DenseBEV: Transforming BEV Grid Cells into 3D Objects Supplementary Material

Marius Dähling[1,2,*] Sebastian Krebs[2,3] J. Marius Zöllner[1,4]

A. Supplementary

A.1. Class awareness in NMS

As stated in the paper, we refrain from using class information when applying Non-Maximum Suppression (NMS), as early misclassifications can adversely affect performance. We investigate this by evaluating DenseBEV with $\tau = 0.2$ on the nuScenes validation set, as shown in Table A1. We tested whether class-aware NMS or Scale NMS [1] could enhance the detection task. Scale NMS resizes objects before calculating the Intersection-over-Union (IoU), making it less intrusive than class-aware NMS. It aims to better handle smaller objects, which our method already excels at. The results indicate that using Scale NMS results in no significant performance difference compared to the baseline (no class-aware NMS & no Scale NMS). Class-aware NMS, however, leads to a substantial performance drop, with a decrease of 4.2% in NDS and 5.6% in mAP. Pairing class-aware NMS with Scale NMS does not significantly alter the results. Hence, we decided to use regular NMS without any modifications.

Class-aware NMS	Scale NMS	**NDS** ↑	**mAP** ↑
✗	✗	0.535	0.433
✗	✓	0.535	0.430
✓	✗	0.493	0.377
✓	✓	0.495	0.375

Table A1. Quantitative analysis of DenseBEV on nuScenes val using different NMS methods.

A.2. Performance on Large Object Detections

We observe that using solely dense queries leads to a slight performance drop for larger objects. To better understand this behavior, we investigate the training dynamics of the model for these classes. For completeness, Table 3 of the main paper reports final performance across all classes.

Class	Epoch	BEVFormer mAP ↑	DenseBEV mAP ↑
Truck	6	30.1	33.1
	16	34.1	35.9
	24	37.0	36.0
Car	6	55.0	59.4
	16	58.8	61.7
	24	61.8	61.3

Table A2. Class-specific performance of BEVFormer and DenseBEV (both base models) across training epochs for a selection of large object classes. The total number of training epochs was 24.

In Table A2 we show a selection of large object classes and the respective model performance at different epochs. The results indicate that dense queries accelerate convergence for large objects such as trucks and cars, particularly in the early epochs, where DenseBEV achieves a 4.4% improvement for cars and a 3.0% improvement for trucks in mAP. As training progresses, however, this difference diminishes, and BEVFormer ultimately achieves better results.

These findings indicate that, despite NMS, multiple dense cells may continue to compete for responsibility when handling large objects.

A.3. Extended Efficiency Analysis

As computation may be limited in applications such as edge devices, we provide a more detailed overview of the runtime of our approach and potential strategies to mitigate overhead. The different options are summarized in Table A3.

In runtime-critical scenarios where peak performance is not strictly required, the most effective alternative is the use of DenseBEV++-small. The model outperforms BEVFormer-Base by more than one point in both NDS and mAP, while achieving around 50% higher inference speed.

Since the overhead primarily stems from the large number of candidates in the first NMS stage before the decoder, reducing the set at this stage is an effective strategy. We evaluated two approaches: (i) top-k candidate selection and

* Corresponding author: marius.daehling@mercedes-benz.com
[1]Karlsruhe Institute of Technology (KIT)
[2]Mercedes-Benz AG, Research and Development
[3]Intelligent Vehicles Group at TU Delft
[4]Research Center for Information Technology (FZI)

Method	**FPS** ↑	**NDS** ↑	**mAP** ↑
BEVFormer-base	1.637	51.7	41.6
DenseBEV++-base	1.327	54.9	44.9
DenseBEV++-small	2.449	52.8	42.8
DenseBEV++-base (top-10k)	1.519	54.9	44.9
DenseBEV++-base (conf. 0.5%)	1.511	54.9	44.9
DenseBEV++-base (conf. 1%)	1.545	54.7	44.7

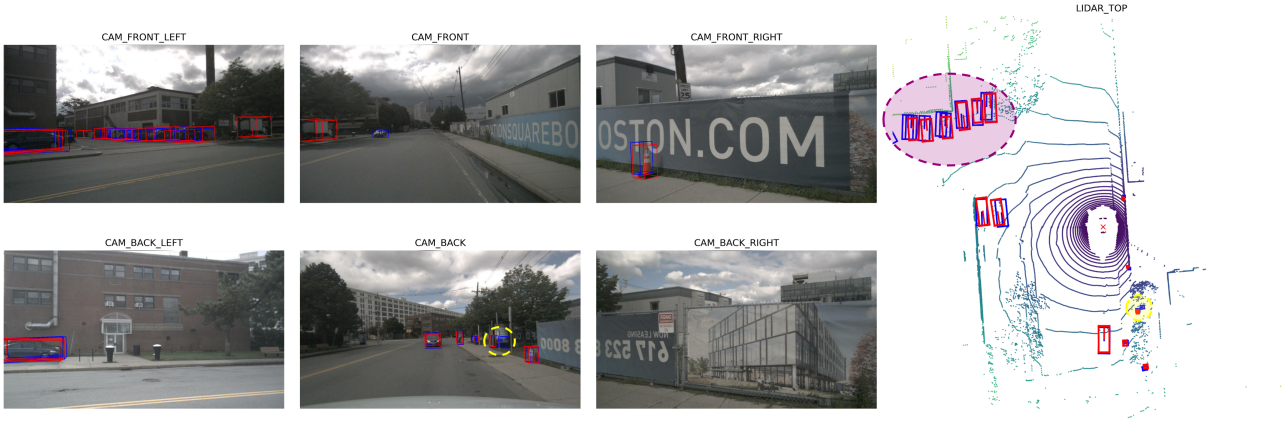
Table A3. Computation comparison measured on a single NVIDIA 2080Ti using a batch size of 1 of NuScenes data. The metrics are evaluated on NuScenes val.

(ii) applying a confidence-based threshold. As shown in Table A3, both methods significantly reduce runtime overhead while mostly maintaining performance. Importantly, this filtering is applied only during inference, as the model at this stage has already learned to differentiate background from objects. While we prefer confidence-based thresholding due to its adaptive nature, both parameters can be manually tuned after training depending on the target application. We note, however, that the thresholding experiments were conducted solely on nuScenes, and care should be taken when applying them to different datasets or deployment settings.

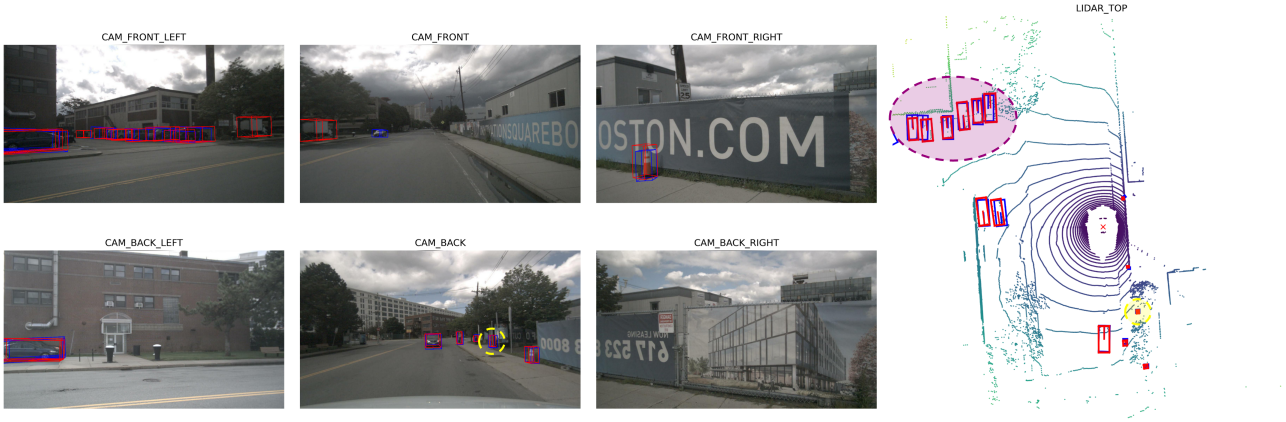
A.4. Additional Visualizations

In this section, we provide additional visualizations to highlight the impact of our method. Figure A1 shows a qualitative comparison between the baseline BEVFormer [2] and our proposed DenseBEV++. The yellow circle in Figure A1a highlights duplicate detections made by the baseline, which are effectively suppressed by our integrated NMS, as shown in Figure A1b. The purple-highlighted area showcases how our dense anchor strategy improves localization, as demonstrated by the closer alignment between predicted and ground-truth boxes in the Bird's-Eye-View (BEV).

We provide two additional examples in Figures A2 and A3, without discussing them in detail. The key differences are highlighted with circles. Overall, the baseline produces more false positives, especially duplicates in regions with many small objects. In contrast, *DenseBEV++* yields fewer false positives, is more reliable, and detects a greater number of small objects.

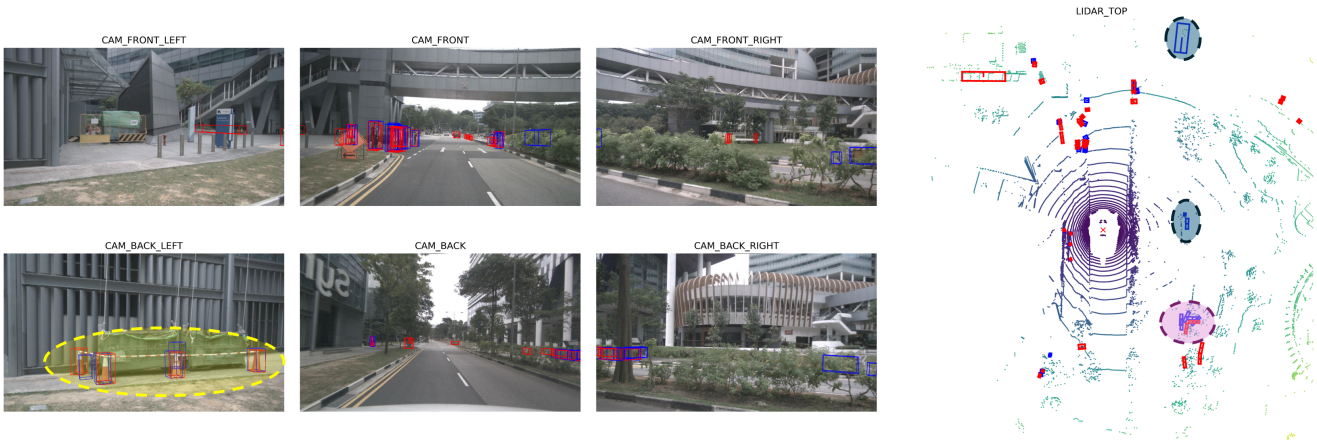


(a) Visualization of baseline (BEVFormer) results.

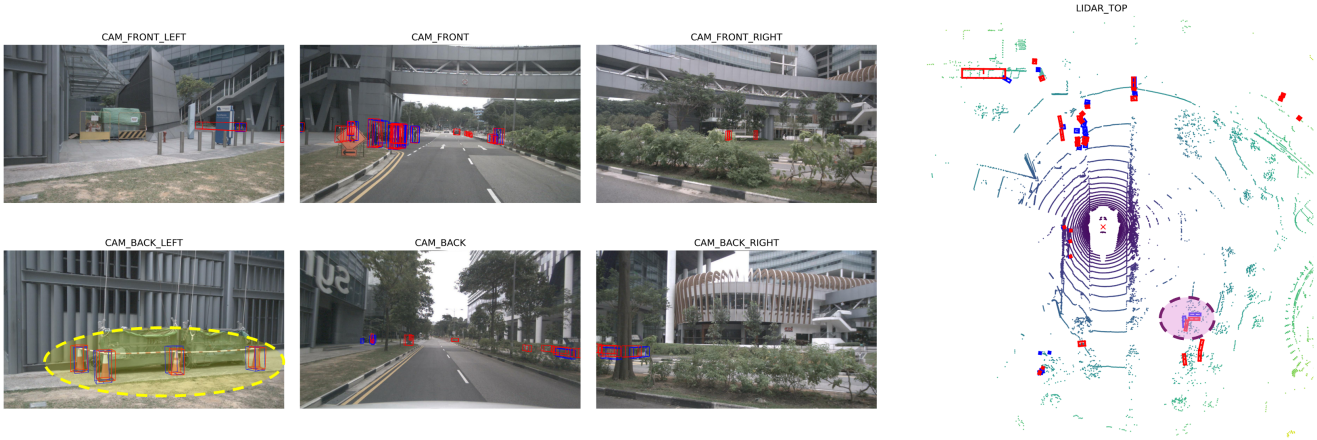


(b) Visualization of DenseBEV++ results.

Figure A1. Comparison of detection results: (a) baseline and (b) DenseBEV++. The red boxes mark the ground truth, and blue boxes the predictions. The circles highlight interesting areas in the scene.

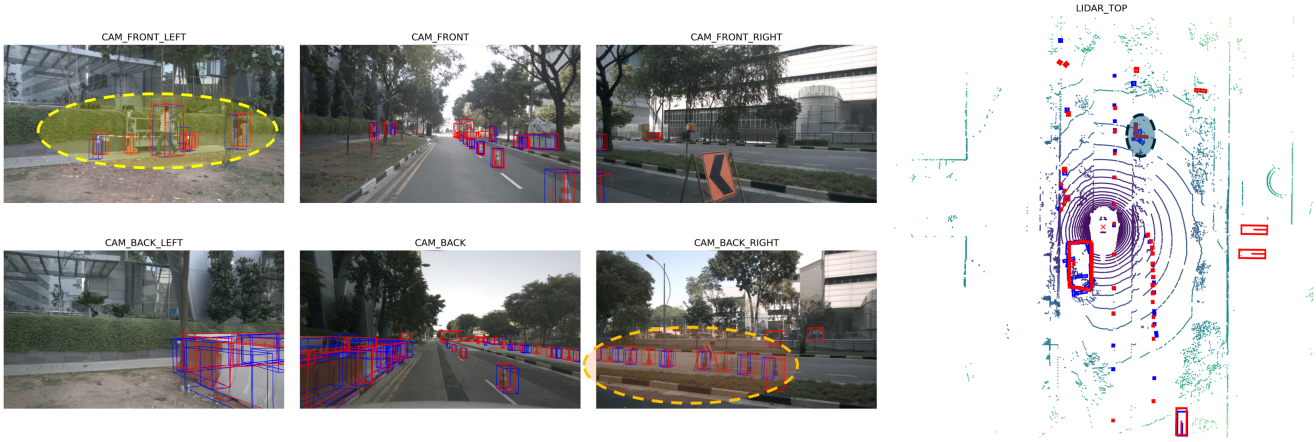


(a) Visualization of baseline (BEVFormer) results.

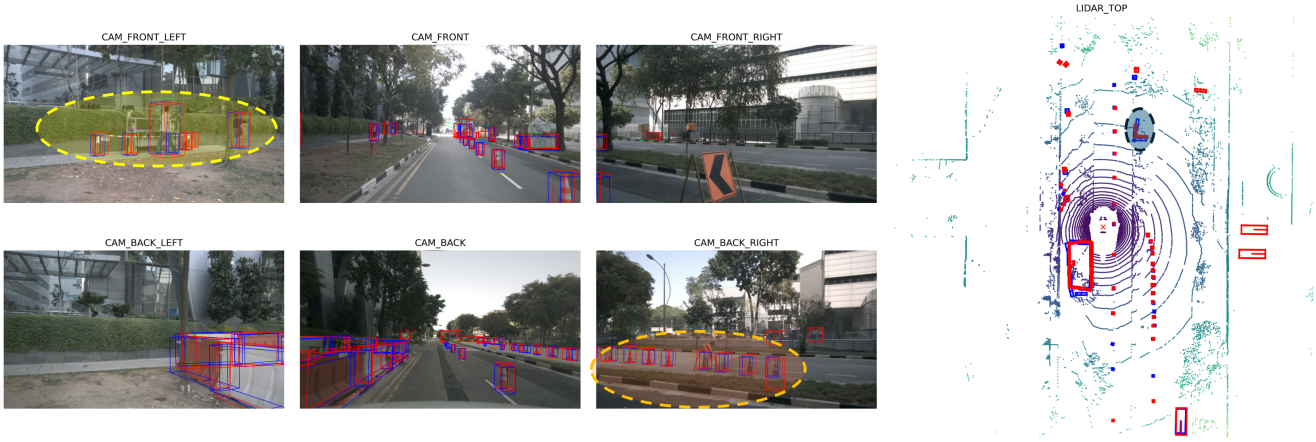


(b) Visualization of DenseBEV++ results.

Figure A2. Comparison of detection results: (a) baseline and (b) DenseBEV++. The red boxes mark the ground truth, and blue boxes the predictions. The circles highlight interesting areas in the scene.



(a) Visualization of baseline (BEVFormer) results.



(b) Visualization of DenseBEV++ results.

Figure A3. Comparison of detection results: (a) baseline and (b) DenseBEV++. The red boxes mark the ground truth, and blue boxes the predictions. The circles highlight interesting areas in the scene.

References

[1] Junjie Huang, Guan Huang, Zheng Zhu, Ye Yun, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1

[2] 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 *Computer Vision – ECCV 2022*, pages 1–18, 2022. 2