Supplementary Material of ToC3D

Dingyuan Zhang[1*], Dingkang Liang[1*], Zichang Tan[2], Xiaoqing Ye[2], Cheng Zhang[1], Jingdong Wang[2], and Xiang Bai[1✉]

[1] Huazhong University of Science and Technology, Wuhan, China
{dyzhang233, dkliang, xbai}@hust.edu.cn
[2] Baidu Inc., Beijing, China

1 Overview

The supplementary material is organized as follows:

- In Sec. 2, we introduce more of our method's implementation details, including the implementation details of Sparse4Dv2 [2] version.
- In Sec. 3, we report more experiment results, including the ablation study of the location of importance score updating, the performance on the nuScenes test set and the Waymo Open Dataset [4]. This section shows more about the properties of our model.
- In Sec. 5, we illustrate more visualizations of our method for an intuitive understanding, which shows the effectiveness of our method in various scenes.

2 More Implementation Details

Table 1: Details of settings using StreamPETR [5] as the basic pipeline.

Configurations	ToC3D-Fast		ToC3D-Faster	
Backbone	ViT-B	ViT-L	ViT-B	ViT-L
Num. of attention heads	12	16	12	16
Patch size	16	16	16	16
Window size of window attention layer	14	16	14	16
Window size of global attention layer	-	20	-	20
Drop path rate	0.0	0.3	0.0	0.3
Weight decay	0.01	0.01	0.01	0.01
Grad clip	35	35	35	35
Num. of warmup iterations	500	500	500	500

We report more detailed implementation settings in Tab. 1, including the detailed architecture of ViTs and more training hyper-parameters.

To prove the generalization of our method, we evaluate our method on the Sparse4Dv2 [2]. The detailed settings are listed in Tab. 2, and we only report settings different from that of the StreamPETR version for simplicity. For data augmentation, we follow the official settings.

* Dingyuan Zhang and Dingkang Liang contributed equally. ✉ Corresponding author.

Table 2: Details of settings using Sparse4Dv2 [2] as basic pipeline.

Configurations	ToC3D-Fast	ToC3D-Faster
Backbone	ViT-L	ViT-L
Keeping ratios ρ	0.7, 0.5, 0.5	0.5, 0.4, 0.3
Token compression loss weight	1.0	1.0
Learning rate of backbone	2.5e-5	2.5e-5
Weight decay	0.001	0.001
Grad clip	25	25

Table 3: Effect of location of importance score updating on the nuScenes val set. We report the backbone inference time before the slash and the whole pipeline inference time after the slash.

Location	NDS(%)	mAP(%)	Infer. Time (ms)
StreamPETR [5]	61.2	52.1	290.0 / 317.0
4, 10, 16	60.2	50.9	205.0 (-29.3%)/ 233.3(-26.4%)
6, 12, 18	60.3	51.2	209.0(-28.0%) / 237.2(-25.2%)
7, 13, 19	60.3	51.1	217.7(-24.9%) / 248.7(-21.5%)
9, 15, 21	60.5	51.4	228.6(-21.2%) / 256.2(-19.2%)
10, 16, 22	61.0	52.2	235.8(-18.7%) / 264.0(-16.7%)

3 Experiment Results

3.1 Impact of Loc. of Importance Score S Updating

Since we only update the importance score S before specific transformer layers, we study the impact of these locations in this section. We conduct experiments using ToC3D-Faster with ViT-L backbone, and all models are trained for 12 epochs. The results are listed in Tab. 3.

We can see that the location of importance score updating affects the accuracy-speed tradeoff. When we conduct token compression in the deeper layers, the NDS and mAP are higher with the sacrifice of inference speed. It is worth noting that when updating the importance score at the 10th, 16th, and 22nd layers, the performance is competitive with the StreamPETR [5] baseline with about 17% acceleration. We empirically find that updating the importance score at the 6th, 12th, and 18th layers can achieve a better tradeoff, which is set by default.

3.2 Performance on Test Set

We report the performance of ToC3D on the nuScenes [1] test set in this section. We train all methods on the train and val set for 24 epochs, and then send the inference results of the test set to the official server for evaluation. We test with input resolution as 320 $\times$ 800.

Tab. 4 shows that the performance is consistent with that on nuScenes val set, *i.e.*, with a performance loss of no more than 0.3% NDS and 1.0% mAP, our method (ToC3D-Faster) brings 25% inference speed gains. Interestingly, our method can achieve comparable or even better performance when it comes to the

Table 4: We use StreamPETR [5] as our baseline and list the performance on the nuScenes test set. We report the backbone inference time before the slash and the whole pipeline inference time after the slash.

Method	Backbone	Resolution	NDS(%)↑	mAP(%)↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	Infer. Time (ms)↓
StreamPETR [5]	ViT-L	320 × 800	62.9	55.2	0.504	0.246	0.333	0.261	0.125	290.0 / 317.0
Ours-Faster	ViT-L	320 × 800	62.6	54.2	0.489	0.246	0.330	0.269	0.117	209.0(-28.0%) / 237.2(-25.2%)

Table 5: The profiling analysis.

Method	Module	MACs (G)	FLOPs (G)	Memory (MB)	Time (ms)
StreamPETR (Baseline)	Backbone	2280.0	4560.0	4972.5	290.0
	Decoder	13.3	26.6	405.0	22.6
	Head	7.2	14.4	94.4	4.4
	Total	2300.5	4601.0	5471.9	317.0
Ours-Faster	Backbone	1545.0	3090.0	3759.8	209.0
	decoder	13.3	26.6	405.0	23.7
	head	7.2	14.4	94.4	4.5
	Total	1565.5(**-31.9%**)	3131.0(**-31.9%**)	4259.2(**-22.2%**)	237.2(**-25.2%**)
Sparse4Dv2 (Baseline)	Backbone	2280	4560.0	4977.5	278.8
	Decoder	10.8	21.6	281.3	37.4
	Head	1.7	3.4	7.5	5.8
	Total	2292.5	4585.0	5266.3	322.0
Ours-Faster	Backbone	1545.0	3090.0	3,759.9	206.6
	Decoder	10.8	21.6	281.3	32.3
	Head	1.7	3.4	7.5	5.9
	Total	1557.5(**-32.1%**)	3115.0(**-32.1%**)	4048.7(**-23.1%**)	244.8(**-24.0%**)

detailed metrics (*i.e.*, mATE, mASE, mAOE, and mAAE). This phenomenon is aligned with our foreground-centric design, as our method weighs more computation resources to foreground tokens, and the model is more object-aware.

4 Profiling Analysis

We provide the MACs and FLOPs, as shown in the Tab. 5. Our method significantly reduces the MACs and FLOPs by up to 32% compared with the baseline methods, *i.e.*, StreamPETR [5] and Sparse4Dv2 [2]. The results prove the computational efficiency of our method.

Besides, we conduct a profiling analysis of methods using the ViT-L backbone (also shown in the Tab. 5). It shows that the ViT backbone is the bottleneck of the computational efficiency, which consumes nearly 90% GFLOPs, GPU memory, and inference time of the whole detector. Our method reduces the resource consumption of the backbone by 30% and significantly improves the efficiency of the whole detector.

5 More Qualitative Results

To better understand the superiority of our MQTS, we provide qualitative comparison results in Fig. 1, which clearly shows that MQTS can focus more on

Fig. 1: The visualization of importance score from SparseDETR [3] and our method. It is better viewed in color and zoomed in.

Fig. 2: The visualization of predictions from StreamPETR [5] and our method. It is better viewed in color and zoomed in.

the foreground objects while SparseDETR [3] fails to do that. We also visualize the predictions from our method and StreamPETR in Fig. 2, proving that our method can significantly improve efficiency with nearly the same performance.

Additionally, we provide more visualization results for qualitative analysis, shown in Fig. 3. These results further prove our claim, *i.e.*, the attention map models the correlations between image tokens and history queries, and thus can represent the foreground information density of each token since history queries contain foreground priors. With the 3D foreground object-aware attention map, the whole model can be more concentrated on foreground tokens when keeping ratios ρ getting lower, improving the efficiency.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020)
2. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d v2: Recurrent temporal fusion with sparse model. arXiv preprint arXiv:2305.14018 (2023)
3. Roh, B., Shin, J., Shin, W., Kim, S.: Sparse detr: Efficient end-to-end object detection with learnable sparsity. In: ICLR (2021)
4. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR. pp. 2446–2454 (2020)

5. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. ICCV (2023)

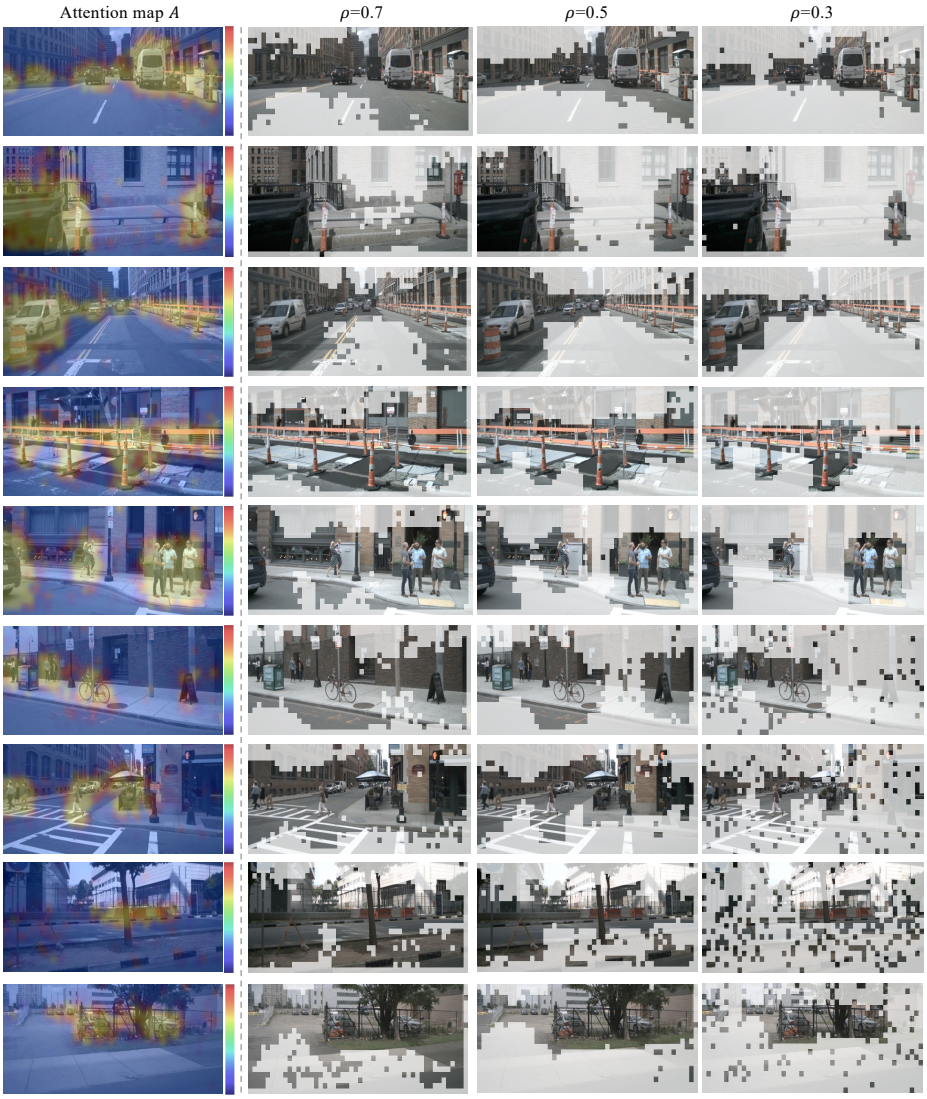

Fig. 3: The visualization of our method in various scenes (better viewed in color). We visualize the attention map in importance score calculation on the left and the salient/redundant tokens after the top-k selection on the right. Redundant tokens are illustrated as translucent.