Boosting Performance on 3D Object Detection with a Plug-in Discrimination Module

Yi Yang[1,2,a] **and Zhang Zhang**[1,2,b,*]

[1]School of Artificial Intelligence, University of Chinese Academy of Science(UCAS).
[2]Institute of Automation, Chinese Academy of Science. Beijing, China.

E-mail: [a]yangyi2021@ia.ac.cn; [b,*]zzhang@nlpr.ia.ac.cn

Abstract. Around-view multi-camera 3D object detection in BEV (Bird's-Eye-View) space has been a research focus over the past few years. As a typical supervised training task, many researchers promote this area with different task-specific key designs, such as exploiting temporal information and correspondence of perspective image plane and BEV space. Most of these works follow the DETR detection framework, yet the nature of learnable queries in DETR, the encodings of objects' center and bounding box information, have not been discussed in previous studies. In this paper, we take advantage of this prior and further extend it to 3D detection tasks. In 3D object detection, the ground-truth bounding boxes are hardly overlapping. Therefore, the queries should be more diverse under this hypothesis. To achieve this goal, we propose a Plug-in Discrimination Module (PDM) to discriminate learnable queries from all the other queries with a discrimination loss to ensure the diversity of queries. The PDM is a simple train-time-only module. It contains a query projection head to project all the object queries into a common latent space. In the latent space, the discrimination loss is conducted on all the queries. Experimental results show that this design can directly improve the 3D detector's performance without modifying the detector's architecture and adding extra inference costs. The NDS improvement on the nuScenes dataset is up to a maximum of 1.62% in the 8th training epoch and remains an average 0.64% improvement in the following epochs, compared with the baseline model.

1. Introduction

In the world of computer vision, the ability to accurately identify and locate objects within a three-dimensional space is a critical step toward machine perception that mirrors human visual understanding. This technology, known as 3D object detection, is pivotal in various applications ranging from autonomous driving to augmented reality. 3D object detection refers to the process by which computers recognize and precisely determine the position and orientation of objects in 3D space. Unlike its 2D counterpart, which can only provide information about objects in the plane of the image (height and width), 3D detection adds a crucial depth component, offering a complete spatial representation.

The fundamental mechanics involve the use of sensors like LiDAR (Light Detection and Ranging), stereo cameras, or even single monocular cameras to capture environmental data. 3D detectors are to process the data, extract features, identify object classes, and localize objects, providing position, shape, and orientation information. Several challenges arise with visual perspective 3D object

detection. First, the association of multi-cameras is difficult in surrounding scene perception. Second, the combination of camera and LiDAR is restricted in perspective view.

To address these challenges, researchers start to transform the image from a perspective plane to a Bird's-Eye-View (BEV) space. BEV space is a view of 3D space from above as if one is looking down from the sky. This view flattens the three-dimensional world onto a two-dimensional plane while preserving the spatial relationships between objects. It provides a clear and intuitive layout of the environment, making it especially useful for tasks such as path planning and navigation in autonomous vehicles. The Bird's-Eye-View simplifies complex scenes by removing the effects of perspective, such as the foreshortening seen in images captured from ground level. This makes it easier to measure distances between objects and understand their relative positions. Moreover, BEV can unify the data coming from multiple sensors, creating a comprehensive and actionable map of the surroundings.

In practical applications and current research focus, 3D detection in BEV space generally follows DETR[1] architecture, a transformer-based encoder-decoder architecture. Related researchers have devoted many efforts to improve 3D detection performance in BEV space, such as various approaches to exploiting temporal information[2, 3, 4] and projection between 2D plane and BEV space[5]. How to propose an efficient set of object queries has been investigated in a series of 2D DETR detection work, but the design of object queries in 3D detection generally follows previous perspective work[6] and is not well discussed in a 3D context. Given a set of object queries that encode the objects' position and bounding box information around several predefined anchors, one main difference from detection in the real-world coordinate to detection in the perspective image plane is that objects are hardly overlapping in 3D space. There is only one object in one 3D position, therefore the rivalry of multiple queries detecting multiple objects in one position does not exist in the detection task for the object's real-world coordinates. Each object query only needs to be in charge of its area. By this intuition, we design a Plug-in Discrimination Module (PDM), a projection head, to discriminate the embeddings of various object queries. In particular, we introduce discrimination loss, a loss function mimicking the design of contrastive loss.

Our contributions can be summarized as follows.

- We propose a Plug-in Discrimination Module (PDM) to improve the performance of the 3D detector. Though the PDM is a simple design, a two-layer non-linear projection head, it still helps improve SOTA 3D detectors without modifying the network architecture.
- We propose to conduct a discrimination loss on a set of learnable parameters, i.e. object queries, unlike traditional unsupervised learning tasks to discriminate samples. Also, we exploit the unsupervised learning loss to aid supervised 3D detection tasks, filling the gap between these two kinds of tasks.

2. Related Work

Object queries in DETR: In the realm of object detection, CNN-based backbone, and anchor- based design are popular choices of early classical detectors[7, 8]. In contrast to previous detectors, DETR[1] is fully anchor-free by detecting objects with a set of learnable queries. However the limitations of DETR are still obvious: (1) The convergence speed of DETR is slow, DETR takes 500 epochs to reach a competitive performance to FasterRCNN[7] trained in less than 100 epochs. (2) The physical meaning of object query, a key design in DETR, is not clear. (3) The attention computation complexity is O(n2), bringing difficulty to leverage high- resolution feature maps. (4) DETR requires a preset large number of queries to detect objects in crowded scenes, which causes high computation complexity of cross-attention and bipartite graph matching. To mitigate these limitations, Deformable-DETR[9] introduces the concept of reference points as a prior of central points of objects, while the queries are in charge of predicting offset from reference points to object center. They also introduce a deformable attention module to efficiently leverage multi-scale feature maps. Similarly, conditional DETR[10] uses position embeddings encoded from reference points. They further decouple and divide the object query into a content query and a spatial query, forcing the spatial query to identify the object's center and bounding box. Anchor DETR[11] exploits spatial information more explicitly. The

queries in Anchor DETR are directly encoded from learned reference points. Apart from reference points, DAB-DETR[12] constrains the query generation with both learned reference points and bounding boxes. These previous studies have extensively clarified that the queries are encodings of objects' position and bounding box information.

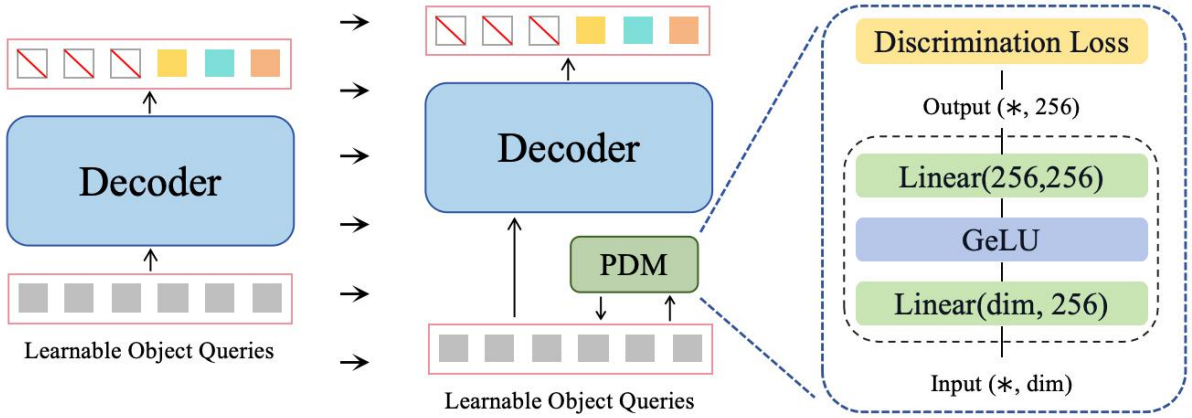

Figure 1. A simplified diagram of DETR-like detector head.

3D Detection in BEV space: Early 3D detection work[13] explicitly predicts object's 3D position given perspective view and camera extrinsic. Further, the concept of Bird's-Eye-View is introduced into 3D detection. By transforming features from perspective view onto BEV space, models can associate image features to real-world 3D coordinates. There are generally two kinds of 3D detection in BEV space, forward and backward[5]. The forward approaches lift the image features from the perspective plane to the BEV space, such as LSS[14] and BEVDepth[15]. LSS predicts a depth distribution of the image plane with a depth network. The feature of a pixel is the outer product of depth distribution and CNN feature. In this way, LSS 'lifts' a flat feature map to a depth feature map. Then the features from pixel coordinates are transformed to world coordinates by camera intrinsic and camera extrinsic. BEVDepth[15] follows this idea and further improves the critical depth network. They exploit point clouds as explicit depth supervision to train a depth network instead of using a fixed pre-trained depth network as in LSS. This line of BEV approaches usually relies heavily on the accuracy of the depth network. The accumulated error from the depth network is difficult to avoid. So some researchers developed another line of BEV approaches that directly pre-define dense voxel locations in BEV space and project these 3D points onto the perspective plane to fetch image features. The DETR3D[16] predicts 3D reference points in world coordinates based on learnable queries, then projects these reference points onto perspective views. The initial predictions of 3D reference points are inefficient. So the PETR[17] generates queries from 3D mesh grid points. With a predefined mesh grid as initialization, the queries are iteratively updated and refined. Apart from the 3D position used for query generation, sparseBEV[6] further takes more information to initiate queries, including 3D bounding box, orientation, and velocity, mimicking the evolution from Anchor DETR to DAB-DETR. BEVFormer[2] exploits this projection further, after the features are fetched from the perspective view, the detection is not directly performed. The features contained in 3D mesh grid points are pooled to form a 2D BEV space feature map. The obtained feature map is further exploited as the context of the detection head.

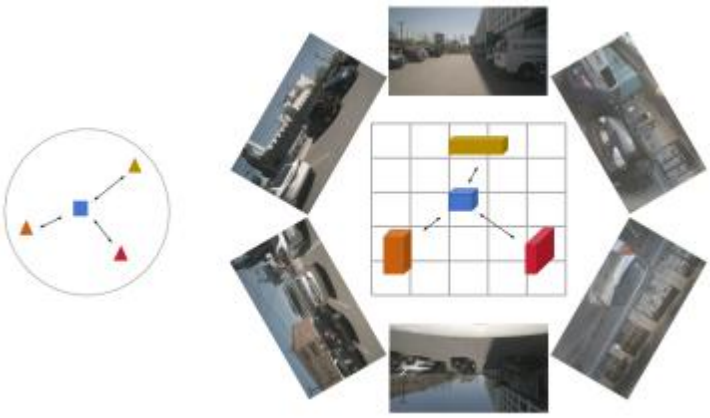

Figure 2. A diagram of how discrimination loss affects detection in BEV space.

3. Method

The general DETR detection head can be diagrammed in the left part of Figure.1. In the decoder, multi-head self-attention and cross attention with image feature are performed on queries. The outputs of the decoder are to predict the object's position and bounding box. A hungarian match is conducted to activate the closest prediction to the ground truth and suppress unmatched predictions. We propose to add a simple auxiliary plug-in discrimination module before queries are sent to the decoder, as depicted in the right part of Figure.1. It's a train-time-only module and contains a simple 2-layer non-linear MLP with GeLU activation to project queries into a shared latent space. The discrimination loss will be conducted on the output of the projection head.

Following the definitions of contrastive learning, we define concepts in our discrimination loss. For a predefined set of N object queries, each query q_i is a positive pairing of its own. While all the other queries $\{q_j \mid j \neq i\}$ are the negative pairings of q_i. Suppose the embedding e_i of the query q_i is obtained by projection head f, and the discrimination loss L is conducted on the embedding with a cosine similarity function. The loss function can be expressed as follows.

$$e_i = f(q_i) \tag{1}$$

$$l(e_i) = -\frac{1}{\sum_{j=1,j\neq i}^{N} \mathrm{sim}(e_i, e_j)} \tag{2}$$

$$\mathrm{sim}(e_i, e_j) = \frac{< v_i, v_j >}{||v_i|| \times ||v_j||} \tag{3}$$

$$L = \sum_{i=1}^{N} l(e_i) \tag{4}$$

An intuitive illustration of how the loss function works is shown in Figure.2. As the discrimination loss separates queries in the latent space, the position encodings of queries in the BEV space are also diverse. This design would help 3D detectors accommodate difficult driving scenes, such as crowded urban streets and crossroads.

From the perspective of unsupervised learning, the discrimination loss is different from traditional contrastive learning conducted on the embedding of samples. We discriminate all the queries to force them to attend to different real-world areas. The queries are forced to be diverse during training by our designed PDM. In inference time, the projection head is removed. Such an approach to contrast on a set of learned parameters has the following advantages:

- Easy to deployed in training progress: Classical contrastive learning relies on special designs like large batch size[18], momentum encoder[19] or memory bank[20] to ensure there are plenty of negative pairings to ensure model optimizing in a right direction. Yet switching dimensions to contrast from sample level to learned parameters releases the learning difficulty in contrastive learning.
- A plug-in training module: The projection heads along the discrimination loss are all auxiliary components. They would not introduce any extra costs to inference or modify network architecture. Therefore, our approach is a general plug-in module for any 3D

4. Experiments

4.1. Implementations

Experiments are conducted with the SOTA methods, SparseBEV[6], on nuScenes dataset[21]. As described above, the network architecture is not modified, except that the PDM, a two-layer MLP with GeLU[22] activation is introduced. All the hyper-parameters and configurations are set to the same as in SparseBEV. To better fit our servers, among the settings SparseBEV supports, we choose the setting of ResNet-50 as the perspective image encoding backbone, 704 × 256 as the input image size, and a total of 24 training epochs. Under such settings, the GPU memory consumption is about 5634MB for one sample per GPU. Total training time is less than 12 hours with an 8-RTX3090 server. The results of the test set will be reported.

Table 1 Main experiment results of SparsBEV baseline and SparseBEV with Discrimination Module

Epoch	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
SparseBEV Baseline							
4	44.54%	33.09%	70.23%	28.66%	69.70%	31.49%	19.92%
8	48.14%	36.94%	67.14%	27.48%	57.92%	31.20%	19.49%
12	52.10%	41.52%	64.73%	27.47%	48.79%	26.18%	19.44%
16	53.16%	42.61%	62.18%	27.08%	47.93%	25.01%	19.27%
20	54.10%	43.82%	61.46%	27.33%	45.42%	25.27%	18.60%
24	54.68%	44.49%	60.38%	27.15%	44.56%	24.90%	**18.69%**
SparseBEV+Plug-in Discrimination Module							
4	44.63%	33.15%	70.21%	29.27%	69.45%	29.45%	21.14%
8	49.76%	39.10%	64.78%	27.31%	57.02%	28.18%	20.63%
12	52.49%	41.98%	60.91%	27.07%	51.70%	25.65%	19.71%
16	53.77%	42.89%	62.18%	26.87%	43.74%	24.06%	19.88%
20	54.95%	44.54%	59.56%	26.88%	43.61%	24.44%	18.71%
24	**55.39%**	**45.09%**	**58.64%**	**26.84%**	**43.06%**	**24.05%**	18.98%

↑ means the greater value the better performance.
↓ means the smaller value the better performance.

4.2 Dataset and Metircs

The experiments are conducted on the nuScenes dataset, a public large-scale dataset for autonomous driving. The dataset consists of 1000 driving scenes in Boston and Singapore. Every scene is a multi-modal data sequence of 20 seconds, containing 6 surround-view cameras, 1 lidar, and 5 radars. The 1000 scenes are split into 700/150/150 for training/validation/testing.

The nuScenes dataset supports 3D detection tasks. Approximately 1.4M object bounding boxes are annotated in 40k keyframes. The annotated objects include 10 classes: barrier, bicycle, bus, car, motorcycle, pedestrian, traffic cone, trailer, truck, and construction vehicle, which are all common objects in driving maneuvers.

The evaluation metrics for 3D detection include 7 metrics:

- AP: Average Precision for detection;
- ATE: Average Translation Error;
- ASE: Average Scale Error;
- AOE: Average Orientation Error;
- AVE: Average Velocity Error;
- AAE: Average Attribute Error;
- NDS: nuScenes Detection Score, a composite metric of metrics above.

4.2. Dataset and Metircs

The main results are shown in Table.1 The experiment is conducted with 900 object queries. In a total of 24 training epochs, we evaluate the model on a validation set every 4 epochs and record the performance of the model. The best results are emphasized. In almost all metrics, SparseBEV with PDM is superior to the baseline. The average attribute error (ATE, indicating the accuracy of classification) of SparseBEV with PDM is slightly inferior to the baseline. This is because discrimination learning is conducted on all the channels of queries, given the attributions of the object is limited to the 10 classes, it might not need to be discriminated too much.

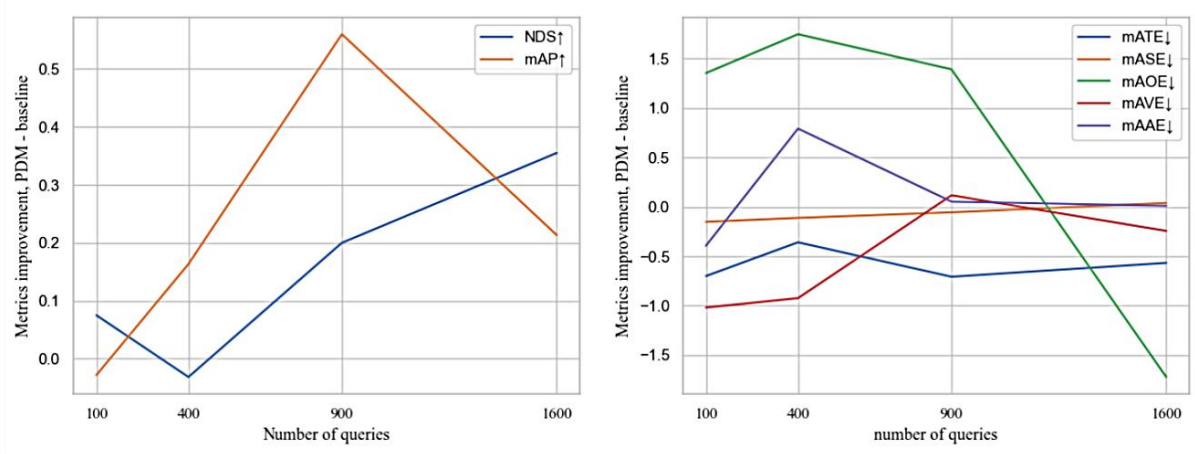

Figure 3. Average metrics improvement after baseline equipped with the PDM on test set during training time. The left part contains the NDS and mAP, which are the greater the better. The right part contains the mATE, mASE, mAOE, mAVE, and mAAE, which are the smaller the better.

4.3. Ablation on number of query

We conduct an ablation study on the number of queries to check out the effect of the PDM under different query settings. We record metrics during training and use the metric from the model with the PDM to minus the metric from the baseline. The number of the query is set from 100 to 1600, results can be seen in Figure.3.

In general, the PDM contributes to more NDS improvement as the number of queries increases. But we notice a drop in 400 queries. We suspect that the setting of 400 queries is a suitable hyper-parameter for the nuScenes dataset. With fewer queries, i.e. 100 queries, the PDM makes the small number of queries able to attend to the overall detection space. On the other hand, when the number of queries is greater, i.e. 900/1600 queries, the queries are crowded in the detection space, and the PDM further forces queries to attend to much more detailed information, such as orientation and velocity.

Further, we look into more detailed components of NDS. The mAP improvement is correlated to the number of queries which verifies our hypothesis, that discriminating queries in 3D detection helps detection. On the other hand, for the detailed information of objects, the mATE and mASE are correlated to mAP, they are generally better than baseline. The mAOE, mAVE, and mAAE are not quite directly related to position information, discriminating queries will not help improve them, thus leading to diverse results.

5. Conclusion

In this paper, we propose a Plug-in Discrimination Module (PDM). This discrimination design originates from the observation that in 3D detection tasks, objects hardly overlap with each other. Given object queries are encodings of object positions, discriminating queries contributes to making them more diverse. Therefore, queries can attend to more areas in the BEV space and detect various shapes of objects. Also, in contrast to previous contrastive learning work conducted on samples, the PDM is conducted on a set of learnable parameters, extending the unsupervised learning to a more general case. Experimental results show that under various number of query settings, 3D detector trained with the PDM is superior to the baseline model on the NDS score and mAP. Yet the errors are diverse and not prone to becoming better. The non-overlapping hypothesis is a distinct property for 3D detection tasks. The more elaborate way to force the model to learn this hypothesis could be studied in further research.

Acknowledgments

This work was jointly supported by National Key R and D Program of China (Grant No. 2022ZD0117901) and National Natural Science Foundation of China (Grant No. 62373355).

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In European conference on computer vision, pages 213 – 229. Springer, 2020.

[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 European conference on computer vision, pages 1 – 18. Springer, 2022.

[3] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Aqi Gao, Tiancai Wang, and Xiangyu Zhang. Petrv2: A unified framework for 3d perception from multi-camera images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3262 – 3272, 2023.

[4] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. arXiv preprint arXiv:2303.11926, 2023.

[5] Zhiqi Li, Zhiding Yu, Wenhai Wang, Anima Anandkumar, Tong Lu, and Jose M Alvarez. Fb-bev: Bev representation from forward-backward view transformations. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6919 – 6928, 2023.

[6] Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 18580 – 18590, 2023.

[7] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440 – 1448, 2015.

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779 – 788, 2016.

[9] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159, 2020.

[10] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3651 – 3660, 2021.

[11] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In Proceedings of the AAAI conference on artificial intelligence, volume 36, pages 2567 – 2575, 2022.

[12] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv preprint arXiv:2201.12329, 2022.

[13] 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, pages 913 – 922, 2021.

[14] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23 – 28, 2020, Proceedings, Part XIV 16, pages 194 – 210. Springer, 2020.

[15] 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. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 1477 – 1485, 2023.

[16] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In Conference on Robot Learning, pages 180 – 191. PMLR, 2022.

[17] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In European Conference on Computer Vision, pages 531 – 548. Springer, 2022.

[18] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597 – 1607. PMLR, 2020.

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9729 – 9738, 2020.

[20] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3733 – 3742, 2018.

[21] 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. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11621 – 11631, 2020.

[22] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016.