Supplementary Materials: FULLER: Unified Multi-modality Multi-task 3D Perception via Multi-level Gradient Calibration

Zhijian Huang[1*] Sihao Lin[2*] Guiyu Liu[3*] Mukun Luo[4] Chaoqiang Ye[3] Hang Xu[3]
Xiaojun Chang[5,6] Xiaodan Liang[1,6†]
[1]Shenzhen Campus of Sun Yat-sen University [2]RMIT University [3]Huawei Noah's Ark Lab
[4]Shanghai Jiao Tong University [5]University of Technology Sydney [6]MBZUAI
huangzhj56@mail2.sysu.edu.cn, {linsihao6,guiyuliou,chromexbjxh,cxj273,xdliang328}@gmail.com,
luomukun@sjtu.edu.cn, yechaoqiang@huawei.com

A. Asset Acknowledgement

We introduced the dataset in the main text. The used code assets are shown in Tab. 1.

B. Network Architecture

We briefly introduced the network architecture in the Sec. 3.1 of the main text. More design details along with necessary math notations are presented in this section.

Modality Branch. For image branch, the image features extracted from the Swin-T [13] will be transformed to BEV (bird-eye's-view) representation via LSS [15], where each pixel has a discrete depth distribution. The depth is discretized into $[1, 60]$ meters with a step size of 1 meter. For LiDAR branch, the point cloud features coming out from the VoxelNet [27] will be compressed to 1 along the the z-axis, forming the BEV representation. The image BEV presentations f^{img} and LiDAR BEV presentations f^{lid} are concatenated and then fed into the fusion block, resulting in the the fusion feature f^{fuse}. The fusion block is a FPN with two down-sample and two up-sample conv blocks, which is implemented as same as BEVFusion [14].

Detection Head. As shown in Fig. 1, both detection head and segmentation head are query-based. Given the fused BEV feature f^{fuse}, the detection head will initial queries by an auxiliary heatmap module, and sort out the top-N candidates as the object queries:

$$Q^d = \texttt{top-n(heatmap(}f^{fuse}\texttt{))} \in \mathbb{R}^{N \times C^d}, \quad (1)$$

where $C^d = 128$ is the feature dimension and $N = 200$ is the number of queries, which is a little more than the ground truth. The heatmap module is borrowed from Transfusion [1], which screens out the local maxima from the heatmap to prevent the object queries from scattering spatially too closely. Then, a one-layer transformer decoder is used to update the Q^d, where f^{fuse} is served as key and value, and Q^d itself is served as query, respectively. Finally, a simple feed-forward network predict the boundary boxes B using the updated query:

$$\begin{aligned} Q^{d'} &= \texttt{transDec(}Q^d, f^{fuse}\texttt{)}, \\ B &= \texttt{ffn(}Q^{d'}\texttt{)}. \end{aligned} \quad (2)$$

For the loss functions, we employ the smoothed l_1 loss and standard focal loss as the regression loss and classification loss, respectively. The weights of regression loss, classification loss, and heatmap loss are 0.25, 1.0, and 1.0.

Segmentation Head. To generate the segmentation queries, the semantic classes L is first converted to one-hot vectors, and then projected by a linear layer:

$$Q^s = \texttt{projector(one-hot(}L\texttt{))} \in \mathbb{R}^{M \times C^s}, \quad (3)$$

where $M = 6$ is the number of semantic categories and $C^s = 256$ is the embedding dimension. To align with the output shape, the fused BEV feature f^{fuse} will be transformed to the segmentation features $f^{fuse'}$ by an interpolation layer and a 1D conv layer:

$$f^{fuse'} = \texttt{1d-conv(intpl(}f^{fuse}\texttt{))} \in \mathbb{R}^{H \times W \times C^s}, \quad (4)$$

where $H, W = 200$ is the spatial size. Similarly, a one-layer transformer decoder is employed to update the Q^s with the input $f^{fuse'}$ and Q^s itself. The updated queries is processed by the MLP to generate mask embeddings K:

$$\begin{aligned} Q^{s'} &= \texttt{transDec(}Q^s, f^{fuse'}\texttt{)}, \\ K &= \texttt{mlp(}Q^{s'}\texttt{)} \in \mathbb{R}^{M \times C}. \end{aligned} \quad (5)$$

*Equal contribution.
†Corresponding author.

Table 1. Acknowledgement for used code assets in this work.

URL	Version	Licence
https://github.com/open-mmlab/OpenPCDet	a9c66fe	Apache-2.0
https://github.com/AvivNavon/nash-mtl	6467e30	NA
https://github.com/mit-han-lab/bevfusion	0e5b9ed	Apache-2.0
https://github.com/ADLab-AutoDrive/BEVFusion	be0cb2e	Apache-2.0
https://github.com/facebookresearch/MaskFormer	da3e60d	MIT license

Table 2. Comparison to more methods.

	Modality	VoxelSize	LiDAR	Image	mAP(%)↑	NDS↑	mIoU(%)↑
			3D Detection				
M^2BEV [20]	C	-	-	ResNeXt-101 [21]	41.7	47.0	-
BEVFormer [9]	C	-	-	ResNet101 [6]	41.6	51.7	-
PointPillars‡ [8]	L	0.075	PointPillars	-	52.3	61.3	-
CenterPoint [23]	L	0.075	VoxelNet	-	59.6	66.8	-
PointPainting‡ [19]	C+L	0.075	PointPillars	-	65.8	69.6	-
MVP‡ [24]	C+L	0.075	VoxelNet	DLA-34	66.1	70.0	-
FusionPainting [22]	C+L	-	Cylinder3D [26]	HTCNet [2]	66.5	70.7	-
AutoAlign [5]	C+L	0.075	CenterPoint	ResNet-50	66.6	71.1	-
FUTR3D [3]	C+L	0.075	VoxelNet	ResNet-101	64.5	68.3	-
TransFusion [1]	C+L	0.075	VoxelNet	DLA-34	67.5	71.3	-
BEVFusion [14]	C+L	0.075	VoxelNet	Swin-T	68.5	71.4	-
Fuller-det	C+L	0.075	VoxelNet	Swin-T	**67.6**	**71.3**	-
Fuller-det (upper bound)	C+L	0.1	VoxelNet	Swin-T	**62.1**	**66.6**	-
			BEV Map Segmentation				
OFT‡ [16]	C	-	-	ResNet-18	-	-	42.1
LSS‡ [15]	C	-	-	EfficientNet-B0 [17]	-	-	44.4
PointPillars‡ [8]	L	0.1	-	PointPillars	-	-	43.8
CenterPoint‡ [23]	L	0.1	VoxelNet	-	-	-	48.6
PointPainting‡ [19]	C+L	0.1	PointPillars	-	-	-	49.1
MVP‡ [24]	C+L	0.1	VoxelNet	DLA-34[25]	-	-	49.0
BEVFusion [14]	C+L	0.1	VoxelNet	Swin-T	-	-	62.7
Fuller-seg(upper bound)	C+L	0.1	VoxelNet	Swin-T	-	-	**62.3**
			3D Detection + BEV Map Segmentation				
BEVFusion† [14] (share)	C+L	0.1	VoxelNet	Swin-T	-	69.7	54.0
BEVFusion† [14] (sep)	C+L	0.1	VoxelNet	Swin-T	-	69.9	58.4
Baseline(share)	C+L	0.1	VoxelNet	Swin-T	59.1	65.0	44.0
Fuller(share)	C+L	0.1	VoxelNet	Swin-T	**60.5**	**65.3**	**58.4**

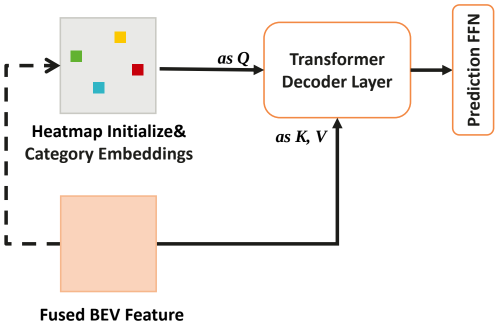
Finally, mask prediction S is obtained via a dot product between K and $f^{fuse'}$, followed by a sigmoid activation.

$$S = \texttt{sigmoid}(<f^{fuse'}, K>) \in \mathbb{R}^{M \times H \times W}. \quad (6)$$

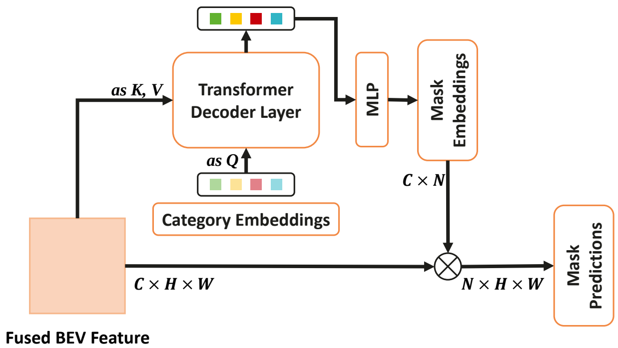
Here $<\cdot, \cdot>$ is the dot-product operator. The loss function for segmentation head is the standard focal loss [10] with mean reduction.

Data Pre-processing. Following BEVFusion [14], the image resolution is downsampled from 900×1600 to 256×704. The perception range of point cloud is truncated to $[-51.2m, 51.2m]$ for X and Y axes, and $[-5m, 3m]$ for Z axis, which is further voxelized with the size of $(0.1m, 0.1m, 0.2m)$. The maximum numbers of non-empty voxels for both training and inference are set to 60,000. For point cloud augmentation, we apply randomly global rotation between $[-\pi/8, \pi/8]$, global scaling with a random factor from $[0.95, 1.05]$, and randomly global translation along X and Y axis within $0.2m$. For image augmentation, we adopt random scaling with the factor $[0.94, 1.11]$, random rotation between $[-5.4°, 5.4°]$, and random flip along H and W.

Backbone Training. The LiDAR backbone (VoxelNet [27]) is initialized with the pretrained weight from Transfusion-L [1] while the image backbone (Swin-T [13]) is pretrained on ImageNet [7]. Without branch frozen or trained in advance, the whole network is trained in an end-to-end fashion. The reported performance is validated without test-time augmentation.



(a) Detection Head



(b) Segmentation Head

Figure 1. Illustration of the task head.

C. More Experimental Results

C.1. Comparison to Benchmark

We compared the Fuller with existing literature in the Tab. 1 of the main text. We compare it with more methods in Tab. 2. The table contains single-modality, multi-modality, single-task, and multi-task methods.

C.2. Optimization of Multi-task Learning

Regarding multi-task optimization, we evaluate several classic methods [12, 4, 11], as shown in Tab. 5. With distinct natures, they improve the model differently. Note that DWA [12] needs to empirically select the initial loss weights, *i.e.*, det:seg=1:10 in our example, while GradNorm [4] needs extra learnable parameters. Considering the scalability, we resort IMTL_G [11] as the technique for inter-gradient calibration, which demonstrates significant improvement in map segmentation and the comparable Δ_{MTL} with GradNorm.

C.3. Improvement upon MTL Baseline

Regarding the evaluation metric of multi-task learning, we presented the metric Δ_{MTL} in the main text based on [18], which is intuitively understood as the average performance drop compared to the single-task upper bound. Here we introduce another metric that measures the performance improvement compared to the multi-task baseline:

Table 3. In addition to the average performance drop Δ_{MTL} compared to upper bounds, we also list Fuller's average performance improvement Λ_{MTL} compared to the multi-task baseline. Here we list three loss weights ratios between detection task and segmentation task.

Method	weight ratio	mAP(%)↑	NDS(%)↑	mIoU(%)↑	Δ_{MTL}(%)↓	Λ_{MTL}(%)↑
Upper bounds	-	62.1	66.6	62.3	-	-
Baseline	1:1	59.1	65.0	44.0	18.3	-
Fuller		**60.5**	**65.3**	**58.4**	**5.4**	5.4
Baseline	1:5	59.8	65.5	55.7	8.0	-
Fuller		**60.1**	**65.6**	**58.2**	**5.7**	1.0
Baseline	1:10	59.3	65.0	57.9	14.0	-
Fuller		**59.9**	**65.2**	**59.2**	**5.3**	0.7

Table 4. Experiments with different image backbones.

	Image Backbone	LiDAR Pretrain	mAP(%)↑	NDS(%)↑	mIoU(%)↑
Baseline	EfficientNet-B0	TransFusion-L	**60.3**	**66.0**	43.8
Fuller			60.1	**66.0**	**51.6**
Baseline	ResNet-50	TransFusion-L	58.8	65.3	43.7
Fuller			**59.1**	**65.6**	**51.5**

Table 5. DWA needs to empirically select the initial loss weights while GradNorm uses extra learnable parameters. Considering scalability and performance, we select the IMTL_G as the inter-gradient calibration.

Method	mAP(%)↑	NDS↑	mIoU(%)↑	Δ_{MTL}(%)↓
Baseline	59.1	65.0	44.0	18.3
GradNorm [4]	58.3	64.4	57.2	8.8
DWA [12]	59.3	65.1	57.7	7.1
IMTL_G [11]	57.1	63.3	**59.4**	8.9
Fuller (Ours)	**60.5**	**65.3**	58.4	**5.4**

$$\Lambda_{\text{MTL}} = \frac{1}{T}\sum_{i=1}^{T}(M_{m,i} - N_{n,i}), \tag{7}$$

where T is the task number, $M_{b,i}$ and $N_{n,i}$ are the performance of i-th task of the evaluated method and the multi-task baseline, respectively. Specifically, we evaluate Fuller under three settings of loss weights. The result is shown in Tab. 3. As the loss weight of segmentation task is increased, the performance gap between the baseline and the upper bounds is narrowed, *i.e.*, lower Δ_{MTL}. Regardless the settings of loss weights, Fuller proves to be able to improve the baseline, as indicated by the Λ_{MTL} metric.

C.4. Ablation Study of Image Backbone

For the experiments of the main text, we empirically find that the pretrained weights of Transfuion_L [1] is favorable for LiDAR branch and Swin Transformer [13] is a strong backbone for image feature extraction. We report more ablation studies using different image backbones, as shown in Tab. 4. Generally, EfficientNet-B0 [17] is more advantageous than ResNet50 [6] for 3D detection. When compared

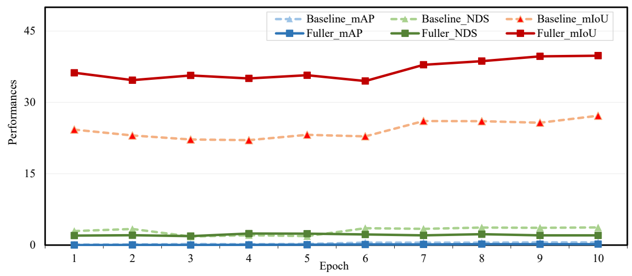


Figure 2. Evaluated task performances in absence of LiDAR scans.

Table 6. Comparison between task heads.

Method	mAP(%)↑	NDS↑	mIoU(%)↑	Params↓
CenterPoint[23]	58.6	64.8	-	1.6m
Fuller-det (Ours)	**62.1**	**66.6**	-	**1.0m**
BEVFusion[14]	-	-	**62.7**	4.7m
Fuller-seg (Ours)	-	-	62.3	**2.7m**

to the baseline, Fuller performs either superiorly or comparably in terms of 3D detection. Notably, Fuller surpasses the baseline by a large margin in the case of map segmentation.

C.5. Analysis of Modality Bias

In the main text, we conducted the experiment to inspect the modality bias, in which a model trained with both modalities is evaluated by dropping off the image input. We observed that in the absence of image input, 3D detection can be supported by LiDAR scan without losing too much performance. In contrast, the performance of segmentation task drops drastically without image input. Here we show another experiment that evaluates the trained model ***without*** LiDAR scan. As shown in Fig. 2, 3D detection suffers extremely that it does not even work properly while map segmentation has a relatively normal result. This phenomenon reveals the issue of modality bias that LiDAR dominates the detection performance and camera may work as an auxiliary modality for refinement. Additionally, our proposed gradient calibration significantly improve the performance of segmentation task. Improving detection performance in such difficult situation is deferred to furture work.

C.6. Comparison between task heads.

To demonstrate the introduced task heads, we compare the accuracy and parameter amount with widely-used detection and segmentation heads in Tab. 6. For 3D detection, Fuller-det surpasses CenterPoint[23] while saving 37.5% parameters. Similarly, for map segmentation, Fuller-seg achieves comparable result to BEVFusion[14] with 42.6% parameters reduction.

D. Limitation.

Currently, our analysis focuses on the concatenation fusion strategy. To generalize our model, we would like to investigate more fusion schemes in future. Besides, as shown in Fig. 2, Fuller still can be improved to deal with the situation of sensor failure in real-world scenarios.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022. 1, 2, 3

[2] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019. 2

[3] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. *arXiv preprint arXiv:2203.10642*, 2022. 2

[4] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803, 2018. 3

[5] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinghong Jiang, Feng Zhao, Bolei Zhou, and Hang Zhao. Autoalign: Pixel-instance feature aggregation for multi-modal 3d object detection. *arXiv preprint arXiv:2201.06493*, 2022. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognitions*, pages 770–778, 2016. 2, 3

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2

[8] 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*, pages 12697–12705, 2019. 2

[9] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proceedings of the European Conference on Computer Vision*, pages 1–18, 2022. 2

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[11] Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021. 3

[12] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019. 3

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 2, 3

[14] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 1, 2, 4

[15] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proceedings of European Conference on Computer Vision*, pages 194–210, 2020. 1, 2

[16] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018. 2

[17] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2, 3

[18] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 3614–3633, 2021. 3

[19] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4604–4612, 2020. 2

[20] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. Mˆ 2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 2

[21] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. 2

[22] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Zhou Bin, and Liangjun Zhang. Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3047–3054. IEEE, 2021. 2

[23] 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*, pages 11784–11793, 2021. 2, 4

[24] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. *Advances in Neural Information Processing Systems*, 34:16494–16507, 2021. 2

[25] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018. 2

[26] Hui Zhou, Xinge Zhu, Xiao Song, Yuexin Ma, Zhe Wang, Hongsheng Li, and Dahua Lin. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv preprint arXiv:2008.01550*, 2020. 2

[27] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 1, 2