# Supplementary Material for Are We Ready for Vision-Centric Driving Streaming Perception? The ASAP Benchmark

In the supplement materials, we first elaborate on the implementation details in the ASAP benchmark. Subsequently, we provide additional baseline results for a more thorough evaluation. Finally, visualizations of the extended 12Hz nuScenes-H dataset are given.

## 1. Additional Implementation Details

### 1.1. Streaming Simulation

As described in Sec. 3.1 (main text), to evaluate the predictions $\hat{Y}$ at input timestamp $t_i$, the ground truth $Y_i$ is desired to match with the most recent prediction, yielding the pair $(Y_i, \hat{Y}_{\theta(i)})$, where $\theta(i) = \arg\max_j t_j < t_i$. The input time $\{t_i\}_{i=1}^{T}$ is a 12Hz sequence, but the output time $\{t_j\}_{j=1}^{M}$ of each prediction is associated with the model runtime on specific hardware. To determine the output timestamps, the streaming evaluation is conducted with a hardware-dependent simulator [12]. Specifically, we run the algorithm over the entire nuScenes [1], and measure the inference time of the algorithm on a specific GPU (the runtime distribution of BEVFormer [14] on NVIDIA RTX3090 is shown in Fig. 1). Then we can randomly sample model runtime from the time distribution, to calculate the output timestamps $\{t_j\}_{j=1}^{M}$ in the simulation.

### 1.2. Streaming Evaluation Details

In the ASAP benchmark, we analyze the streaming performance of seven modern 3D detectors. We use their open-sourced code and pretrained model (BEVDet-Tiny [2], BEVDet4D-Tiny [4], BEVFormer-Base [5], BEVDepth-R50 [3], PETR-R50 [8], FCOS3D-R101 [6], PGD-R101 [7]) to generate detection results from the 12Hz streaming inputs. Notably, for multi-frame methods (*e.g.*, BEVFormer [14], BEVDepth [13]) that use sequential frames as input, we set the input-frame-interval as six (instead of one in the original 2Hz input configuration). Such a strategy maintains the input-timestamp-interval as 0.5s, which guarantees sufficient *Triangulation Priority* [16] for 3D perception. As shown in Tab. 1, for BEVFormer and BEVDepth, the proposed configuration (input frequency (I.F.)=12Hz, input-frame-interval (I.F.I)=6) significantly outperforms the original setting (I.F.=12Hz, I.F.I=1), and the corresponding metrics (mAP, ATE, ASE, AOE, AVE, AAE) are comparable to those of the 2Hz result (I.F.=2Hz, I.F.I=1).

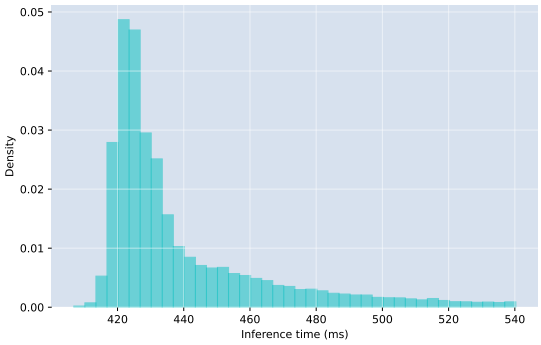


Figure 1. Inference time distribution for BEVFormer [14] (Backbone: ResNet101 [9], Input size: 1600 × 900) on NVIDIA RTX3090.

Table 1. Offline performance of BEVFormer [14] and BEVDepth [13] on the nuScenes (I.F.=2Hz) and nuScenes-H (I.F.=12Hz) , where I.F. represents the input frequency, and I.F.I denotes the input-frame-interval.

| Method | I.F (Hz) | I.F.I | **mAP**↑ | ATE↓ | ASE↓ | AOE↓ | AVE↓ | AAE↓ |
|---|---|---|---|---|---|---|---|---|
| BEVFormer | 2 | 1 | 0.415 | 0.672 | 0.274 | 0.369 | 0.397 | 0.198 |
| BEVFormer | 12 | 1 | 0.341 | 0.769 | 0.279 | 0.400 | 0.699 | 0.203 |
| BEVFormer | 12 | 6 | 0.410 | 0.691 | 0.274 | 0.376 | 0.401 | 0.197 |
| BEVDepth | 2 | 1 | 0.348 | 0.616 | 0.272 | 0.415 | 0.440 | 0.196 |
| BEVDepth | 12 | 1 | 0.311 | 0.640 | 0.274 | 0.470 | 0.893 | 0.209 |
| BEVDepth | 12 | 6 | 0.341 | 0.622 | 0.273 | 0.412 | 0.453 | 0.193 |

## 2. Additional Baseline Results

In this section, we provide additional experiment results of the velocity-based updating baseline. As shown in Tab. 2, the proposed baselines built upon [10, 11, 14, 15, 17, 18] consistently enhance the streaming performance, suggesting that the velocity-based updating baseline can

Table 2. Streaming performance (mAP-S) of FCOS3D [17], PGD [18], BEVFormer [14], BEVDet [11], BEVDet4D [10], PETR [15] and the corresponding velocity-based updating baselines. The experiments are conducted on RTX3090.

| Method | **mAP-S**↑ | ATE-S↓ | ASE-S↓ | AOE-S↓ | AAE-S↓ |
|---|---|---|---|---|---|
| FCOS3D | 0.208 | 0.828 | 0.268 | 0.511 | 0.170 |
| FCOS3D-Sv | 0.218 (+4.8%) | 0.820 | 0.267 | 0.506 | 0.169 |
| PGD | 0.206 | 0.817 | 0.273 | 0.488 | 0.185 |
| PGD-Sv | 0.217 (+5.3%) | 0.813 | 0.273 | 0.485 | 0.183 |
| BEVFormer | 0.310 | 0.760 | 0.276 | 0.385 | 0.216 |
| BEVFormer-Sv | 0.344 (+10.9%) | 0.748 | 0.274 | 0.382 | 0.208 |
| BEVDet | 0.289 | 0.730 | 0.273 | 0.533 | 0.209 |
| BEVDet-Sv | 0.291 (+0.7%) | 0.728 | 0.273 | 0.532 | 0.207 |
| BEVDet4D | 0.309 | 0.755 | 0.275 | 0.480 | 0.200 |
| BEVDet4D-Sv | 0.316 (+2.3%) | 0.750 | 0.274 | 0.476 | 0.198 |
| PETR | 0.282 | 0.883 | 0.288 | 0.639 | 0.249 |
| PETR-Sv | 0.291 (+3.2%) | 0.880 | 0.287 | 0.636 | 0.247 |

Table 3. Ablation study of the velocity-based updating baseline, where *C.V.* represents the *constant velocity motion model*, and *K.F.* denotes the Kalman filter refinement. The streaming evaluation is conducted on RTX3090.

| Methods | C.V. | K.F. | mAP-S ↑ | NDS-S↑ | ATE-S ↓ | AOE-S ↓ |
|---|---|---|---|---|---|---|
| BEVFormer | | | 0.310 | 0.452 | 0.760 | 0.385 |
| BEVFormer | ✓ | | 0.332 | 0.460 | 0.756 | 0.384 |
| BEVFormer | ✓ | ✓ | 0.344 | 0.465 | 0.748 | 0.382 |
| FCOS3D | | | 0.208 | 0.326 | 0.828 | 0.512 |
| FCOS3D | ✓ | | 0.212 | 0.329 | 0.823 | 0.509 |
| FCOS3D | ✓ | ✓ | 0.218 | 0.332 | 0.820 | 0.506 |

compensate for the inference delay. Note that BEVDet-Sv and BEVDet4D-Sv obtain relatively lower improvements than other methods, as they suffer little from the influence of inference delay. Namely, the model speed of BEVDet@RTX3090 and BEVDet4D@RTX3090 are ∼12Hz, which is close to the input frame rate.

Besides, we conduct ablation study to validate the effectiveness of the Kalman filter refinement. As shown in Tab. 3, FCOS3D [17] and BEVFormer [14] relatively improve the mAP-S by 4.8% and 7.1% using the *constant velocity motion model*. Notably, the Kalman filter further boosts the mAP-S (11.0%@BEVFormer and 9.1%@FCOS3D), which indicates that multi-frame association and state refinement can benefit the streaming perception.

## 3. Visualizations

As depicted in Fig. 2, we visualize the 12Hz annotations of nuScenes-H (more visualization comparison between the 2Hz nuScenes and 12Hz nuScenes-H can be found in the uploaded video files). Besides, the streaming detection results are shown in Fig. 3, where the predicted bounding boxes are displaced from the object locations, especially for the high-speed vehicles.

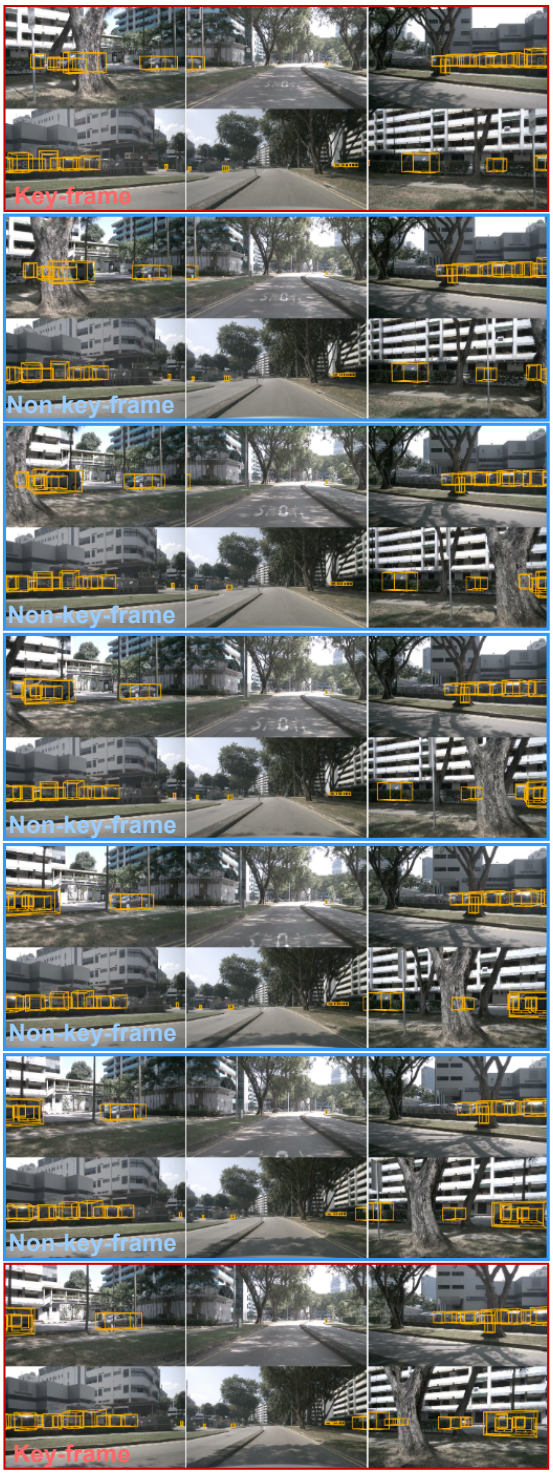


Figure 2. Visualization of the surround-view annotation in nuScenes-H, where the key-frames are the 2Hz images in the original nuScenes dataset [1], and the intermediate non-key-frames are the annotated 12Hz images.

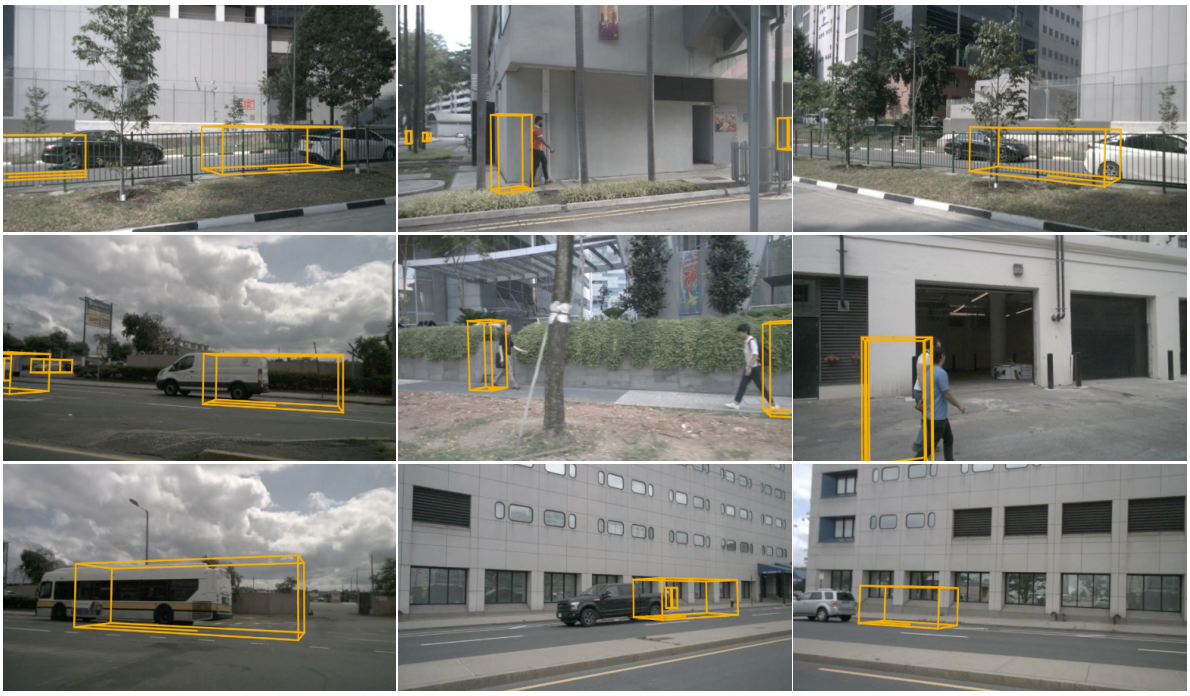

Figure 3. Visualization of the streaming perception results, where the predicted bounding boxes are displaced from the moving objects (*e.g.*, car, pedestrian).

## References


[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CVPR*, 2019. 1, 2

[2] BEVDet Contributors. Release code for bevdet. https://github.com/HuangJunJie2017/BEVDet/blob/master/configs/bevdet/bevdet-sttiny.py, 2021. 1

[3] BEVDepth Contributors. Release code for bevdepth. https://github.com/Megvii-BaseDetection/BEVDepth/blob/main/exps/mv/bev_depth_lss_r50_256x704_128x128_20e_cbgs_2key_da.py, 2022. 1

[4] BEVDet4D Contributors. Release code for bevdet4d. https://github.com/HuangJunJie2017/BEVDet/blob/master/configs/bevdet4d/bevdet4d-sttiny.py, 2022. 1

[5] BEVFormer Contributors. Release code for bevformer. https://github.com/fundamentalvision/BEVFormer/blob/master/projects/configs/bevformer/bevformer_base.py, 2022. 1

[6] MMDetection3D Contributors. Release code for fcos3d. https://github.com/open-mmlab/mmdetection3d/blob/master/configs/fcos3d/fcos3d_r101_caffe_fpn_gn-head_dcn_2x8_1x_nus-mono3d.py, 2022. 1

[7] MMDetection3D Contributors. Release code for pgd. https://github.com/open-mmlab/mmdetection3d/blob/master/configs/pgd/pgd_r101_caffe_fpn_gn-head_2x16_2x_nus-mono3d.py, 2022. 1

[8] PETR Contributors. Release code for petr. https://github.com/megvii-research/PETR/blob/main/projects/configs/petr/petr_r50dcn_gridmask_p4.py, 2022. 1

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[10] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv: 2203.17054*, 2022. 1, 2

[11] 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

[12] Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. Towards streaming perception. *ECCV*, 2020. 1

[13] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv: 2206.10092*, 2022. 1

[14] 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. *ECCV*, 2022. 1, 2

[15] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *ECCV*, 2022. 1, 2

[16] Johannes L. Schonberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1

[17] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3d object detection. In *ICCV Workshops*, 2021. 1, 2

[18] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and Geometric Depth: Detecting objects in perspective. In *CoRL*, 2021. 1, 2