A Appendix

The supplementary materials are organized as follows. First, in section A.1, we provide some limitations of our method. Then, in section A.2, we present the extra experiments of illustrating the capability of our SEED, applying different backbones, varying the score threshold τ for quality query selection in DQS and the distinct grids in DGA on the Waymo validation set [6] with 20% training data, respectively. Besides, we explore the impact of different numbers of SEED decoder layers on detection performance. In section A.3, we present the comparisons of several variant attention operations for query interaction. In section A.4, we discuss the differences of our proposed DQS and DGA with the existing related methods. Finally, we provide the analysis of visualization, including the learned attention map of DGA and the 3D detection results under different settings in section A.5.

A.1 Limitation

Our method mainly improves the detection head based on the DETR paradigm for 3D object detection. Therefore, the advanced 3D detectors that focus on enhancing the representation ability of 3D backbone are orthogonal to SEED. In the future, we plan to apply our SEED to more powerful 3D backbones on more datasets to further explore the scalability of our method. Besides, we observe that SEED may fail to detect some distant and small 3D objects, but they are clearly visible in 2D camera images. Therefore, exploiting the complementarity of multiple modalities (*i.e.*, 3D point clouds, and 2D camera images) to detect these challenging objects is also our next step.

A.2 Extra Experiments

Table 1: Effectiveness of our SEED. For a fair comparison, we adopt 100% Waymo training data for all models. The results are evaluated by the metric of mAP/mAPH (L2).

Methods	Detection Head	mAP/mAPH (L2)	FLOPs (G)	Params (M)	Latency (ms)
SECOND [9]	Anchor-based	61.0/57.2	91.2	5.3	33.3
CenterPoint [10]	Center-based	68.2/65.8	141.2	7.8	44.0
PV-RCNN++ [5]	RoI-based	71.7/69.5	166.6	16.1	149.0
VoxelNeXt [3]	Center-based	72.2/70.1	624.9	29.3	124.7
TransFusion [1]	DETR-based	−/64.9	96.8	7.9	70.5
ConQueR [12]	DETR-based	70.3/67.7	167.3	15.1	99.1
FocalFormer3D [2]	DETR-based	71.5/69.0	144.9	19.4	97.2
SEED-S (Ours)	DETR-based	**73.1/70.8**	168.7	12.8	74.2
SEED-L (Ours)	DETR-based	**75.5/73.5**	648.1	33.1	163.8

Capability of our SEED. To verify the capability of our SEED, we adopt the small version SEED-S with the same 3D backbone as CenterPoint [10] for a fair

comparison with existing representative 3D object detection methods, including anchor-based [9], center-based [10], RoI-based [5] and DETR-based [1, 2, 12] detectors. We conduct the comparisons of these methods in terms of performance, FLOPs, parameters, and latency, shown in Table 1. Note that the main difference between these methods is the design of the detection head. Moreover, we evaluate the running speed of all approaches on one NVIDIA GeForce RTX 3090 with a batch size of 1 according to their corresponding official open-source code for a fair comparison. Compared with SECOND [9] and CenterPoint [10], our SEED-S has a slower running speed, but our performance greatly exceeds them with 13.6 and 5.0 mAPH/L2, respectively. Furthermore, benefiting from the well-designed DQS module for selecting high-quality queries and the superior DGA operation for effective feature interaction, the detection performance of our SEED-S even outperforms PV-RCNN++ [5] of 1.3 mAPH/L2 with 2× faster running speed. However, existing DETR-based methods still fall behind PV-RCNN++ in terms of detection performance. The above experimental results effectively illustrate the powerful capability of our SEED.

Table 2: Effectiveness of our SEED on different backbones on the Waymo validation set [6] with 20% training data. We use mAP/mAPH (L2) for evaluating the detection performance. $*$ means our reproduced performance from the official code.

Methods	3D AP/APH (L2)			mAP/mAPH (L2)
	Vehicle	*Pedestrian*	*Cyclist*	
CenterPoint-Pillar [10]	62.2/61.7	65.1/55.0	63.0/61.5	63.4/59.4
+**SEED Detection Head**	67.0/66.5	71.3/62.0	65.8/64.5	**68.0/64.3**
CenterPoint [10]	63.2/62.7	64.3/58.2	66.1/64.9	64.5/61.9
+**SEED Detection Head**	68.5/68.1	72.1/66.5	71.2/70.0	**70.6/68.2**
DSVT-Pillar$*$ [8]	69.7/69.2	74.9/68.0	70.7/69.6	71.8/68.9
+**SEED Detection Head**	71.7/71.3	75.4/68.7	73.0/71.8	**73.4/70.6**
HEDNet$*$ [11]	70.8/70.3	75.0/70.3	73.6/72.6	73.1/71.1
+**SEED Detection Head**	72.4/72.0	76.3/71.3	74.9/73.8	**74.5/72.4**

Effectiveness of our SEED with Different Backbones. Note that our SEED focuses on the design of detection head based on the DETR paradigm. Therefore, to verify the effectiveness of our SEED, we decorate our SEED detection head with different backbones, including CenterPoint-Pillar (pillar-based) [10], CenterPoint (voxel-based) [10], DSVT-Pillar [8] and HEDNet [11]. In Table 2, we present the corresponding detection results on the Waymo validation set [6] with 20% training data. We clearly observe that our approach yields consistent performance improvement under different backbones, proving the generality of our SEED detection head.

Effect of τ for Quality Query Selection. To explore the effect of the classification score threshold τ in formula (4) of the main paper for quality query selection, we set different score thresholds of $\tau = 0.0$, $\tau = 0.2$, and $\tau = 0.3$, whose results are summarized in Table 3. When the score threshold is set as 0.0, we find there is a drastic drop in detection performance. Since the predicted object

Table 3: The effect of different classification score thresholds for quality query selection in dual query selection (DQS). We use mAP/mAPH (L2) for evaluating the detection performance.

τ	3D AP/APH (L2)			mAP/mAPH
	Vehicle	*Pedestrian*	*Cyclist*	(L2)
0.0	66.6/66.2	70.1/64.5	69.1/67.8	68.6/66.2
0.2	68.5/68.1	72.1/66.5	71.2/70.0	70.6/68.2
0.3	68.7/68.2	71.9/66.4	71.0/69.8	70.5/68.1

Table 4: Effectiveness of our SEED. The results are evaluated by the metric of mAP/mAPH (L1 and L2). We evaluate the latency of our SEED for different gird sizes on one NVIDIA GeForce RTX 3090 with a batch size of 1.

Grids	mAP/mAPH (L1)	mAP/mAPH (L2)	Latency (ms)
3×3	76.7/74.1	70.3/67.8	73.1
5×5	77.0/74.4	70.6/68.2	74.2
7×7	77.1/74.5	70.7/68.3	77.8

Table 5: The effect of the number of SEED decoder layers in transformer decoder for 3D detection performance. We use mAP/mAPH (L2) to evaluate the detection performance.

Layers	3D AP/APH (L2)			mAP/mAPH
	Vehicle	*Pedestrian*	*Cyclist*	(L2)
1	66.8/66.2	69.4/62.0	69.2/67.8	68.5/65.3
3	68.4/68.0	71.5/65.6	70.9/69.7	70.3/67.8
6	68.5/68.1	72.1/66.5	71.2/70.0	70.6/68.2

score is close to 0.0, it is more likely to be considered a background object. At this time, the estimated localization scores that are mainly for foreground objects rather than background objects are unreasonable, leading to selecting out poor queries in the stage of quality query selection. Therefore, setting a proper score threshold (*e.g.*, 0.2 or 0.3) to eliminate the negative impact of background objects for quality query selection in DQS is necessary.

Ablation for Distinct Grids. As shown in Table 4, we conduct experiments of varying grid sizes in DGA to investigate their impact on the detection performance and latency. With increasing the grid sizes ($3 \times 3 \rightarrow 5 \times 5$), the detection performance of our SEED can be consistently improved in terms of mAP/mAPH (L2). However, the corresponding computational costs are also increasing due to more sampled features being performed for query interaction, leading to more latency. Therefore, in our paper, we choose a proper grid size of 5×5 as default to trade off the detection performance and latency.

Number of SEED Decoder Layers. To analyze the effect of different numbers of SEED decoder layers on detection performance, we provide the experimental results in Table 5. When only one SEED decoder layer is applied in

Table 6: The comparison of different query selection in terms of latency.

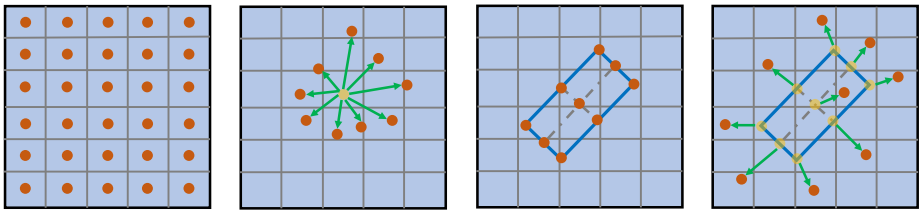
Query Selection Methods	mAP / mAPH (L2)	Latency (ms)
TransFusion [1] (Heatmap-based)	67.5 / 65.0	**2.5**
ConQueR [12] (Top-N)	69.1 / 66.8	7.3
SEED (DQS)	**70.6 / 68.2**	10.0

the transformer decoder to extract contextual features of point clouds, a relatively poor detection performance with 65.3 mAPH/L2 is obtained. In contrast, stacking three SEED decoder layers brings an obvious performance gain with 2.5 mAPH/L2 thanks to their more powerful feature extraction capabilities. Intuitively, stacking more SEED decoder layers is beneficial. Therefore, in this paper, we adopt the commonly used six SEED decoder layers in the transformer decoder, which produces a better result with 68.2 mAPH/L2 than the settings of using fewer decoder layers (*i.e.*, 65.3 mAPH/L2 for one decoder layer or 67.8 mAPH/L2 for three decoder layers).

Latency of Different Query Selection. In Table 6, we provide the latency of different query selection methods. We can observe that our DQS with high performance does not bring significant latency compared with the Top-N method.

A.3 Different Attention Operations

To clearly illustrate the difference between our proposed deformable grid attention (DGA) and existing representative attention operations (*i,e*, global attention [7], deformable attention [13] and box attention [4]), we present the simple schematic diagrams of these methods as shown in Figure 1. For the global attention in Figure 1 (a), each query implements feature interaction with all features (as key and value). This operation usually brings unacceptable computational costs, especially for using high-resolution feature maps as keys or values. Therefore, the local attention operations including the deformable attention, the box attention, and our DGA in Figure 1 (b) (c) (d) are more proper to perform query interaction than the global attention in point clouds. Specifically, deformable attention is good at capturing the crucial regions of objects in a flexible receptive filed manner, but the learned offsets without geometric prior information as reference are difficult to predict accurately. The box attention operation can make use of geometric information of some regular objects (*e.g.*, *Vehicle*), but it requires a precise box regression, and its receptive field is not as flexible as the deformable attention. In contrast, our deformable grid attention has the advantages of both the flexible receptive field of deformable attention and the rich geometric information of the box attention, which can enable the network to focus on relevant regions and capture more informative features even for objects with diverse shapes.



(a) Global Attention (b) Deformable Attention (c) Box Attention **(d) DGA (Ours)**

Fig. 1: Comparison of deformable gird attention (DGA) with other attention operations. The orange points represent the sampling features, the yellow points represent the reference points, and the green arrows represent the predicted offsets. Note that global attention adopts a global manner for query interaction, that is, treating all features as sampling features.

Table 7: Comparison of our SEED and FocalFormer3D. $*$ indicates the deformable attention in FocalFormer3D [2]

DQS	multi-stage	mAP/mAPH (L2)
–	–	67.5/65.0
–	✓	68.2/65.5
✓	–	70.6/68.2
✓	✓	**70.9/68.3**

(a) Comparison for query selection.

Method	mAP/mAPH (L2)
Deformable Attention [13]	69.9/67.5
Deformable Attention$*$ [2]	70.0/67.6
DGA (Ours)	**70.6/68.2**

(b) Comparison for query interaction.

A.4 Discussion

DQS *vs.* Multi-stages to Select Queries. Actually, our DQS not only uses a foreground query selection module to select coarse queries with a high recall, but also leverages a quality query selection module to obtain high-quality queries. However, FocalFormer3D [2] primarily utilizes multi-stage foreground scores to obtain queries with higher recall, but it overlooks the importance of query quality for box localization. Furthermore, we present a comparison between our DQS and the multi-stage approach in Table 7a. We observe that DQS achieves much better performance (68.2 *vs.* 65.5), which indicates the importance of selecting high-quality queries. In Table 7a, we also integrate this multi-stage strategy into our DQS, which brings a subtle gain of 0.1 mAPH/L2.

DGA *vs.* Deformable Attention in FocalFormer3D. Here, we discuss the difference between our proposed DGA and deformable attention in FocalFormer3D [2] for query interaction. In fact, FocalFormer3D adopts the **same** deformable attention with deformable DETR [13]. The only difference with [13] is that FocalFormer3D uses the enhanced queries by combining the RoI features for feature interaction instead of the original queries. In contrast, our DGA is a **new** deformable attention, which uniformly divides each reference box into grids as the reference points and then utilizes the predicted offsets to achieve a flexible receptive field. In Table 7b, we provide the comparison with FocalFormer3D, whose

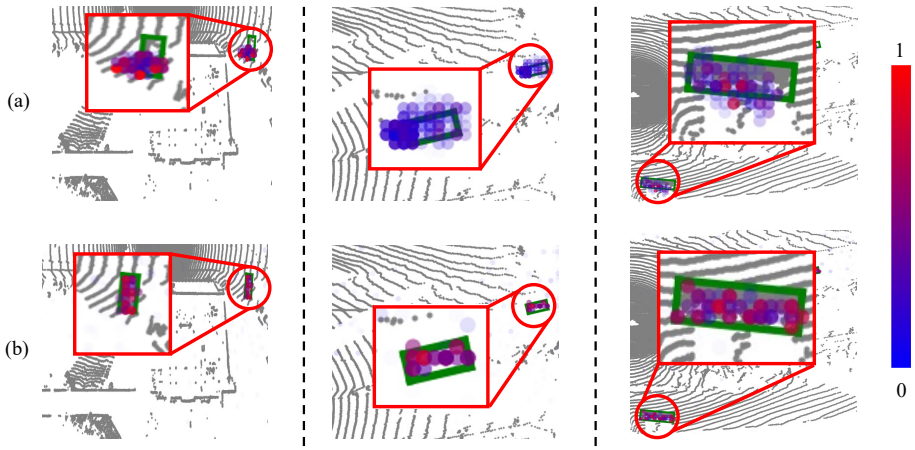


Fig. 2: Comparison of attention map without DGA (a) and with DGA (b) on the Waymo validation set. Green boxes are the ground truths. The circle represents the position of the attention, and its corresponding color means the weight of the attention. After utilizing DGA, SEED can capture the geometric information of 3D objects in a flexible receptive field and achieve better query interaction.

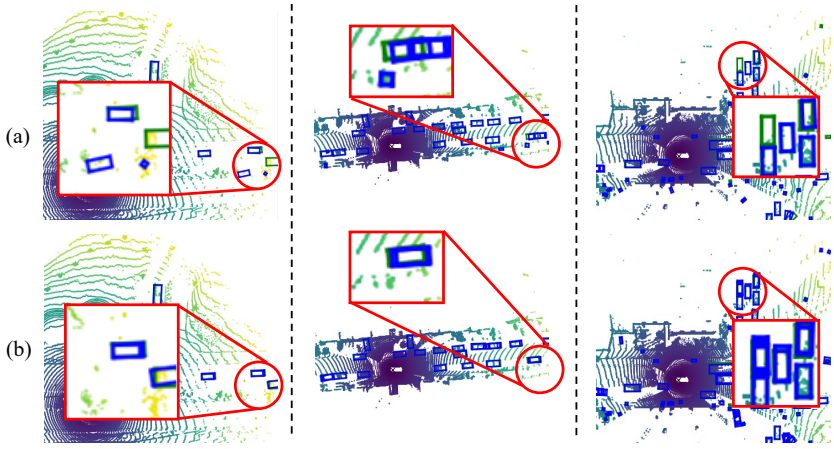


Fig. 3: Comparison of detection results without DQS (a) and with DQS (b) on the Waymo validation set. Blue and green boxes are the prediction and ground truths, respectively. After utilizing DQS, our SEED can successfully detect some hard objects and reduce some false positives, which are highlighted by red circles.

performance (67.6 mAPH/L2) is still inferior to our DGA (68.2 mAPH/L2). Additionally, we provide a clear illustration of the difference between our DGA and deformable attention in Figure 1.

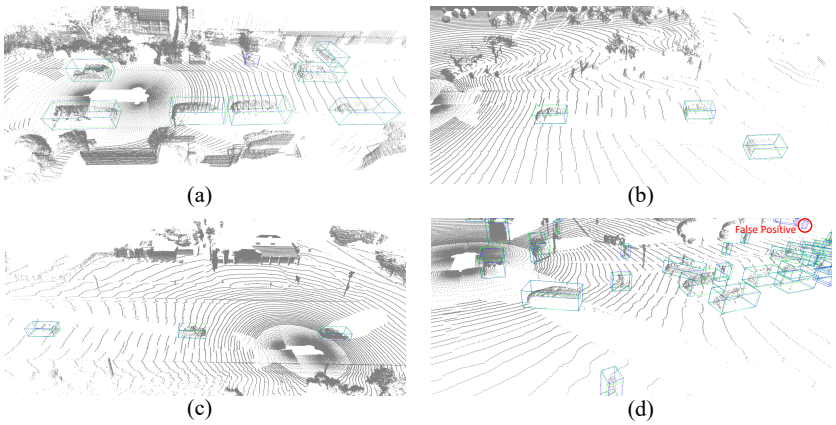


Fig. 4: Qualitative results of SEED on the Waymo validation set. Blue and green boxes are the predictions and ground truths, respectively. Besides, we highlight the false positive with a red circle.

A.5 Visualization

Visualization of Learned Attention Map. As shown in Figure 2, we present the visualization of learned attention maps under the settings of our SEED with DGA (b) and without DGA (a) (*i.e.*, box attention [4]). In the first column, we can observe that DGA captures the key regions even if there is no accurate proposal box as a reference, benefiting from its flexible receptive field. In the second column, we find that DGA produces higher attention weight on objects than the manner without DGA. In the third column, our DGA not only has good robustness in estimating the direction angle but also focuses on key features, such as the boundary and center of the object. The above visualizations effectively demonstrate the superiority of our DGA for query interaction.

Comparisons for w/ and w/o DQS. To verify the effectiveness of our DQS, we visualize the detection results of our SEED with DQS and without DQS (*i.e.*, directly select Top N_f queries in one step) on the Waymo validation set, which is depicted in Figure 3. In the first column, our method can accurately locate all objects and distinguish a False Positive (FP). Besides, as shown in the second column of Figure 3, we observe that our SEED with DQS can pick out some high-quality queries for accurate localization. Finally, surprisingly, our method has the ability to detect a hard distant object even with some occlusions, as shown in the third column of Figure 3. These interesting phenomena illustrate the effectiveness of our approach.

Visualization for SEED. We visualize the qualitative results of SEED on the Waymo validation set, which is shown in Figure 4. Benefiting from the dual query selection for high-quality query selection and the deformable grid attention for effective query interaction, our SEED can detect 3D objects well on large-scale point clouds. Besides, in Figure 4 (d), we carefully find that there are several

False Positives (*e.g.*, *Pedestrian*) in the distant areas. Therefore, we plan to utilize the complementarity of multiple modalities (*i.e.*, 3D point clouds, and 2D camera images) to distinguish these challenging objects in the future.

References

1. Bai, X., Hu, Z., Zhu, X., Huang, Q., Chen, Y., Fu, H., Tai, C.L.: Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In: CVPR. pp. 1090–1099 (2022)
2. Chen, Y., Yu, Z., Chen, Y., Lan, S., Anandkumar, A., Jia, J., Alvarez, J.M.: Focalformer3d: Focusing on hard instance for 3d object detection. In: ICCV. pp. 8394–8405 (2023)
3. Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J.: Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In: CVPR. pp. 21674–21683 (2023)
4. Nguyen, D.K., Ju, J., Booij, O., Oswald, M.R., Snoek, C.G.: Boxer: Box-attention for 2d and 3d transformers. In: CVPR. pp. 4773–4782 (2022)
5. Shi, S., Jiang, L., Deng, J., Wang, Z., Guo, C., Shi, J., Wang, X., Li, H.: Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. IJCV (2021)
6. 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)
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. vol. 30 (2017)
8. Wang, H., Shi, C., Shi, S., Lei, M., Wang, S., He, D., Schiele, B., Wang, L.: Dsvt: Dynamic sparse voxel transformer with rotated sets. In: CVPR. pp. 13520–13529 (2023)
9. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
10. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: CVPR (2021)
11. Zhang, G., Junnan, C., Gao, G., Li, J., Hu, X.: Hednet: A hierarchical encoder-decoder network for 3d object detection in point clouds. In: NeurIPS. vol. 36 (2024)
12. Zhu, B., Wang, Z., Shi, S., Xu, H., Hong, L., Li, H.: Conquer: Query contrast voxel-detr for 3d object detection. In: CVPR. pp. 9296–9305 (2023)
13. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR (2021)