2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

MDHA: Multi-Scale Deformable Transformer with Hybrid Anchors for Multi-View 3D Object Detection

Michelle Adeline[1], Junn Yong Loo[1,2], Vishnu Monn Baskaran[1]

***Abstract*— Multi-view 3D object detection is a crucial component of autonomous driving systems. Contemporary query-based methods primarily depend either on dataset-specific initialization of 3D anchors, introducing bias, or utilize dense attention mechanisms, which are computationally inefficient and unscalable. To overcome these issues, we present MDHA, a novel sparse query-based framework, which constructs adaptive 3D output proposals using hybrid anchors from multi-view, multi-scale image input. Fixed 2D anchors are combined with depth predictions to form 2.5D anchors, which are projected to obtain 3D proposals. To ensure high efficiency, our proposed Anchor Encoder performs sparse refinement and selects the top-k anchors and features. Moreover, while existing multi-view attention mechanisms rely on projecting reference points to multiple images, our novel Circular Deformable Attention mechanism only projects to a single image but allows reference points to seamlessly attend to adjacent images, improving efficiency without compromising on performance. On the nuScenes val set, it achieves 46.4% mAP and 55.0% NDS with a ResNet101 backbone. MDHA significantly outperforms the baseline where anchor proposals are modelled as learnable embeddings. Code is available at `https://github.com/NaomiEX/MDHA`.**

I. INTRODUCTION

Multi-view 3D object detection plays a pivotal role in mapping and understanding a vehicle's surroundings for reliable autonomous driving. Among existing methods, camera-only approaches have gained immense traction as of late due to the accessibility and low deployment cost of cameras as opposed to conventional LiDAR sensors. Camera-only methods can be split into: Bird's-Eye-View (BEV) methods [1]–[6] where multi-view features are fused into a unified BEV representation, and query-based methods [7]–[17] where 3D objects are directly modelled as queries and progressively refined based on image features.

The construction of BEV maps from input features involves a non-trivial view transformation, rendering it computationally expensive. Moreover, BEV representations suffer from having a fixed perception range, constraining their adaptability to diverse driving scenarios. For instance, in scarcely populated rural areas with minimal visual elements of interest, constructing these dense and rich BEV maps is a waste of computational resources. In contrast, query-based methods bypass the computationally intensive BEV construction and have recently achieved comparable performance to BEV-based methods with straightforward sparsification.

Two predominant series of models have emerged within query-based approaches: the PETR-series [8]–[11], which generates 3D position-aware features from 2D image features via positional embeddings, and the Sparse4D-series [12]–[14], which refines 3D anchors using sparse feature sampling of spatio-temporal input. Sparse methods are attractive since they enable easy tuning to achieve the desired balance between performance and efficiency for real-world application. However, since PETR models employ a dense cross-attention mechanism, they cannot be considered as sparse methods. Moreover, they do not utilize multi-scale input features, limiting their scalability to detect objects of varying sizes. Although these are rectified by the Sparse4D models, they instead rely on anchors initialized from k-means clustering on the nuScenes [18] train set, thus potentially compromising their ability to generalize to real-world driving scenarios.

To address these issues, we propose a novel framework for camera-only query-based 3D object detection centered on hybrid anchors. Motivated by the effectiveness of 2D priors in 3D object detection [6], [15]–[17], we propose the

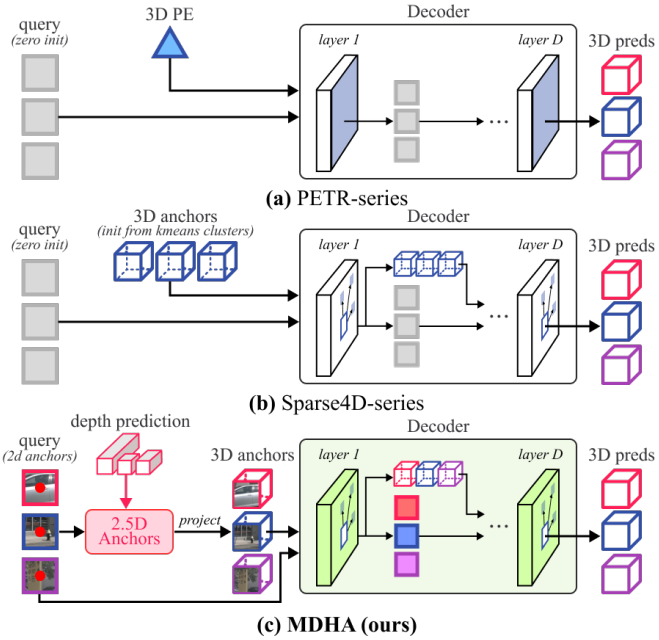

Fig. 1. **Comparison between PETR and Sparse4D models with our proposed architecture.** (a) PETR models encode 3D information via positional embeddings and refine queries using dense attention within their decoder. (b) Sparse4D models initialize 3D anchors from k-means clustering on nuScenes and iteratively refine both queries and anchors using sparse attention within the decoder. (c) MDHA employs image tokens as queries; the (x, y) center coordinates of each token, paired with depth predictions, form 2.5D anchors, which are projected into 3D anchors. This eliminates the need for good anchor initialization. Anchors and queries undergo iterative refinement using multi-view-spanning sparse attention within the decoder.

[1]The authors are with the School of Information Technology, Monash University Malaysia (e-mail: made0008@student.monash.edu, [loo.junnyong,vishnu.monn]@monash.edu).
[2]Corresponding author

formation of 2.5D anchors by pairing each token's 2D center coordinates with corresponding depth predictions. These anchors can be projected, with known camera transformation matrices, to generate reasonable 3D output proposals. Our usage of multi-scale, multi-view input leads to a large number of tokens, and thus, of 3D proposals, posing significant computational burden for refinement. To alleviate this, our proposed Anchor Encoder serves the dual-purpose of refining and selecting top-k image features and 3D proposals for subsequent iterative refinement within the spatio-temporal MDHA Decoder. On top of this, to improve efficiency, we propose a novel Circular Deformable Attention (CDA) mechanism, which treats multi-view input as a contiguous 360° panoramic image. This allows reference points to seamlessly attend to locations in adjacent images. Thus, our proposed method eliminates the reliance on good 3D anchor initialization, leverages multi-scale input features for improved detection at varying scales, and improves efficiency through our novel sparse attention mechanism, which offers greater flexibility than existing multi-view attention mechanisms, where attention is confined to the image in which the reference point is projected. Figure 1 illustrates the comparison between our method and the PETR and Sparse4D models.

To summarize, our main contributions are as follows:

- We propose a novel framework for sparse query-based 3D object detection from multi-view cameras, MDHA, which constructs adaptive and diverse 3D output proposals from 2D→2.5D→3D anchors. Top-k proposals are sparsely selected in the Anchor Encoder and refined within the MDHA decoder, thereby reducing reliance on 3D anchor initialization.
- An elegant multi-view-spanning sparse attention mechanism, Circular Deformable Attention, which improves efficiency without compromising performance.
- On the nuScenes val set, MDHA significantly outperforms the learned anchors baseline, where proposals are implemented as learnable embeddings, and surpasses most state-of-the-art query-based methods.

II. RELATED WORK

A. Camera-Only Multi-View 3D Object Detection

BEV-based methods perform 3D object detection by leveraging a Bird's-Eye-View feature representation acquired by transforming image features. In most works [1]–[4], this transformation follows the Lift-Splat paradigm [1], which involves "lifting" image features into 3D space using depth predictions, and "splatting" them onto the BEV plane by fusing features that fall into the pre-defined grids. BEVDet [2] constructs a BEV map through this view transformation, refines it with a BEV-Encoder and generates predictions with a detection head. BEVDet4D [3] extends this framework by incorporating temporal features which are projected onto the current frame, while BEVDepth [4] introduces explicit depth supervision with a camera-aware depth estimation module. Meanwhile, BEVFormer [5] models BEV pillars as dense queries and utilizes deformable attention to aggregate spatio-temporal information for BEV refinement.

In contrast, query-based methods circumvent the complex construction of BEV maps by modelling objects implicitly as queries. DETR3D [7] spearheaded this class of methods by learning a 3D-to-2D projection of predicted 2D object centers, yielding reference points which, in conjunction with sampled image features, are employed for query refinement in the decoder. Despite being a representative sparse query-based method, it suffers from poor performance as it neglects temporal information. Sparse4D [12] rectifies this by projecting 4D keypoints onto multi-view frames across multiple timestamps. Sparse4Dv2 [13] improves both its performance and efficiency by adopting a recurrent temporal feature fusion module. PETR [8] diverges from DETR3D by encoding 3D spatial information into input features via positional embedding, eliminating the need for 3D-to-2D projection. PETRv2 [9] extends this framework to other 3D perception tasks, namely BEV segmentation and lane detection, while Focal-PETR [10] adds a focal sampling module which selects discriminative foreground features and converts them to 3D-aware features via spatial alignment. Finally, StreamPETR [11] demonstrates impressive performance by adopting an object-centric temporal fusion method which propagates top-k queries and reference points from prior frames into a small memory queue for improved temporal modelling. Inspired by the aforementioned models, our query-based method adopts object-centric recurrent temporal fusion and employs 3D-to-2D projections in our attention mechanism. However, our method uniquely builds up adaptive 3D output proposals from image features rather than using learnable embeddings.

B. Depth and 2D Auxiliary Tasks for 3D Object Detection

In an emerging trend, more frameworks are integrating depth or 2D modules for auxiliary supervision [10], [11], [13], [14] or for the final 3D prediction [6], [15]–[17]. Sparse4Dv2 [13] and Sparse4Dv3 [14] both implement auxiliary dense depth supervision to improve training stability while Focal-PETR [10] and StreamPETR [11] both utilize auxiliary 2D supervision for the same purpose.

In contrast, certain methods have integrated the outputs of these auxiliary modules into the final 3D prediction. For instance, BEVFormer v2 [6] proposes a two-stage detector, featuring a perspective head that suggests proposals used within the BEV decoder. SimMOD [15] generates proposals for each token via four convolutional branches, predicting the object class, centerness, offset, and depth, where the first two are used for proposal selection and the latter two predict the 2.5D center. MV2D [16] employs a 2D detector to suggest regions of interest, from which aligned features and queries are extracted for its decoder. Far3D [17] focuses on long-range object detection by constructing 3D adaptive queries from 2D bounding box and depth predictions. Our approach differs from existing works by streamlining the auxiliary module, requiring only depth predictions. By pairing depth values with the 2D coordinates of each feature token, we bypass the need to predict 2D anchors, reducing the learning burden of

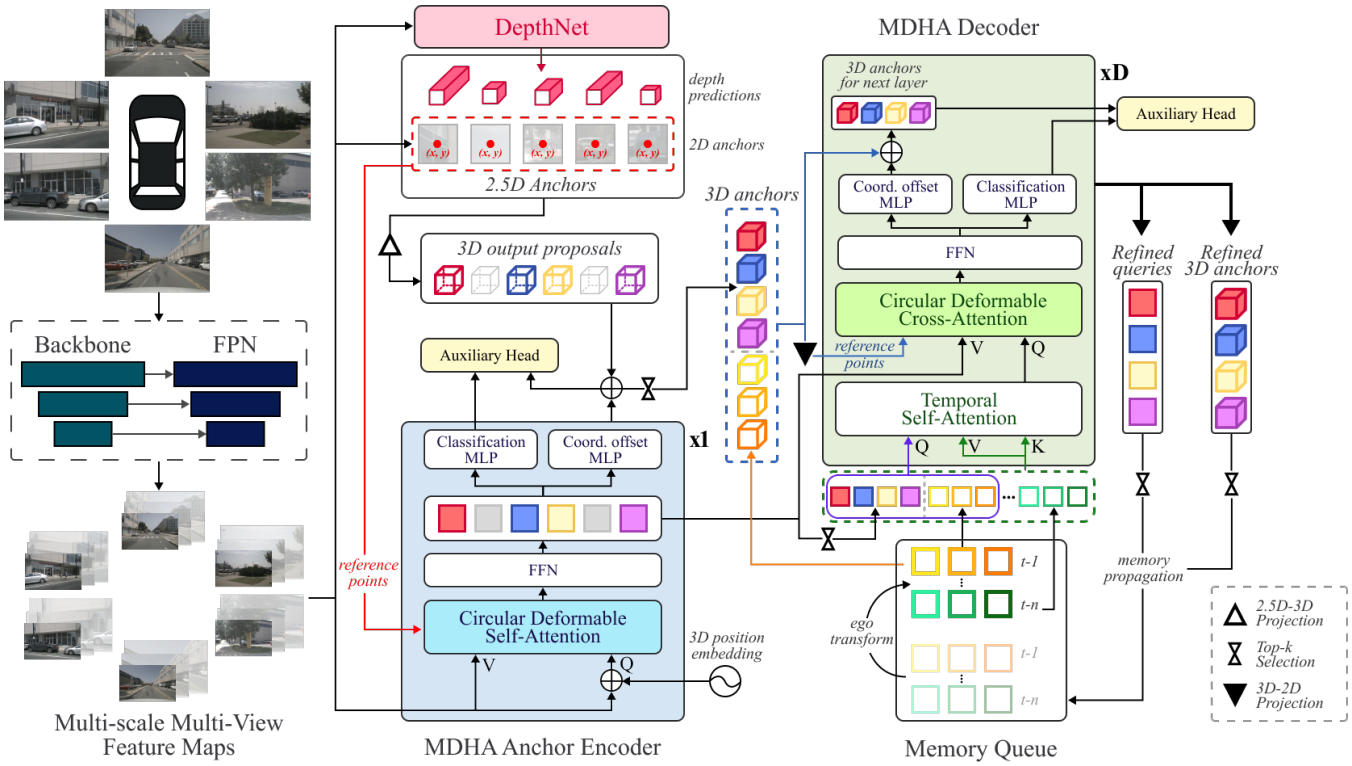

Fig. 2. **Our proposed MDHA Architecture**. Multi-view images are fed into the backbone and FPN neck to extract multi-scale, multi-view image features. These feature tokens serve as input for the DepthNet, which pairs their 2D center coordinates with predicted depth, forming 2.5D anchors, which are projected to obtain 3D output proposals. The tokens and proposals undergo refinement in the 1-layer MDHA Anchor Encoder. It also selects the top-k queries and proposals for further refinement in the MDHA Decoder, which additionally considers temporal information via the memory queue.

the model and improving efficiency. Furthermore, proposal selection, feature aggregation, and sparse refinement are all executed by a shallow transformer encoder.

III. METHOD

A. Overview

Figure 2 shows the overall architecture of MDHA. Given N-view RGB images, the backbone and Feature Pyramid Network (FPN) neck extracts multi-scale feature maps, $\{F_l\}_{l=1}^{L}$, where $F_l \in \mathbb{R}^{N\times C\times H_l\times W_l}$, C denotes the feature dimension, and (H_l, W_l) refers to the height and width of the feature map at level l. For each feature token (feature map cell), the DepthNet constructs 2.5D anchors, which are projected to obtain 3D output proposals (Section III-B). The single-layer MDHA Anchor Encoder refines and selects the top-k features and proposals to pass onto the decoder (Section III-C). The D-layer MDHA Decoder conducts iterative anchor refinement using spatio-temporal information (Section III-D). Within the model, CDA is employed as an efficient multi-view-spanning sparse attention mechanism (Section III-E). The model is trained end-to-end with detection and classification (final and auxiliary) losses, with explicit depth supervision (Section III-F).

B. DepthNet

Constructing reasonable proposals for 3D object detection is not an easy feat. While a naive approach involves parameterizing initial anchors, making them learnable, the large search space makes this a non-trivial task, and sub-optimal anchors run the risk of destabilizing training and providing inadequate coverage of the perception range (Section IV-D.1). Although Sparse4D avoids this issue by initializing 900 anchors from k-means clustering on the nuScenes train set, this introduces bias towards the data distribution present in nuScenes. Thus, with the aim of generating robust proposals for generalized driving scenarios, we opt to use the 2D center coordinates of each feature token to construct a set of 2.5D anchors. For the i-th token, we define the 2.5D anchor as:

$$A_i^{2.5D} = [x_i^{2D}, y_i^{2D}, z_i]^T \tag{1}$$

where $(x_i^{2D}, y_i^{2D}) = (\frac{x_i+0.5}{W_l}\cdot W_{inp}, \frac{y_i+0.5}{H_l}\cdot H_{inp})$ are the 2D coordinates of the token's center, z_i is the predicted depth of the object center, and (W_{inp}, H_{inp}) is the input image resolution. By fixing the (x, y) coordinates, the model only focuses on learning the depth value, reducing the search space from $\mathbb{R}^3$ to $\mathbb{R}^1$. Furthermore, since each feature token covers a separate part of the image, the resulting 3D anchors are naturally well-distributed around the ego vehicle.

These 2.5D anchors can then be projected into 3D output proposals using E_i and I_i, the camera extrinsic and intrinsic matrices, respectively, for the view in which the token belongs to:

$$A_i^{3D} = E_i^{-1} I_i^{-1} [x_i^{2D} * z_i, y_i^{2D} * z_i, z_i, 1]^T \tag{2}$$

The DepthNet assumes that an object is centered within a

token and predicts its depth relative to the ego vehicle. Below, we examine two approaches in obtaining depth suggestions:

Fixed Depth. Figure 3 shows a clear pattern in the depth distribution within each camera: depth increases as we travel up the image. Motivated by this observation, we sample z_i from this distribution, eliminating the need for the model to predict the depth entirely. We argue that this is generalizable since, in most driving scenarios, all objects are situated on a level plane. Hence, for an object center to appear higher up on the image, it must be further away from the ego vehicle.

Learnable Depth. We also explore a more adaptive approach for obtaining depth maps, $\{z_l\}_{l=1}^{L}$, where $z_l \in \mathbb{R}^{N\times C\times H_l\times W_l}$ is predicted as follows:

$$z_l = \sigma(\mathbf{\Psi}(F_l)) * (D^{max} - D^{min}) + D^{min} \tag{3}$$

where $\mathbf{\Psi}$ is a single layer convolutional head and D^{max}, D^{min} are the maximum and minimum depth values.

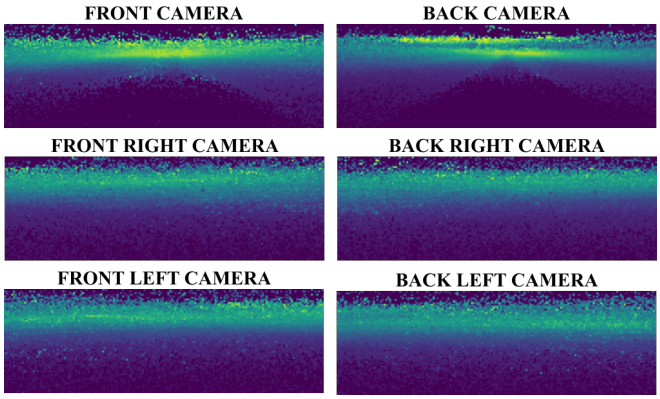

Fig. 3. **Depth distribution of 3D object centers projected onto all 6 cameras in the nuScenes train set.** The brighter the colour, the farther away the object. The semicircular voids at the bottom of the front and back cameras represent the front and rear part of the vehicle which juts out and where no objects can be located.

C. MDHA Anchor Encoder

Within the encoder, each image token acts as a query and undergoes refinement via self-attention using our sparse CDA mechanism, where (x_i^{2D}, y_i^{2D}) serve as 2D reference points for each query. The addition of 3D position embeddings from PETR was found to stabilize training by providing 3D positional context for the 2D image tokens. We also employ $(x_i^{2D}, y_i^{2D}, z_i)$ sinusoidal position embeddings to encode the token's learned 2.5D anchor. For each token, two Multilayer Perceptron (MLP) heads predict the object classification score and offsets to refine the token's corresponding 3D proposal. The resulting proposals with the top-k classification scores, A_{enc}^{3D}, are selected for further refinement within the decoder. We follow DINO [19] in initializing the decoder queries as the top-k queries from the encoder, Q_{enc}.

D. MDHA Decoder

The decoder utilizes temporal information by leveraging a short memory queue consisting of sparse features and anchors from the last m frames. To account for ego vehicle movement between frames, historic 3D anchors, $A_{t-\tau}^{3D} = (x^{3D}, y^{3D}, z)_{t-\tau}$, are aligned to the current frame as follows:

$$A_{t-\tau\to t}^{3D} = \text{EGO}_t^{-1} \cdot \text{EGO}_{t-\tau}[A_{t-\tau}^{3D}, 1]^T \tag{4}$$

where EGO_t is the lidar-to-global transformation matrix at time t.

Relevant historic features are then selected via the Temporal Self-Attention module which is implemented as vanilla multi-head attention [20] where the features from the memory queue serve as the key and $value$, while the $query$ consists of (Q_{enc}, Q_{t-1}), with Q_{t-1} being the query propagated from the previous frame. Due to the fixed-length memory queue, this operation's computational complexity does not scale with increasing image resolution and since memory size $<<$ feature map size, it is dominated by the encoder's complexity which *does* scale with feature map size. Thus, sparse attention is unnecessary for this operation.

In the Circular Deformable Cross-Attention module, these selected queries efficiently attend to refined image features from the encoder via our CDA mechanism. Its reference points for decoder layer d, are obtained by projecting 3D anchors, A^{d-1}, from the previous layer, into 2D. For the first layer, these anchors are defined as $A^0 = [A_{enc}^{3D}, A_{t-1\to t}^{3D}]$, and the 3D-to-2D projection for view n is given by:

$$\text{Proj}(A^{3D}, n) = I_n \cdot E_n[A^{3D}, 1]^T \tag{5}$$

Since a 3D point can be projected to multiple camera views, we either contend with a one-to-many relationship between queries and reference points or choose a single projected point via a heuristic. For this work, we opt for the latter as it is more efficient, and with our novel CDA mechanism, performance is not compromised. Our chosen heuristic involves selecting the point that is within image boundaries and is closest to the center of the view it is projected. Thus, our reference points are given by:

$$(r_x^d, r_y^d) = \underset{(r_x,r_y)\in\mathcal{R}}{\arg\min} \|[r_x, r_y]^T - [W_{inp}/2, H_{inp}/2]^T\|_2 \tag{6}$$

where $\mathcal{R} = \{\text{Proj}(A^{d-1}, n)\}_{n=1}^{N} \setminus \mathcal{R}_\oslash$ is the set of projected 2D points, excluding points outside image boundaries, $\mathcal{R}_\oslash$.

For each query, two MLP heads predict classification scores and offsets, ΔA^d, which are used to obtain the refined anchors $A^d = A^{d-1} + \Delta A^d$ for each decoder layer. Anchors and queries with the top-q scores are propagated into the memory queue. For improved efficiency, intermediate classification predictions are omitted during inference time.

E. Circular Deformable Attention (CDA)

Self-attention in the encoder and cross-attention in the decoder utilize multi-scale, multi-view image features as attention targets. Due to the large number of feature tokens, executing these operations using vanilla multi-head attention is computationally expensive. Instead, we employ a modification of the deformable attention mechanism [21]. A straightforward implementation of deformable attention in a multi-view setting is to treat each of the N views as separate images within the batch, then project each

3D anchor into one or more 2D reference points spread across multiple views as is done in most existing works [7], [12]–[14]. However, this approach limits reference points to only attend to locations within their projected image. We overcome this limitation by concatenating the N views into a single contiguous 360° image. Thus, our feature maps are concatenated horizontally as $\mathcal{M} = \{M_l\}_{l=1}^{L}$, where $M_l = [F_{l1}, F_{l2}, \ldots, F_{lN}]$ and F_{ln} represents the input features at level l for view n.

Given 2D reference points, $r_x \in [0, W_{inp}], r_y \in [0, H_{inp}]$ local to view n, we can obtain normalized global 2D reference points, $\hat{r}^{cda} = (\hat{r}_x^{cda}, \hat{r}_y^{cda}) \in [0,1]^2$, for $\mathcal{M}$ as follows:

$$\hat{r}_x^{cda} = \frac{r_x + (n-1)W_{inp}}{N \times W_{inp}}, \quad \hat{r}_y^{cda} = \frac{r_y}{H_{inp}} \tag{7}$$

Letting q_i be the i-th query, CDA is then formulated as:

$$\text{CDA}(q_i, \hat{r}_i^{cda}, \mathcal{M}) = \sum_{h=1}^{N_h} W_h \sum_{l=1}^{L} \sum_{s=1}^{S} A_{hlis} \cdot W_h' x_l(p_{hlis}) \tag{8}$$

where h indexes the attention head with a total of N_h heads, and s indexes the sampling location with a total of S locations. $W_h \in \mathbb{R}^{C \times (C/N_h)}$, $W_h' \in \mathbb{R}^{(C/N_h) \times C}$ are learnable weights, and $A_{hlis} \in [0,1]$ is the predicted attention weight. Here, $x_l(p_{hlis})$ refers to the input feature sampled via bilinear interpolation from sampling location $p_{hlis} = \phi(\hat{r}_i^{cda} + \Delta p_{hlis})$, where ϕ scales the value to the feature map's resolution, and Δp_{hlis} is the sampling offset obtained as follows:

$$\Delta p_{hlis} = \mathbf{\Phi}(q_i) \,/\, (W_l^{\mathcal{M}}, H_l^{\mathcal{M}}) \tag{9}$$

with $\mathbf{\Phi}$ denoting the linear projection. Since $(W_l^{\mathcal{M}}, H_l^{\mathcal{M}}) = (N \times W_l, H_l)$, Δp_{hlis} is not bound by the dimensions of the view it is projected onto, inherently allowing the model to learn sampling locations beyond image boundaries. Considering that the $\mathbf{\Phi}(q_i)$ is unbounded, the offset might exceed the size of the feature map at that level. Therefore, we wrap sampling points around as if the input were circular:

$$\mathring{p}_{hlis} = p_{hlis} \bmod 1.0 \tag{10}$$

In effect, this treats the first and last view as if they are adjacent. CDA works best if visual continuity is maintained between neighbouring images. Thus, we reorder inputs to follow a circular camera order, i.e., Front → Front-Right → Back-Right → Back → Back-Left → Front-Left. This arrangement places related features next to one another. No other camera calibrations were done.

Since autonomous vehicles require a 360° view for safe navigation, CDA is highly viable for real-world application. Furthermore, given that CDA only involves reshaping input features and inexpensive pre-processing of reference points, it adds negligible overhead on top of vanilla deformable attention. Figure 4 illustrates the CDA mechanism.

F. Training Loss

We define a 3D detection as follows:

$$\left[x^{3D},\ y^{3D},\ z,\ w,\ l,\ h,\ \theta,\ v^x,\ v^y\right]$$

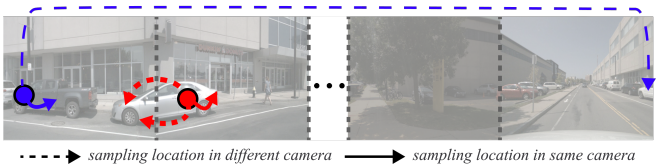

Fig. 4. **CDA on horizontally concatenated input**. The circles denote reference points and the arrows represent sampling locations.

consisting of the object's 3D center, width, length, height, yaw, and its x and y velocity. Let $\{\tilde{b}_j\}_{j=1}^{N_{gt}}$ and $\{\tilde{z}_j\}_{j=1}^{N_{gt}}$ be the set of ground-truth 3D detections and depths, while $\{\hat{b}_i\}_{i=1}^{N_q}$ and $\{\hat{z}_k\}_{k=1}^{N_{tok}}$ represent the set of predicted 3D detections and depths, where $N_{tok} = \sum_{l=1}^{L} N \times H_l \times W_l$. Under this setting, MDHA is trained end-to-end to minimize:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{det} + \lambda_3 \mathcal{L}_{depth} \tag{11}$$

where the classification loss, $\mathcal{L}_{cls}$, is implemented using focal loss [22], and the detection loss is defined as $\mathcal{L}_{det} = \frac{1}{N_{gt}} \sum_{i=1}^{N_q} \mathbb{1}_{\{b_i^{\text{targ}} \neq \varnothing\}} |b_i^{\text{targ}} - \hat{b}_i|$, with the chosen ground-truth target b_i^{targ} of prediction $\hat{b}_i$ obtained via bipartite matching [23]. Auxiliary classification and detection losses are used in the encoder and intermediate decoder layers. To calculate the depth loss $\mathcal{L}_{depth}$, we project $\tilde{b}_j$ to all N views using (5) to obtain $\{(\tilde{x}_{jn}^{2D}, \tilde{y}_{jn}^{2D})\}_{n=1}^{N_j^{proj}}$, where $N_j^{proj} = N - N_{\oslash}$ and $N_{\oslash}$ refers to the number of projected 2D points outside image boundaries. The target of depth prediction $\hat{z}_k$ is denoted as $\tilde{z}_{\hat{m}}$, with index $\hat{m}$ obtained via:

$$\hat{m} = \underset{j \in [1, N_{gt}]}{\arg\min}\ D_{kj} \tag{12}$$

where $D_{kj} = \min_n \|[\hat{x}_k^{2D}, \hat{y}_k^{2D}]^T - [\tilde{x}_{jn}^{2D}, \tilde{y}_{jn}^{2D}]^T\|_1$. Finally, we define the depth loss in (11) as

$$\mathcal{L}_{depth} = \frac{1}{\sum_{j=1}^{N_{gt}} N_j^{proj} \times L} \sum_{k=1}^{N_{tok}} W_k |\tilde{z}_{\hat{m}} - \hat{z}_k| \tag{13}$$

Let $W_k^{decay} = e^{-D_{k\hat{m}}/\varepsilon}$ represent the exponentially decayed weight of prediction k, which decreases the further away the token's 2D coordinates are from the target ground-truth object. The final weight of prediction k is obtained by imposing a strict distance cutoff:

$$W_k = \begin{cases} W_k^{decay}, & \text{if } W_k^{decay} > \rho \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

to ensure that only predictions in close proximity to ground truth objects propagate gradients, striking a balance between incredibly sparse one-to-one matching and exhaustive gradient propagation, for stable training.

IV. EXPERIMENTS

A. Implementation Details

For fair comparison, MDHA is tested with two backbones: ResNet50 [25] pre-trained on ImageNet [26] and ResNet101 pre-trained on nuImages [18]. We set $D^{max} = 61.2$ and $D^{min} = 1.0$; a total of $N_q = 900$ queries are used in

TABLE I

PERFORMANCE COMPARISON ON THE NUSCENES VAL SET. * IS TRAINED WITH CBGS. † USES PRE-TRAINED WEIGHTS FROM FCOS3D [24]. ‡ BENEFITED FROM PRE-TRAINING ON NUIMAGES. § USES 900 ANCHORS INITIALIZED FROM KMEANS CLUSTERING OF THE NUSCENES TRAIN SET.

Method	Backbone	Image Size	Sparse Attention	Frames	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
PETR [8]*	ResNet50-DCN	384×1056	✗	1	0.313	0.381	0.768	0.278	0.564	0.923	0.225
Focal-PETR [10]	ResNet50-DCN	320×800	✗	1	0.320	0.381	0.788	0.278	0.595	0.893	0.228
SimMOD [15]	ResNet50-DCN	900×1600	✗	1	0.339	0.432	0.727	-	0.356	-	-
PETRv2 [9]	ResNet50	256×704	✗	2	0.349	0.456	0.700	0.275	0.580	0.437	0.187
StreamPETR [11]	ResNet50	256×704	✗	8	0.432	0.540	0.581	0.272	0.413	0.295	0.195
Sparse4Dv2 [13]§	ResNet50	256×704	✓	1	0.439	0.539	0.598	0.270	0.475	0.282	0.179
MDHA-fixed	ResNet50	256×704	✓	1	0.388	0.497	0.681	0.277	0.535	0.301	0.179
MDHA-conv	ResNet50	256×704	✓	1	0.396	0.498	0.681	0.276	0.517	0.338	0.183
DETR3D [7]*†	ResNet101-DCN	900×1600	✓	1	0.349	0.434	0.716	0.268	0.379	0.842	0.200
PETR [8] *†	ResNet101-DCN	512×1408	✗	1	0.366	0.441	0.717	0.267	0.412	0.834	0.190
Focal-PETR [10] †	ResNet101-DCN	512×1408	✗	1	0.390	0.461	0.678	0.263	0.395	0.804	0.202
SimMOD [15] †	ResNet101-DCN	900×1600	✗	1	0.366	0.455	0.698	0.264	0.340	0.784	0.197
PETRv2 [9] †	ResNet101	640×1600	✗	2	0.421	0.524	0.681	0.267	0.357	0.377	0.186
Sparse4D [12] †§	ResNet101-DCN	900×1600	✓	4	0.436	0.541	0.633	0.279	0.363	0.317	0.177
StreamPETR [11] ‡	ResNet101	512×1408	✗	8	0.504	0.592	0.569	0.262	0.315	0.257	0.199
Sparse4Dv2 [13] ‡§	ResNet101	512×1408	✓	1	0.505	0.594	0.548	0.268	0.348	0.239	0.184
MDHA-fixed‡	ResNet101	512×1408	✓	1	0.451	0.544	0.615	0.265	0.465	0.289	0.182
MDHA-conv‡	ResNet101	512×1408	✓	1	0.464	0.550	0.608	0.261	0.444	0.321	0.184

the decoder, with $k = 644$ queries from the encoder and $q = 256$ values propagated from the previous frame. The memory queue retains sparse features and anchors from the last $m = 4$ frames. We use a single-layer encoder to keep training times manageable while the decoder has $D = 6$ layers. Training loss weights are set to be $\lambda_1 = 2.0$, $\lambda_2 = 0.25$, $\lambda_3 = 0.01$ with auxiliary losses employing the same weights. For depth loss, we set $\varepsilon = \frac{10}{l}$ and $\rho = 0.01$. During training, denoising is applied for auxiliary supervision within the decoder, using 10 denoising groups per ground-truth. Following Sparse4Dv2 [13], the model is trained for 100 epochs for Table I and 25 epochs for Section IV-D, both using the AdamW optimizer [27] with 0.01 weight decay. Batch size of 16 is used with initial learning rate of 4e-4, decayed following the cosine annealing schedule [28]. Input augmentation follows PETR [8]. No CBGS [29] or test time augmentation was used in all experiments.

B. Dataset

We assess MDHA's performance on the large-scale autonomous driving nuScenes [18] dataset using its official performance metrics. It captures driving scenes with 6 surround-view cameras as 20-second video clips at 2 frames per second (FPS), with a total of 1000 scenes split up into 700/150/150 for training/validation/testing. The dataset is fully annotated with 3D bounding boxes for 10 object classes.

C. Main Results

Table I compares MDHA-conv and MDHA-fixed, which uses the Learnable and Fixed Depth approaches, respectively, against state-of-the-art camera-only query-based methods. With ResNet50, MDHA-conv outperforms most existing models, except for StreamPETR and Sparse4Dv2, which achieve slightly higher mAP and NDS. However, we would like to point out that StreamPETR benefits from dense attention and is trained with a sliding window, using 8 frames *and* a memory queue, for a single prediction. In contrast, our method employs efficient and scalable sparse attention with only a window of size 1 using the same memory queue. As a result, MDHA trains **1.9× faster** than StreamPETR with the same number of epochs. Additionally, Sparse4Dv2 initializes anchors from k-means clustering on the nuScenes train set, introducing bias towards the dataset, whereas our method utilizes adaptive proposals without anchor initialization. Thus, our method does not require prior knowledge of the dataset distribution. Furthermore, we observe that despite a lower input resolution, MDHA-conv achieves 5.7 percentage points (pp) higher mAP and 6.6 pp higher NDS compared to SimMOD, another proposal-based framework which uses complex 2D priors. With a ResNet101 backbone, MDHA-conv again outperforms most existing models except for StreamPETR and Sparse4Dv2, though it does reduce mASE by 0.1 pp and 0.7 pp respectively, which could be attributed to the multi-scale feature maps and our CDA mechanism enabling queries to better reason about large or multi-view-spanning vehicles. Even with lower input resolution, MDHA-conv once again outperforms SimMOD by 9.8 pp mAP and 9.5 pp NDS. For both backbones, MDHA-fixed slightly underperforms compared to MDHA-conv due to its fixed depth distribution, but offers an inference speed of 15.1 FPS on an RTX 4090, which is 0.7 higher than MDHA-conv.

D. Analysis

1) Effectiveness of hybrid anchors: To verify the effectiveness of our hybrid anchor scheme, we perform a quantitative and qualitative comparison with a learnable anchors baseline with no encoder, where proposals are implemented as parameterized embeddings with weights W^{emb}. Initial 3D anchor centers are obtained via $\sigma^{-1}(W^{emb})$ with zero-initialized queries. This setting mimics StreamPETR [11].

Figure 5 illustrates the comparison between proposals generated by these two approaches. In the learnable anchors

Fig. 5. **Qualitative comparison between 3D proposals obtained from the learnable anchors setting (top) and our MDHA Anchor Encoder (bottom).** We visualize *selected* proposals on all 6 cameras (left) and *all* proposals in bird's-eye-view (right). For visual clarity, we display non-overlapping proposals for learnable anchors, and the top-20 proposals based on classification score for MDHA.

setting, the same set of learned anchors are proposed regardless of the input scene, resulting in significant deviations from the actual objects. Thus, this approach relies heavily on the decoder to perform broad adjustments. It is also evident from the BEV that the proposals only cover a limited area around the vehicle, rendering them unsuitable for long-range detection. In contrast, our proposed method adapts to the input scene in two ways: first, the DepthNet predicts object depth based on image features; second, the Anchor Encoder selects only the top-k proposals most likely to contain an object based on the feature map. Visually, our proposals not only manage to discriminate the relevant objects within the scene, they also encompass these objects quite well, detecting even the partially obstructed barriers in the back-right camera without explicit occlusion handling. Furthermore, as our proposals are unbounded, the BEV shows many proposals far away from the ego vehicle, making it more effective for detecting objects further away. These alleviate the decoder's load, allowing it to focus on fine-tuned adjustments and thus enhancing overall efficiency and performance.

These observations are consistent with the quantitative comparison in Table II, where our hybrid scheme outperforms the learnable anchors baseline for *all* metrics. Notably, it achieves a 7.1 pp improvement in mAP and a 5.8 pp improvement in NDS. Due to the limited range of the learnable anchors, it suffers from a large translation error, which was reduced by 12.4 pp in the hybrid anchors scheme.

2) Ablation study on Circular Deformable Attention: Table III shows that without view-spanning, multiple projected reference points (multi-projection) outperforms a single projected reference point (single-projection, Section III-D) by 1.2 pp and 1.4 pp mAP and NDS. With view-spanning, although performance improves for both settings, multi-projection models obtain *worse* mAP and NDS than their single-projection counterparts. The performance jump between models 4 and 5 validates the efficacy of our multi-view-spanning mechanism, and we hypothesize that the one-to-many relationship between queries and reference points in multi-projection models slows down convergence compared to a single well-chosen point, causing it to lag behind in performance. Moreover, multi-projection models are slower by 0.5 FPS due to the additional feature sampling. Wrapping sampling points around (wraparound) also yields a small performance increase. Both view-spanning and wraparound have no impact on the FPS. Thus, our CDA adopts the single-projection approach for improved efficiency, and view-spanning with wraparound for maximal performance.

3) Number of sampling locations per reference point: Increasing the number of sampling locations per reference point in both the encoder (S_{enc}) and decoder (S_{dec}) enables queries to attend to more features. Table IV shows that as S_{enc} increases from 4 to 12, NDS improves by 0.4 pp, and as S_{dec} increases from 12 to 24, both mAP and NDS improve by 0.1 pp and 0.9 pp, respectively. Comparing the results of

TABLE II

PERFORMANCE COMPARISON WHEN TRAINED WITH PROPOSALS GENERATED VIA LEARNABLE VS HYBRID ANCHORS

Method	mAP↑	NDS↑	mATE↓	mASE↓	mAVE↓
Learnable Anchors	0.267	0.393	0.860	0.283	0.369
Hybrid Anchors (Ours)	**0.338**	**0.451**	**0.736**	**0.268**	**0.365**
Improvement (pp)	7.1	5.8	12.4	1.5	0.4

TABLE III

ABLATION STUDY ON CIRCULAR DEFORMABLE ATTENTION

ID	Single-Proj.	Multi-Proj.	View-Spanning	Wrap	mAP↑	NDS↑	FPS↑
1		✓			0.313	0.416	
2		✓	✓		0.321	0.435	13.8
3		✓	✓	✓	0.323	0.438	
4	✓				0.301	0.402	
5	✓		✓		0.331	0.445	**14.3**
CDA	✓		✓	✓	**0.338**	**0.451**	

TABLE IV

EFFECT OF NUMBER OF SAMPLING LOCATIONS PER REFERENCE POINT

Encoder	Decoder	mAP↑	NDS↑	mATE↓	mAVE↓	FPS↑
4	12	0.337	0.442	0.756	**0.347**	**14.4**
12	12	0.328	0.446	0.740	0.357	13.8
24	4	0.325	0.425	0.752	0.421	13.0
4	24	**0.338**	**0.451**	**0.736**	0.365	14.3

$(S_{enc}, S_{dec}) = (24, 4)$ with that of $(S_{enc}, S_{dec}) = (4, 24)$, the latter achieves 1.3 pp and 2.6 pp higher mAP and NDS, while being faster by 1.3 FPS. Thus, increasing S_{dec} yields a larger performance gain compared to increasing S_{enc} by the same amount. This could be attributed to the difference in the number of queries, which is much higher in the encoder than the decoder. Hence, even with a low S_{enc}, the encoder's queries provide adequate input coverage. This also explains why increasing S_{enc} results in more FPS reduction than increasing S_{dec}.

V. CONCLUSION

In this paper, we introduce MDHA, a novel framework which generates adaptive 3D output proposals using hybrid anchors. These proposals are sparsely refined and selected within our Anchor Encoder, followed by iterative refinement in the MDHA decoder. Sparse attention is performed by the multi-view-spanning CDA mechanism. MDHA significantly outperforms the learnable anchors baseline and achieves 46.4% mAP and 55.0% NDS on the nuScenes val set.

There are many possible improvements for MDHA. For instance, feature token sparsification [10], [30] could enhance encoder efficiency, and the use of full 3D anchors [12] as opposed to only 3D centers could improve performance. We hope that MDHA serves as a baseline for future advancements in query-based multi-camera 3D object detection.

ACKNOWLEDGMENT

The work of Junn Yong Loo is supported by the Ministry of Higher Education Malaysia under the Fundamental Grant Scheme (FRGS) G-M010-MOH-000206.

REFERENCES

[1] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *European Conference on Computer Vision*, 2020.

[2] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, "Bevdet: High-performance multi-camera 3d object detection in bird-eye-view," *arXiv preprint arXiv:2112.11790*, 2021.

[3] J. Huang and G. Huang, "Bevdet4d: Exploit temporal cues in multi-camera 3d object detection," *arXiv preprint arXiv:2203.17054*, 2022.

[4] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[5] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *European Conference on Computer Vision*, 2022.

[6] C. Yang, Y. Chen, H. Tian, C. Tao, X. Zhu, Z. Zhang, G. Huang, H. Li, Y. Qiao, L. Lu, J. Zhou, and J. Dai, "Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[7] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *Conference on Robot Learning*, 2022.

[8] Y. Liu, T. Wang, X. Zhang, and J. Sun, "Petr: Position embedding transformation for multi-view 3d object detection," in *European Conference on Computer Vision*, 2022.

[9] Y. Liu, J. Yan, F. Jia, S. Li, A. Gao, T. Wang, and X. Zhang, "Petrv2: A unified framework for 3d perception from multi-camera images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[10] S. Wang, X. Jiang, and Y. Li, "Focal-petr: Embracing foreground for efficient multi-camera 3d object detection," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1481–1489, 2024.

[11] S. Wang, Y. Liu, T. Wang, Y. Li, and X. Zhang, "Exploring object-centric temporal modeling for efficient multi-view 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[12] X. Lin, T. Lin, Z. Pei, L. Huang, and Z. Su, "Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion," *arXiv preprint arXiv:2211.10581*, 2022.

[13] ——, "Sparse4d v2: Recurrent temporal fusion with sparse model," *arXiv preprint arXiv:2305.14018*, 2023.

[14] X. Lin, Z. Pei, T. Lin, L. Huang, and Z. Su, "Sparse4d v3: Advancing end-to-end 3d detection and tracking," *arXiv preprint arXiv:2311.11722*, 2023.

[15] Y. Zhang, W. Zheng, Z. Zhu, G. Huang, J. Lu, and J. Zhou, "A simple baseline for multi-camera 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[16] Z. Wang, Z. Huang, J. Fu, N. Wang, and S. Liu, "Object as query: Lifting any 2d object detector to 3d detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[17] X. Jiang, S. Li, Y. Liu, S. Wang, F. Jia, T. Wang, L. Han, and X. Zhang, "Far3d: Expanding the horizon for surround-view 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[18] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[19] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," in *International Conference on Learning Representations*, 2023.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[21] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations*, 2021.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[23] 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*, 2020.

[24] T. Wang, X. Zhu, J. Pang, and D. Lin, "Fcos3d: Fully convolutional one-stage monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.

[28] ——, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017.

[29] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.

[30] B. Roh, J. Shin, W. Shin, and S. Kim, "Sparse detr: Efficient end-to-end object detection with learnable sparsity," in *International Conference on Learning Representations*, 2022.