Supplementary Material for SparseBEV

Haisong Liu [1] Yao Teng [1] Tao Lu [1] Haiguang Wang [1] Limin Wang [1,2,✉]
[1]State Key Laboratory for Novel Software Technology, Nanjing University [2]Shanghai AI Lab
{liuhs, yaoteng, taolu, haiguangwang}@smail.nju.edu.cn, lmwang@nju.edu.cn

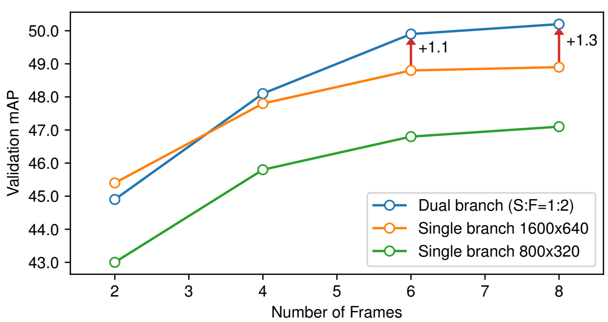


Figure 1: Comparison between single-branch and dual-branch under different settings. Dual branch design brings more gain as the number of frames increases.

A. Details of Dual-branch SparseBEV

In this section, we provide detailed explanations and ablations on the dual branch design. As shown in Fig. 2, the input multi-camera videos are divided into a high-resolution "slow" stream and a low-resolution "fast" stream. Sampling points are projected to the two streams respectively and the sampled features are stacked before adaptive mixing. Experiments are conducted with a V2-99 [1] backbone pre-trained by FCOS3D [2] on the training set of nuScenes. [1]

In Fig. 1, we compare our dual branch design with single branch baselines. If we use a single branch of 1600 × 640 (orange curve) resolution, adding more frames does not provide as much benefit as it does at 800 × 320 resolution (green curve). By using dual branch of 1600 × 640 and 640 × 256 resolution with 1:2 ratio, we decouple spatial appearance and temporal motion, unlocking better performance. As we can see from the blue curve, the longer the frame sequence, the more gain the dual branch design brings.

In Tab. 1, we provide detailed quanlitative results. Under the setting of 8 frames (∼ 4 seconds), our dual branch design with only *two* high resolution (HR) frames surpasses the baseline with *eight* HR frames. By increasing the number of HR frames to 4, we further improve the performance by 0.8 mAP and 0.5 NDS. Moreover, increasing the resolution of the LR frames does not bring any improvement, which clearly demonstrates that appearance detail and temporal motion are decoupled to different branches.

Method	Setting	mAP	NDS
Single branch	8f × 1600	48.9	57.3
Dual branch	2f × 1600 + 8f × 640	49.4	57.9
Dual branch	4f × 1600 + 8f × 640	**50.2**	**58.4**
Dual branch	4f × 1600 + 8f × 800	50.1	58.0

Table 1: Ablations on the dual branch design. Nf × M indices the number of frames is N and the longer side of the image has M pixels. For example, "8f × 640" denotes 8 frames with 640 × 256 resolution.

Method	Feature Maps	Train. Cost	mAP
Single branch	C_2, C_3, C_4, C_5	2d 17h	48.9
Single branch	C_2, C_3, C_4, C_5, C_6	2d 18h	49.3
Dual branch	C_2, C_3, C_4, C_5	**1d 19h**	**50.2**

Table 2: Detailed analyses on the dual-branch design. For single branch baselines, simply adding an extra C_6 feature map has limited effect. In contrast, our dual branch design can boost the performance significantly.

Since the dual-branch design also enlarges the receptive field (smaller resolution provides larger receptive field) which may improve performance, we further analyse where the improvement comes from in Tab. 2. The first row is our baseline which takes 8 frames with a single branch of 1600 × 640 as input. We first try to increase the receptive field by adding an extra C_6 feature map (Row 2), and observe that the performance is slightly improved. This demonstrates that a larger receptive field is required for high-resolution and long-term inputs. However, the spatial appearance and temporal motion is still coupling, limiting the performance. By using dual branches of 1600 × 640

✉: Corresponding author.

[1]Note that the experiment setting used here is different from that in the main paper, since the experiments are conducted before the submission of ICCV 2023. After submission, we further improve our implementation to refesh our results. The conclusion is consistent between these different implmentations.

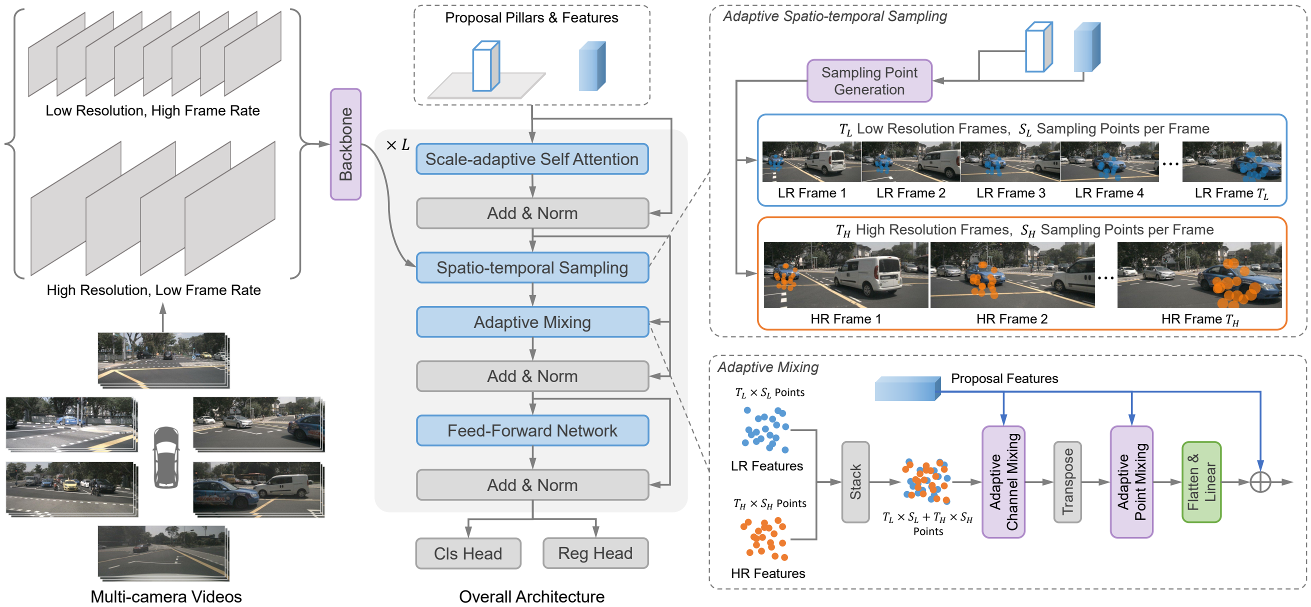


Figure 2: Architecture of dual-branch SparseBEV. The input multi-camera videos are divided into a high-resolution "slow" stream and a low-resolution "fast" stream.

Self Attention	Distance Function	NDS	mAP
SASA-beta	τD	55.2	44.8
SASA	τD	**55.6**	**45.4**

Table 3: Compared with SASA-beta, SASA not only has the ability of multi-scale feature aggregation, but generates adaptive receptive field for each query as well.

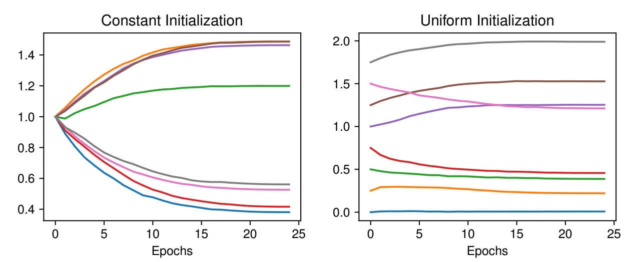


Figure 3: The change of τ of each head in SASA-beta during training. Regardless of the initialization, each head learns a different τ, enabling local and multi-scale feature aggregation.

and 640 $\times$ 256 with 1:2 ratio (Row 3), we decouple spatial appearance and temporal motion, leading to better performance. Moreover, the training cost is also reduced by 1/3. This experiment demonstrates that we not only need larger receptive fields, but also decouple spatial appearance and temporal motion.

B. Study on Scale-adaptive Self Attention

In this section, we'll talk about how we came up with scale-adaptive self attention (SASA). In the main paper, the receptive field coefficient τ is specific to each head and adaptive to each query. In the development of SASA, there is an intermediate version (dubbed SASA-beta for convenience): the τ for each head is simply a learnable parameter shared by all queries.

In Fig. 3, we take a closer look at how τ changes with training. We surprisingly find that regardless of the initialization, each head learns a different τ from the others and all of them are distributed in range $[0, 2]$, enabling the network to aggregate local and multi-scale features from multiple heads.

Next, we improve SASA-beta by generating the τ adaptively from the query, which corresponds to the version in the main paper. Compared with SASA-beta, SASA not only has the ability of multi-scale feature aggregation, but generates adaptive receptive field for each query as well. The quanlitative comparison between SASA-beta and SASA is shown in Tab. 3.

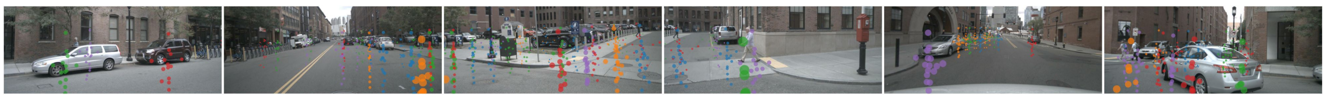
C. More Visualizations

In Fig. 4, we provide more visualizations of the sampling points from different stages. In the initial stage, the sampling points have the shape of pillars. In later stages, they are refined to cover objects with different sizes.

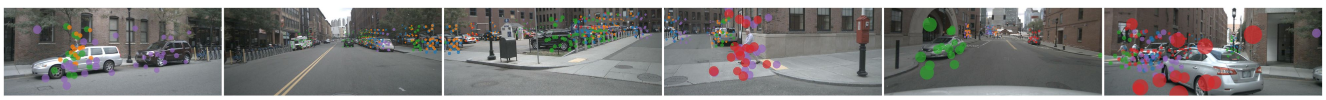
(a) Sample 0005, stage 1

(b) Sample 0005, stage 2

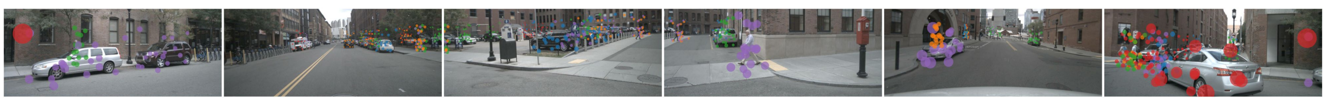
(c) Sample 0005, stage 3



(d) Sample 0028, stage 1



(e) Sample 0028, stage 2



(f) Sample 0028, stage 3

Figure 4: Visualized sampling points from different stages. Different instances are distinguished by colors.

References

[1] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 13906–13915, 2020. 1

[2] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 913–922, 2021. 1