DN-PETR: Denoising Position Embedding Transformation for Multi-View 3D Object Detection

Tung Yen Chiang
Electrical and Computer Engineering
University of California, San Diego
La Jolla, California
t2chiang@ucsd.edu

Tung Hsiao
Electrical and Computer Engineering
University of California, San Diego
La Jolla, California
tuhsiao@ucsd.edu

***Abstract*—In this paper, we introduce a novel denoising training method designed to accelerate PETR training and provide deeper insights into the slow convergence issues associated with PETR-like methods. We identify that the slow convergence is due to the instability of bipartite graph matching, which leads to inconsistent optimization goals in the early stages of training. To address this, in addition to using the Hungarian loss, our method incorporates feeding ground truth bounding boxes with added noise into the Transformer decoder, training the model to reconstruct the original boxes. This approach effectively reduces the difficulty of bipartite graph matching and accelerates convergence.**

***Index Terms*—Vision Transformer, Object Detection, PETR, Denoising Training**

I. Introduction

Object detection, a core computer vision task, involves identifying objects in images by predicting their bounding boxes and classes. While classical detectors, primarily based on convolutional neural networks, have shown significant advancements, the field recently saw a paradigm shift. Carion et al. introduced Transformers to object detection with their DETR [1] model, offering a robust alternative for 2D object detection.

Over time, research in object detection has transitioned from 2D to 3D paradigms, with a growing interest in 3D object detection from multi-view images. This shift is particularly prominent in autonomous driving systems, where the cost-effectiveness of multi-view approaches is highly valued.

DETR has gained significant traction for its innovative end-to-end object detection framework. In DETR, each object is represented by a unique object query that interacts with 2D image features in the transformer decoder to generate predictions.

Building on this concept, DETR3D [2] offers an intuitive extension of the DETR framework for end-to-end 3D object detection. The key idea is to project each object query's predicted 3D reference point back into the image spaces of all camera views using known camera parameters. This projection enables the sampling of 2D features from each view. The decoder then processes these sampled features along with the object queries, refining the object query representations.

To overcome the accuracy limitations of DETR3D, researchers developed PETR [3], a model that enhances 3D object detection by transforming 2D features from multi-view images into 3D representations. This is achieved by encoding 3D position embeddings, a process that begins with discretizing the shared camera frustum space into a meshgrid. These coordinates are then transformed using various camera parameters to obtain 3D world space coordinates. By inputting the extracted 2D image features and 3D coordinates into a 3D position encoder, PETR generates 3D position-aware features that interact with object queries in the transformer decoder. The updated queries are then used to predict object classes and 3D bounding boxes.

Despite such advancements, few studies have focused on improving the bipartite graph matching component for more efficient training. The slow convergence is primarily due to this discrete component, which is particularly unstable in early training stages because of stochastic optimization's inherent nature. As a result, a query often matches with different objects across various epochs for the same image, leading to ambiguous and inconsistent optimization.

To address this problem, we propose a novel training method that introduces a query denoising task to help stabilize bipartite graph matching during the training process. Building on previous works [4] that effectively interpret queries as reference points or anchor boxes containing positional information, we adopt their approach and use 3D anchor boxes as queries. Our solution involves feeding noised ground truth (GT) bounding boxes as noised queries, along with learnable anchor queries, into Transformer decoders. Both types of queries share the same input format of (x, y, z) and can be fed into Transformer decoders simultaneously.

For the noised queries, we perform a denoising task to reconstruct their corresponding GT boxes. For the other learnable anchor queries, we use the same training loss and bipartite matching as in the vanilla PETR. Since the noised bounding boxes bypass the bipartite graph matching component, the denoising task serves as an easier auxiliary task, helping PETR mitigate the instability of discrete bipartite matching and accelerate bounding box prediction. Additionally, the denoising task lowers the optimization difficulty because the introduced random noise is typically small.

In summary, our method is a denoising training approach. Our loss function consists of two components. One is a

reconstruction loss and the other is a Hungarian loss which is the same as in PETR methods. We summarize our contribution as follows:

- We design a novel training method to speed up PETR training. Experimental results show that our method not only accelerates training convergence but also leads to a remarkably better training result. Moreover, our method shows a remarkable improvement over our baseline PETR.

II. Why does denoising do in training?

A. Stablize Hungarian Mathcing

Hungarian matching is a well-known algorithm in graph matching that, given a cost matrix, outputs an optimal matching result. DETR is the first algorithm to incorporate Hungarian matching in object detection, addressing the challenge of matching predicted objects to ground-truth objects. DETR transforms ground-truth assignment into a dynamic process, introducing an instability issue due to its discrete bipartite matching and the stochastic nature of the training process. Studies [5] have shown that Hungarian matching does not always yield stable results, as blocking pairs can exist. Even minor changes in the cost matrix can lead to significant shifts in the matching outcome, causing inconsistent optimization goals for decoder queries.

We conceptualize the training of DETR-like models as comprising two stages: learning "good anchors" and learning "relative offsets." Decoder queries are tasked with learning these anchors. Inconsistent updates to anchors can impede the learning of relative offsets. To address this, our method employs a denoising task as a training shortcut to facilitate the learning of relative offsets, effectively bypassing bipartite matching.

In our approach, each decoder query is interpreted as a 3-D anchor box. A noisy query can be seen as a "good anchor" with a nearby corresponding ground-truth box. The denoising training then has a clear optimization goal: to predict the original bounding box, thereby avoiding the ambiguity introduced by Hungarian matching.

B. Make Query Search More Locally

We also demonstrate that DN-PETR can enhance detection by minimizing the gap between anchors and their corresponding targets. Visualization of DETR reveals that its positional queries exhibit various operational modes, leading to a broad search range for predicted boxes. In contrast, DN-PETR exhibits significantly smaller mean distances between initial anchors (positional queries) and targets.

Through denoising training, the model learns to reconstruct boxes from noisy inputs that closely resemble the ground truth. Consequently, the model tends to search more locally for predictions, causing each query to focus on nearby regions and mitigating potential conflicts in predictions between queries.

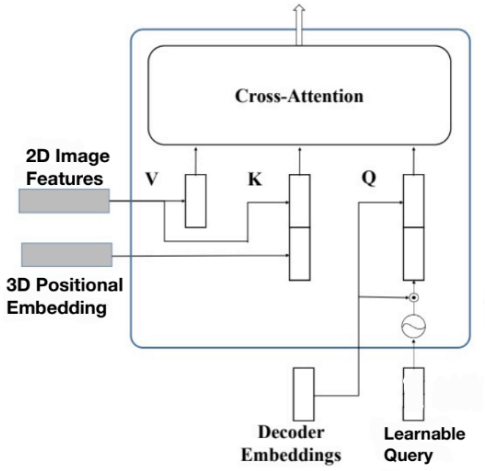

Fig. 1. (a) Cross-attention in decoder of PETR, vanilla PETR directly use learnable queries.

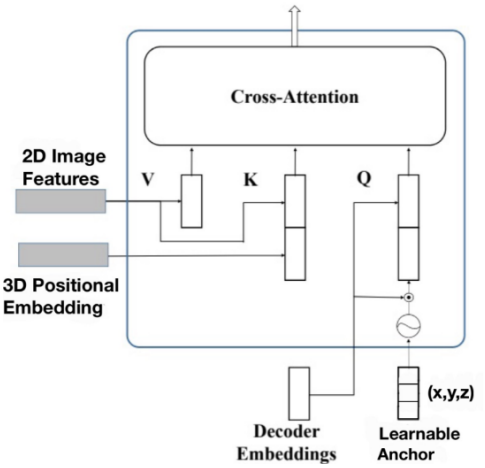

Fig. 2. (b) Cross-attention in decoder of DN-PETR, DN-DETR specifies use the learnable anchors.

III. DN-PETR

A. Overview

We base on the architecture of PETR to implement our training method. We explicitly formulate the decoder queries as box coordinates. The only difference between our architecture and PETR lies in the learnable Query shows in Fig 1 and 2.

Similar to PETR, our architecture contains a Transformer encoder and a Transformer decoder. On the encoder side, the 2D image features are extracted with a CNN backbone(VoVNet) and then combine with 3D positional embedding to formed the 3D positional-aware Features before fed into the Transformer encoder. On the decoder side, we replace the Learnable Query with Learnable Anchor and then fed into the decoder to search for objects through cross-attention.

We denote decoder queries as $\mathbf{q} = \{q_0, q_1, ..., q_{n-1}\}$ and the output of the Transformer decoder as $\mathbf{o} = \{o_0, o_1, ..., o_{n-1}\}$. We also use F and A to denote the refined image features after the Transformer encoder, and the attention mask derived based

on the denoising task design. We can formulate our method as follows.

$$\mathbf{o} = D(\mathbf{q}, F \mid A) \tag{1}$$

where D denotes the Transformer decoder. Decoder queries consist of two components: the matching part and the denoising part. In the matching part, learnable anchors, akin to those used in PETR, serve as inputs. Here, bipartite graph matching is employed to pair predicted truth-box label pairs with corresponding ground truth pairs. In contrast, the denoising part utilizes as inputs noisy ground-truth (GT) box-label pairs and learnable anchors. Unlike the matching part, the denoising component does not necessitate bipartite graph matching. For convenience, we denote the denoising part as $\mathbf{q} = \{q_0, q_1, ..., q_{n-1}\}$ and the matching part as $\mathbf{Q} = \{Q_0, Q_1, ..., Q_{L-1}\}$. Thus, the formulation of our method can be expressed as follows:

$$\mathbf{o} = D(\mathbf{q}, \mathbf{Q}, F \mid A) \tag{2}$$

Additionally, we employ an attention mask to safeguard against information leakage from the denoising part to the matching part and among different noisy versions of the same ground truth object.

B. Anchor

We explicitly represent each query as 3D anchor coordinates. A query is defined as a tuple (x, y, z), where x, y, z denote the center coordinates of each box. Moreover, the anchor coordinates are dynamically updated across layers. At the output of each decoder layer, a tuple $(\Delta x, \Delta y, \Delta z)$ is generated, and the anchor is updated to $(x+\Delta x, y+\Delta y, z+\Delta z)$.

C. Denoising

For each image, we collect all GT objects and add random noises to both their bounding boxes. We consider adding noise to boxes by center shifting. In center shifting, we add a random noise $(\Delta x, \Delta y, \Delta z)$ to the box center and make sure that $|\Delta x| < \frac{\lambda_1 \cdot x}{2}$, $|\Delta y| < \frac{\lambda_1 \cdot y}{2}$ and $|\Delta z| < \frac{\lambda_1 \cdot z}{2}$, where $\lambda_1 \in (0, 1)$ to make sure that the noised box will still be inside the original bounding box. Notice that denoising is only considered in training, during inference the denoising part is removed, leaving only the matching part.

D. Attention mask

The attention mask plays a crucial role in our model. Without it, the denoising training could hinder performance rather than enhancing it. To incorporate the attention mask, we begin by grouping the noisy ground truth (GT) objects. Each group consists of various noisy versions of the same GT object. Subsequently, the denoising part transforms into:

$$\mathbf{q} = \{g_0, g_1, ..., g_{P-1}\} \tag{3}$$

where g_p is defined as the p-th denoising group. Each denoising group contains M queries where M is the number of GT objects in the image. So we have

$$\mathbf{g_p} = \{q_0^p, q_1^p, ..., q_{M-1}^p\} \tag{4}$$

where $q_m^p = \delta(t_m)$.

The purpose of the attention mask is to prevent information leakage. There are two types of potential information leakage. One is that the matching part may see the noised GT objects and easily predict GT objects. The other is that one noised version of a GT object may see another version. Therefore, our attention mask is to make sure the matching part cannot see the denoising part and the denoising groups cannot see each other.

We use $\mathbf{A} = [a_{ij}]_{W \times W}$ to denote the attention mask where $W = P \times M + N$. P and M are the number of groups and GT objects. N is the number of queries in the matching part. We let the first $P \times M$ rows and columns represent the denoising part and the latter represents the matching part. $a_{ij} = 1$ means the i-th query cannot see the j-th query and $a_{ij} = 0$ otherwise. We devise the attention mask as follows

$$a_{ij} = \begin{cases} 1, & \text{if } j < P \times M \text{ and } \left\lfloor \frac{i}{M} \right\rfloor \neq \left\lfloor \frac{j}{M} \right\rfloor; \\ 1, & \text{if } j < P \times M \text{ and } i \geq P \times M; \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Note that whether the denoising part can see the matching part or not will not influence the performance, since the queries of the matching part are learned queries that contain no information about the GT objects.

IV. Experiment

A. Parameter configuration

Similar to PETR, DN-PETR applies AdamW optimizer with weight decay of 0.01. And the learning rate is a little different from the original model,which is 2.0×10^{-4}. For DN, we use smaller learning rate 1.0×10^{-4} since loss of this model will be larger compared to PETR due to adding noise. Also, the learning rate decayed with cosine annealing policy like PETR as well. For training part, we use 4 NVIDIA GetForce RTX-3090 to run the training process. The total consuming time for PETR and DN-PETR are 50 hours and 52 hours respectively, total epoch is 24.

B. Metrics

In this paper, several metrics for 3D object detection are used, such as mAP, mATE, mASE, mAOE, NDS, etc. Also, since the key points lying in DN-PETR is the faster convergence speed. We will also provide the metric of DN-PETR and PETR based on different training epoch number.

C. Result

Firstly, Table I below demonstrates that DN-PETR can uphold PETR's high performance even when incorporating noise to expedite convergence.

Secondly, Table II below records the accuracy throughout the training process. It's evident that before 20 epochs, DN-PETR achieves higher accuracy, indicating that our approach accelerates the training procedure.

TABLE I
METRICS OF PETR AND DN-PETR

Model	mAP(%)	mATE	mASE	mAOE	NDS(%)
PETR	40.40	0.7541	0.2936	0.3951	45.50
DN-PETR(ours)	40.36	0.7557	0.2963	0.4015	45.98

TABLE II
MAP(%) V.S EPOCH NUMBER

Model/Epoch	6	12	18	24
PETR	33.89	35.11	38.62	40.40
DN-PETR(ours)	36.53	39.85	40.17	40.36

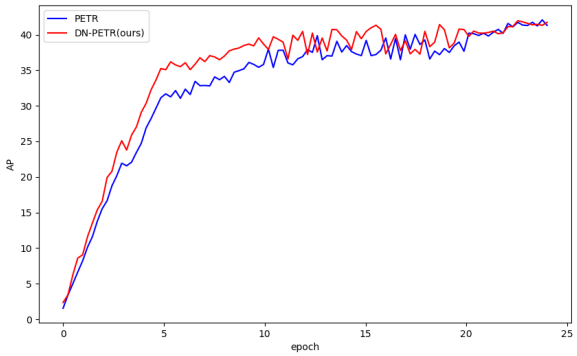

Fig. 3. Convergence curves of PETR and DN-PETR

We additionally present the convergence curves for both vanilla PETR and DN-PETR in Fig 3 for clearer visualization. It's evident that DN-PETR consistently outperforms PETR throughout most of the training duration, particularly in the initial phase. PETR demonstrates faster attainment of higher accuracy. Before the learning rate reduction, DN-PETR achieves an average precision (AP) of 35 in just 5 epochs, whereas PETR requires 12 epochs to reach the same milestone.

Below, we provide visualizations of object detection on the NuScenes dataset, featuring two sets of data, each with three figures: the ground truth, PETR results, and DN-PETR results. We observe that DN-PETR detects more objects than the vanilla PETR. For instance, in Fig. 9 "CAM_FRONT_LEFT," our model identifies an object (blue box) that is not detected in the PETR results, demonstrating its superior performance over the vanilla PETR model.

Figures 5 and 6 demonstrate that both models exhibit the same level of predictive accuracy, but with reduced training time. Faster training times make it more feasible to scale up solution development, which is especially crucial for autonomous vehicle companies. These companies often need to train multiple models simultaneously or retrain models frequently as new data emerges.

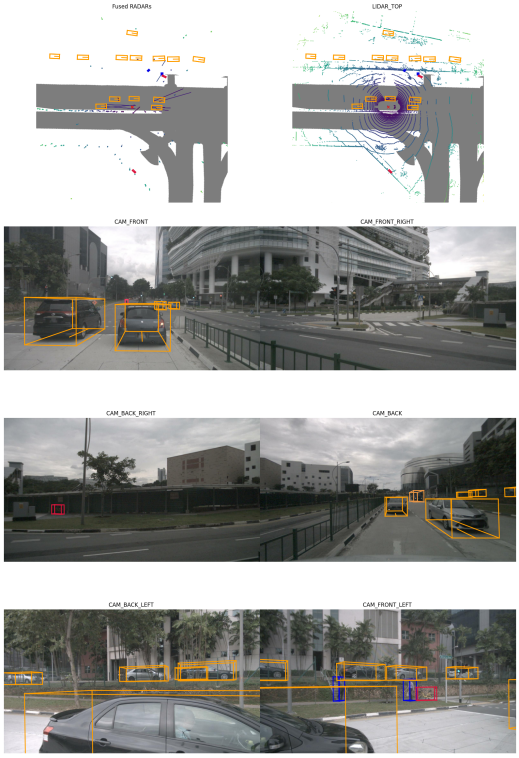

Fig. 4. Ground Truth I

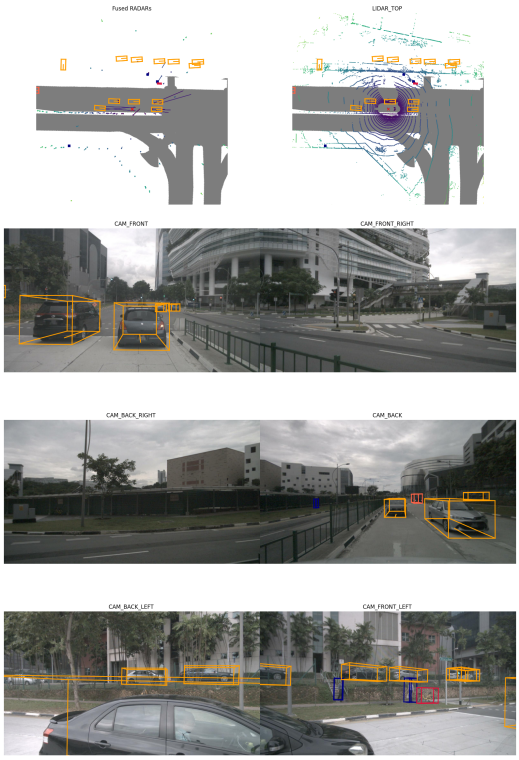

Fig. 5. Result of PETR I

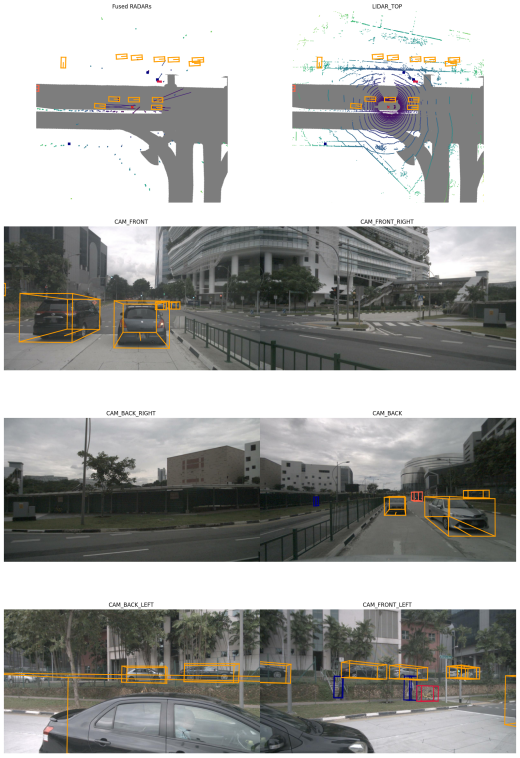

Fig. 6. Result of DN-PETR I

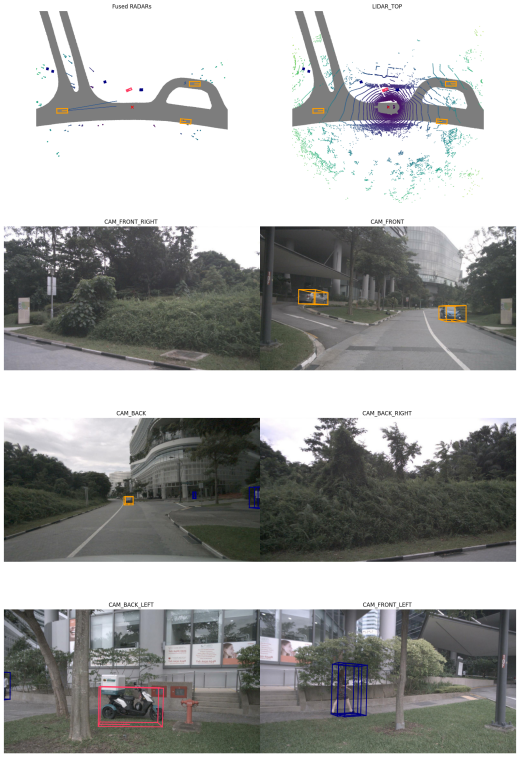

Fig. 8. Result of PETR II

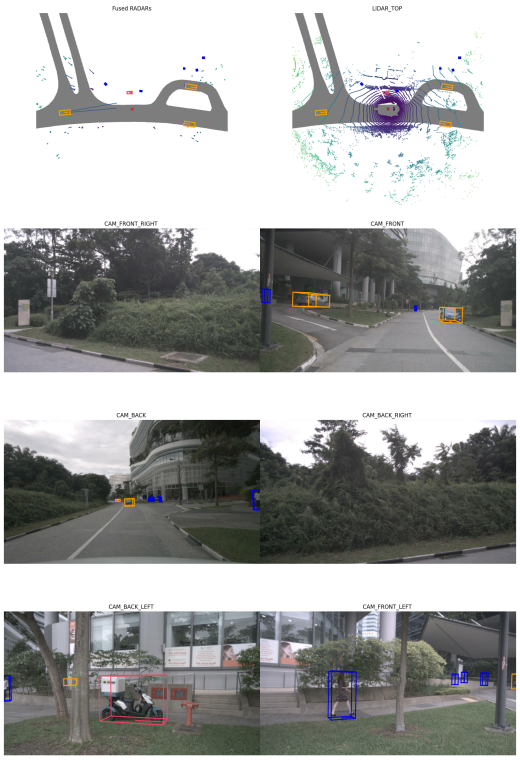

Fig. 7. Ground Truth II

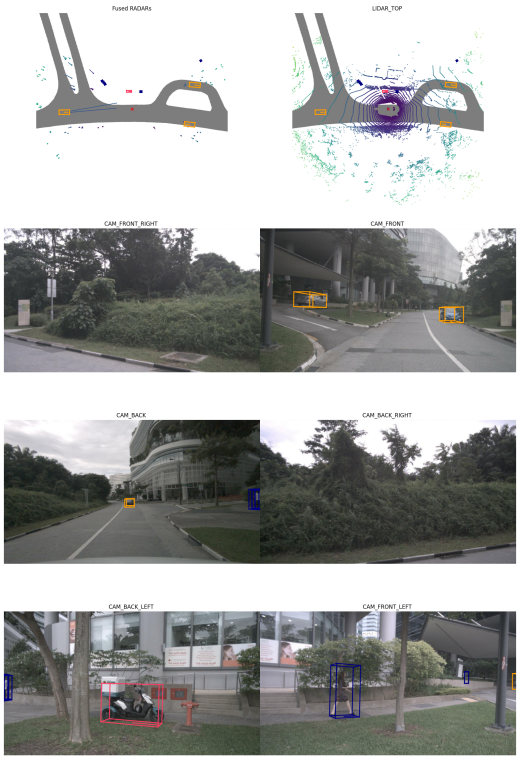

Fig. 9. Result of DN-PETR II

V. Implementation

In this project, our primary objective is to enhance the performance of the current PETR model. To ensure a fair comparison and build upon established work, we based our vanilla PETR implementation on existing, community-vetted code. This approach not only saves development time but also allows us to directly measure our improvements against the original model.

The core of our innovation lies in accelerating PETR's training convergence. To achieve this, we designed and implemented a novel denoising training module entirely from scratch. Our key technical contribution is the transformation of object queries into learnable anchors. In traditional PETR, object queries are static and learn to predict object locations through attention mechanisms. By modifying these queries to be learnable anchors, we introduce spatial priors that can adapt during training. This change helps stabilize the bipartite matching process, which we identified as the root cause of PETR's slow convergence.

By grounding our work in existing code while introducing novel components, we demonstrate both our respect for prior research and our ability to innovate. Our project showcases how targeted modifications, even to fundamental elements like object queries, can significantly improve model performance without overhauling the entire architecture.

VI. Future goal

Initially, DN-PETR was trained on the KITTI dataset, which only includes 2 views from 2 RGB cameras. Reasonably, the result is much worse compared to using 8-view on the NuScenes dataset. Hence, one of the extensions of this project is to maintain or even outperform the current model while decreasing the number of cameras, which is good news for industrial manufacturers to reduce the budget. Furthermore, since the original author already released PETRV2, then it is worth trying combining DN-mechanism to such a model and even coming up with a way for PETRV3. Since the pure-vision autonomous vehicle technique is already shown in the world, more applications of multi-view for object detection will prevail recently.

VII. Conclusion

In this study, we investigated the root cause of PETR's slow convergence during training, attributing it to unstable bipartite matching. To address this issue, we introduced a novel denoising training method. Building on this analysis, we developed DN-PETR, which integrates denoising training into the PETR framework to test its effectiveness. Specifically, DN-PETR applies denoising training to bounding boxes. And other parts are done by mmcv, mmdet, and mmdet3d.

Our results demonstrate that denoising training significantly enhances both convergence speed and overall performance. This research reveals that denoising training can be seamlessly incorporated into the PETR model as a versatile training technique. Notably, this integration incurs only a minor increase in training cost while delivering substantial improvements in both training convergence and detection accuracy.

References

[1] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in European Conference on Computer Vision (ECCV), 2020, pp. 213-229.

[2] Wang, Y., Guizilini, V., Zhang, T., Wang, Y., Zhao, H., & Solomon, J. (2021). DETR3D: 3D object detection from multi-view images via 3D-to-2D queries. arXiv. https://doi.org/10.48550/arXiv.2110.06922

[3] Liu, Y., Wang, T., Zhang, X., & Sun, J. (2022). PETR: Position Embedding Transformation for Multi-View 3D Object Detection. ECCV 2022. arXiv:2203.05625v3.

[4] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M. Ni, Lei Zhang. DN-DETR: Accelerate DETR Training by Introducing Query DeNoising. CVPR 2022. arXiv:2203.01305v3

[5] Enrico Maria Fenoaltea, Izat B Baybusinov, Jianyang Zhao, Lei Zhou, and Yi-Cheng Zhang. The Stable Marriage Problem: An interdisciplinary review from the physicist's perspective. Physics Reports, 2021.