. Comparison with Larger Single-Modal Models

The goal of this section is to compare the Single-Modal baseline with DeepFusion under the same computational budget. To achieve this, we first scale up the single-modality model. Since we have sufficiently scaled up the voxel feature encoder and backbone when building the baseline models, enlarging the resolution of the pseudo image is probably the most effective way for further scaling the

single-modal model to match the latency with multi-modal model, and thus we adopt such a strategy here. Specifically, we train the models under resolutions ranging from 512 up to 960, and test the performance of each setting. Figure 3 clearly shows that DeepFusion achieves 67.0 L2 APH with 0.32s latency while the single-modal model can only achieve 65.7 L2 APH with the same latency budget. Further scaling up the single-modal model brings marginal gain to the performance, which is capped at 66.5 L2 APH and still worse than DeepFusion.

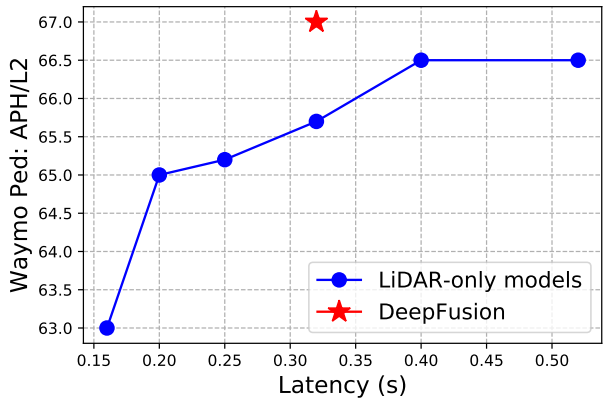

Figure 3. Model latency *vs.* detection performance. DeepFusion significantly outperforms single-modal models under all latencies.

Limitations: This paper focuses on fusing lidar and camera information. However, our proposed method could also be potentially extended to other modalities, such as range image, radar and high-definition map. Besides, we simply adopt voxel-based methods such as PointPillars [4], but it is possible to further improve the performance by adopting strong baselines [9].

License of used assets: Waymo Open Dataset [8]: Waymo Dataset License Agreement for Non-Commercial Use (August 2019). [1]

References

[1] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018. 1, 2

[2] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951, 2000. 1, 2

[3] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. *arXiv preprint arXiv:2112.09205*, 2021. 2

[4] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019. 1, 3

[5] Gregory P Meyer and Niranjan Thakurdesai. Learning an uncertainty-aware object detector for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10521–10527. IEEE, 2020. 2

[6] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, pages 807–814, 2010. 1, 2

[7] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 1, 2

[8] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 3

[9] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5725–5734, 2021. 3

[10] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3d-man: 3d multi-frame attention network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1863–1872, 2021. 1

[11] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, 2021. 1

[12] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018. 1

[13] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[1] https://waymo.com/intl/en_us/dataset-download-terms/