Exploring Object-Centric Temporal Modeling for Efficient Multi-View 3D Object Detection

Shihao Wang[1†] Yingfei Liu[2] Tiancai Wang[2] Ying Li[1] Xiangyu Zhang[2]
[1]Beijing Institute of Technology [2]MEGVII Technology

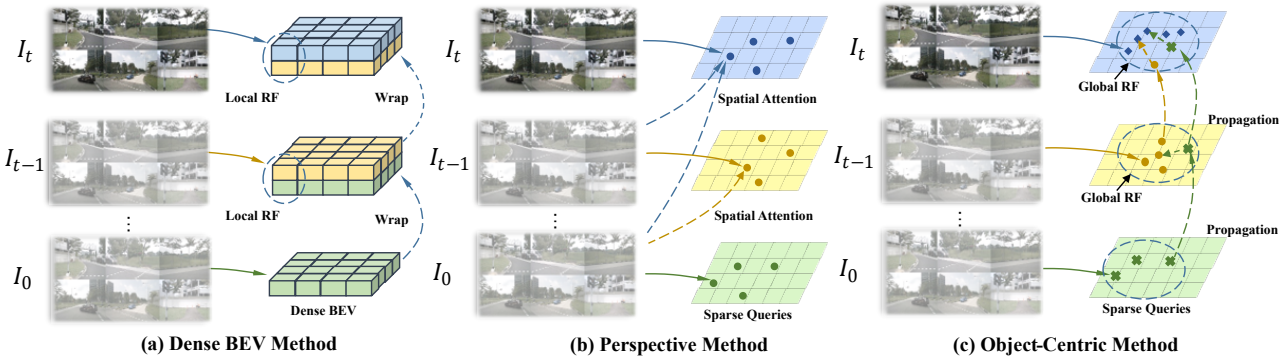

Figure 1. Different temporal fusion methods from bird-eye-view (BEV) space, perspective view, and our proposed object-centric. RF indicates receptive field. The solid lines and dotted lines represent spatial and temporal operations respectively.

Abstract

In this paper, we propose a long-sequence modeling framework, named StreamPETR, for multi-view 3D object detection. Built upon the sparse query design in the PETR series, we systematically develop an object-centric temporal mechanism. The model is performed in an online manner and the long-term historical information is propagated through object queries frame by frame. Besides, we introduce a motion-aware layer normalization to model the movement of the objects. StreamPETR achieves significant performance improvements only with negligible computation cost, compared to the single-frame baseline. On the standard nuScenes benchmark, it is the first online multi-view method that achieves comparable performance (67.6% NDS & 65.3% AMOTA) with lidar-based methods. The lightweight version realizes 45.0% mAP and 31.7 FPS, outperforming the state-of-the-art method (SOLOFusion) by 2.3% mAP and 1.8× faster FPS. Code has been available at https://github.com/exiawsh/StreamPETR.git.

1. Introduction

Camera-only 3D detection is crucial for autonomous driving because of the low deployment cost and ease of detecting road elements. Recently, multi-view object detection has made remarkable progress by leveraging temporal information [27, 16, 31, 25, 39, 29]. The historical features facilitate the detection of occlusion objects and greatly improve the performance. According to the differences between temporal representations, existing methods can be roughly divided into *BEV temporal* and *perspective temporal* methods.

BEV temporal methods [27, 16, 25, 39] explicitly warp BEV features from historical to current frame, as illustrated in Fig. 1 (a), where BEV features serve as an efficient intermediate representation for temporal modeling. However, the highly structured BEV features limit the modeling of moving objects. This paradigm requires a large receptive field to alleviate this problem [16, 39, 27].

Different from these approaches, perspective temporal methods [31, 29] are mainly based on DETR [4, 60]. The sparse query design facilitates the modeling of moving objects [29]. However, the sparse object queries need to interact with multi-frame image features for long-term temporal dependence (see Fig. 1 (b)), leading to multiple computations. Thus, existing works are either stuck in solving the moving objects or introducing multiple computation costs.

Based on the above analysis, we suppose it is possible to employ sparse queries as the hidden states of temporal propagation. In this way, we can utilize object queries to model moving objects while keeping high efficiency. Therefore, we introduce a new paradigm: *object-centric temporal* modeling and design an efficient framework, termed StreamPETR, as shown in Fig. 1 (c). StreamPETR directly performs frame-by-frame 3D predictions on stream-

† Work done during the internship at MEGVII Technology.

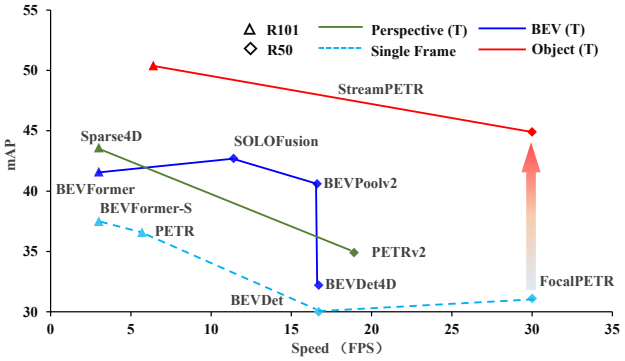

Figure 2. The speed-accuracy trade-off of different models on nuScenes val set. The inference speed is calculated on RTX3090 GPU in online streaming video. T indicates the model with temporal modeling.

ing video. It is effective for motion modeling and is able to build long-term spatial-temporal interaction.

Specifically, a memory queue is first built to store the historical object queries. Then a propagation transformer conducts long-range temporal and spatial interaction with current object queries. The updated object queries are used to generate 3D bounding boxes and pushed into the memory queue. Besides, a motion-aware layer normalization (MLN) is introduced to implicitly encode the motion of the ego vehicle and surrounding objects at different time stamps.

Compared with existing temporal methods, the proposed object-centric temporal modeling brings several advantages. StreamPETR only processes a small number of object queries instead of dense feature maps at each time stamp, consuming negligible computational burden (as shown in Fig. 2). For moving objects, MLN alleviates the cumulative error in video streaming. Except for the location prior used in previous methods, StreamPETR additionally considers the semantic similarity by global attention, which facilitates the detection in motion scenes. To summarize, our contributions are:

- We pull out the key of streaming multi-view 3D detection and systematically design an *object-centric* temporal modeling paradigm. The long-term historical information is propagated through object queries frame by frame.

- We develop an object-centric temporal modeling framework, termed StreamPETR. It models moving objects and long-term spatial-temporal interaction simultaneously, consuming negligible storage and computation costs.

- On the standard nuScenes dataset, StreamPETR outperforms all online camera-only algorithms. Extensive experiment shows that it can be well generalized to other sparse query-based methods, e.g. DETR3D [47].

2. Related Work

2.1. Multi-view 3D Object Detection

Multi-view 3D detection is an important task in autonomous driving, which needs to continuously process multi-camera images and predict 3D bounding boxes over time. Pioneer's works [47, 30, 18, 27, 19, 48] focus on the efficient transformation from multiple perspective views to a unified 3D space in a single frame. The transformation can be divided into BEV-based methods [18, 27, 49, 15, 25, 19] and sparse query based methods [47, 30, 29, 5, 48]. To alleviate the occlusion problem and ease the difficulty of speed prediction, recent works additionally introduce temporal information to extend these two paradigms.

It is relatively intuitive to extend the single-frame BEV methods for temporal modeling. BEVFormer [27] first introduces sequential temporal modeling into multi-view 3D object detection and applies temporal self-attention. BEVDet series [16, 25, 23] use concatenate operation to fuse the adjacent BEV features and achieve remarkable results. Furthermore, SOLOFusion [39] extends BEVStereo [23] to long-term memory and reaches a promising performance. Without an intermediate feature representation, the temporal modeling of query-based methods is more challenging. PETRv2 [31] performs the global cross-attention, while DETR4D [34] and Sparse4D [29] apply sparse attention to model the interaction between multi-frames, which introduce multiple computations. However, the sparse query design is convenient to model the moving objects [29]. In order to combine the advantages of the two paradigms, we utilize sparse object queries as the intermediate representation, which can model moving objects and efficiently propagate long-term temporal information.

2.2. Query Propagation

Since DETR [4] is proposed in 2D object detection, the object query has been applied in many downstream tasks [54, 35, 57, 56, 12] to model the temporal interaction. For video object detection, LWDN [20] adopts a brain-inspired memory mechanism to propagate and update the memory feature. QueryProp [12] performs query interaction to reduce the computational cost on non-key frames. It achieves significant improvements and maintains high efficiency. 3D-MAN [52] has a similar idea and extends a single-frame Lidar detector to multi-frames, which effectively combines the features coming from different perspectives of a scene. In object tracking, MOTR [54] and TrackFormer [35] propose the track query to model the object association across frames. MeMOT [2] employs a memory bank to build long temporal dependence, which further boosts performance. MOTRv2 [57] eases the conflict between the detection and association tasks by incorporating an extra detector. MUTR [56] and PF-Track [36] extend

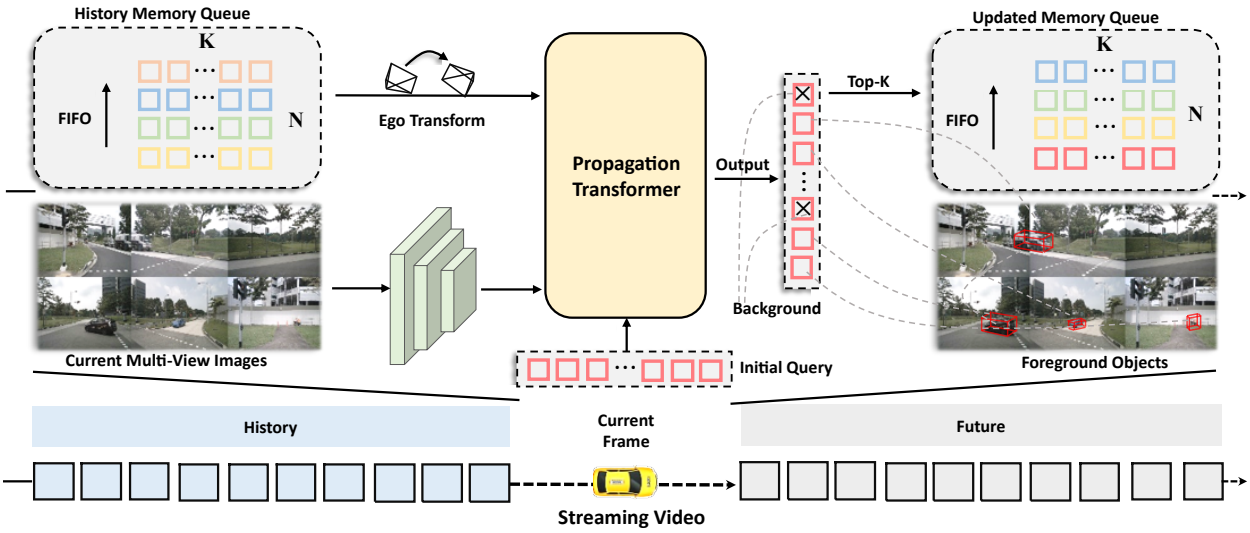

Figure 3. Overall architecture of the proposed StreamPETR. The memory queue stores the historical object queries. In the propagation transformer, recent object queries successively interact with historical queries and current image features to obtain temporal and spatial information. The output queries are further used to generate detection results and the top-K foreground queries are pushed into the memory queue. Through the recurrent update of the memory queue, the long-term temporal information is propagated frame by frame.

MOTR [54] into multi-view 3D object tracking and achieve a promising result.

3. Delving into Temporal Modeling

To facilitate our study, we present a generalized formulation for various temporal modeling designs. Given the perspective view features $F_{2d} = \{F_{2d}^0 \cdots F_{2d}^t\}$, dense BEV features $F_{bev} = \{F_{bev}^0 \cdots F_{bev}^t\}$ and sparse object features $F_{obj} = \{F_{obj}^0 \cdots F_{obj}^t\}$. The dominant temporal modeling methods can be formulated as:

$$\tilde{F}_{out} = \varphi(F_{2d}, F_{bev}, F_{obj}) \tag{1}$$

where φ is the temporal fusion operation, $\tilde{F}_{out}$ is the output feature that includes temporal information. We first describe the existing temporal modeling from BEV and perspective view. After that, the proposed object-centric temporal modeling is elaborated.

BEV Temporal Modeling uses the grid-structured BEV features to perform the temporal fusion. To compensate for the ego vehicle motion, the last frame feature F_{bev}^{t-1} is usually aligned to the current frame.

$$\tilde{F}_{bev}^t = \varphi(F_{bev}^{t-1}, F_{bev}^t) \tag{2}$$

Then a temporal fusion function φ (concatenation [16, 25] or deformable attention [27]) can be applied for intermediate temporal representation $\tilde{F}_{bev}^t$. Extending the above process to long temporal modeling, there are two main routes. The first one is to align the historical k BEV features and concatenate them with the current frame.

$$\tilde{F}_{bev}^t = \varphi(F_{bev}^{t-k}, \cdots, F_{bev}^{t-1}, F_{bev}^t) \tag{3}$$

For another one, the long-term historical information is propagated through the hidden states of BEV features $\tilde{F}_{bev}^{t-1}$ in a recurrent manner.

$$\tilde{F}_{bev}^t = \varphi(\tilde{F}_{bev}^{t-1}, F_{bev}^t) \tag{4}$$

However, the BEV temporal fusion only considers the static BEV features and ignores the movement of the objects, leading to spatial dislocation.

Perspective Temporal Modeling is mainly performed via interactions between object queries and perspective features. The temporal function φ is usually achieved by the spatial cross-attention [31, 29, 34]:

$$\tilde{F}_{obj}^t = \varphi(F_{2d}^{t-k}, F_{obj}^t) \cdots + \varphi(F_{2d}^t, F_{obj}^t) \tag{5}$$

The cross-attention between object query and multi-frame perspective view requires repeated feature aggregation. Simply extending to long-term temporal modeling greatly increases the computation cost.

Object-centric Temporal Modeling is our proposed object-centric solution, which models the temporal interaction by object queries. Through object queries, the motion compensation can be conveniently applied based on estimated states F_{obj}^{t-1}.

$$\tilde{F}_{obj}^{t-1} = \mu(F_{obj}^{t-1}, M) \tag{6}$$

where μ is an explicit linear velocity model or implicit function to encode motion attributes M (including the relative time interval $\triangle t$, estimated velocity v, and ego-pose matrix E, which are the same definition in Sec. 4). Further, a global attention φ is constructed to propagate temporal information through object queries frame by frame:

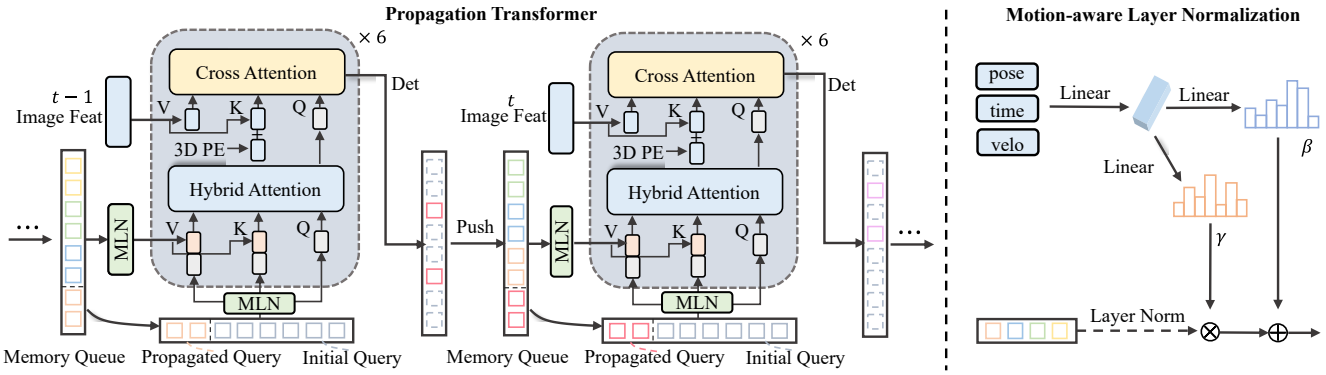

Figure 4. The details of the propagation transformer and motion-aware layer normalization. In the propagation Transformer [43], object queries interact with hybrid queries and image features iteratively. The motion-aware layer normalization encodes the motion attributes (ego pose, timestamps, velocity) and performs a compensation implicitly. Rectangles of varying hues symbolize queries from distinct frames, gray rectangles represent initialized queries of current frame, dashed rectangles correspond to background queries.

$$\tilde{F}^t_{obj} = \varphi(\tilde{F}^{t-1}_{obj}, F^t_{obj}) \tag{7}$$

4. Method

4.1. Overall Architecture

As illustrated in Fig. 3, StreamPETR is built upon end-to-end sparse query-based 3D object detectors [30, 47]. It consists of an image encoder, a recursively updated memory queue, and a propagation transformer [43]. The image encoder is a standard 2D backbone, which is applied to extract semantic features from multi-view images. Then the extracted features, information in the memory queue, and object queries are fed into the propagation transformer to perform the spatial-temporal interaction. The main difference between StreamPETR and single-frame baseline is the memory queue, which recursively updates the temporal information of object queries. Combined with the propagation transformer, the memory queue can propagate temporal priors from previous to current frames efficiently.

4.2. Memory Queue

We design a memory queue of $N \times K$ for effective temporal modeling. N is the number of stored frames and K is the number of objects stored per frame. According to the experience, we set $N = 4$ and $K = 256$ (ensuring high recall in complex scenarios). After the preset time interval τ, the relative time interval $\triangle t$, context embedding Q_c, object center Q_p, velocity v, and ego-pose matrix E of selected object queries are stored in memory queue. Specifically, the above information, corresponding to foreground objects (with top-K highest classification score), is selected and pushed into the memory queue. The entrance and exit of the memory queue follow the first-in, first-out (FIFO) rule. When information from a new frame is added to the memory queue, the oldest is discarded. Actually, the proposed memory queue is highly flexible and customized, users can freely control the maximal memory size $N \times K$ and saving interval τ during both training and inference.

4.3. Propagation Transformer

As illustrated in Fig. 4, the propagation transformer consists of three main components: (1) the motion-aware layer normalization module implicitly updates the object state according to the context embedding and motion information recorded in the memory queue; (2) the hybrid attention replaces the default self-attention operation. It plays the role of temporal modeling and removing duplicated predictions; (3) the cross-attention is adopted for feature aggregation. It can be replaced with an arbitrary spatial operation to build the relationship between image tokens and 3D object queries, such as global attention in PETR [30] or sparse projective attention in DETR3D [47].

Motion-aware Layer Normalization is designed to model the movement of objects. For simplicity, we take the transformation process from the last frame $t-1$ as the example and adopt the same operation for other previous frames. Given the ego pose matrix from the last frame E_{t-1} and current frame E_t, the ego transformation E^t_{t-1} can be calculated as:

$$E^t_{t-1} = E^{inv}_t \cdot E_{t-1} \tag{8}$$

Assume that objects are static, 3D centers Q^{t-1}_p in memory queue can be explicitly aligned to the current frame, which is formulated as:

$$\tilde{Q}^t_p = E^t_{t-1} \cdot Q^{t-1}_p \tag{9}$$

where $\tilde{Q}^t_p$ is the aligned centers. Motivated by the task-specific control in generative model [9, 40, 55], we adopt a conditional layer normalization to model the movement of the objects. As shown in Fig. 4, the default affine transformation in layer normalization (LN) is closed. The motion attributes $(E^t_{t-1}, v, \triangle t)$ are flattened and converted to affine

Table 1. Comparison on the nuScenes `val` set. *Benefited from the perspective-view pre-training. ‡ 300 randomly initialized queries and 128 propagation queries. † Offline method using future frames. FPS is measured on RTX3090 with fp32.

Methods	Backbone	Image Size	Frames	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	FPS↑
BEVDet [18]	ResNet50	256 × 704	1	0.298	0.379	0.725	0.279	0.589	0.860	0.245	16.7
BEVDet4D [16]	ResNet50	256 × 704	2	0.322	0.457	0.703	0.278	0.495	0.354	0.206	16.7
PETRv2 [31]	ResNet50	256 × 704	2	0.349	0.456	0.700	0.275	0.580	0.437	0.187	18.9
BEVDepth [25]	ResNet50	256 × 704	2	0.351	0.475	0.639	0.267	0.479	0.428	0.198	15.7
BEVStereo [23]	ResNet50	256 × 704	2	0.372	0.500	0.598	0.270	0.438	0.367	0.190	12.2
BEVFormerv2 [50] † *	ResNet50	-	-	0.423	0.529	0.618	0.273	0.413	0.333	0.188	-
SOLOFusion [39]	ResNet50	256 × 704	16+1	0.427	0.534	0.567	0.274	0.511	0.252	0.181	11.4
BEVPoolv2 [17]	ResNet50	256 × 704	8+1	0.406	0.526	0.572	0.275	0.463	0.275	0.188	16.6
StreamPETR	ResNet50	256 × 704	8	0.432	0.540	0.581	0.272	0.413	0.295	0.195	27.1
StreamPETR* ‡	ResNet50	256 × 704	8	0.450	0.550	0.613	0.267	0.413	0.265	0.196	**31.7**
DETR3D [47]*	ResNet101-DCN	900 × 1600	1	0.349	0.434	0.716	0.268	0.379	0.842	0.200	3.7
Focal-PETR [44]	ResNet101-DCN	512 × 1408	1	0.390	0.461	0.678	0.263	0.395	0.804	0.202	6.6
PETR [30]*	ResNet101-DCN	512 × 1408	1	0.366	0.441	0.717	0.267	0.412	0.834	0.190	5.7
BEVFormer [27]*	ResNet101-DCN	900 × 1600	4	0.416	0.517	0.673	0.274	0.372	0.394	0.198	3.0
PolarDETR [5]-T*	ResNet101-DCN	900 × 1600	2	0.383	0.488	0.707	0.269	0.344	0.518	0.196	3.5
Sparse4D [29]*	ResNet101-DCN	900 × 1600	4	0.436	0.541	0.633	0.279	0.363	0.317	**0.177**	4.3
BEVDepth	ResNet101	512 × 1408	2	0.412	0.535	0.565	0.266	0.358	0.331	0.190	-
SOLOFusion	ResNet101	512 × 1408	16+1	0.483	0.582	**0.503**	0.264	0.381	**0.246**	0.207	-
StreamPETR*	ResNet101	512 × 1408	8	**0.504**	**0.592**	0.569	**0.262**	**0.315**	0.257	0.199	6.4

vectors γ and β by two linear layers (ξ_1, ξ_2):

$$\begin{aligned} \gamma &= \xi_1(E_{t-1}^t, v, \triangle t), \\ \beta &= \xi_2(E_{t-1}^t, v, \triangle t) \end{aligned} \tag{10}$$

Afterward, the affine transformation is performed to get the motion-aware context embedding $\tilde{Q}_c^t$ and motion-aware position encoding $\tilde{Q}_{pe}^t$.

$$\begin{aligned} \tilde{Q}_{pe}^t &= \gamma \cdot LN(\psi(\tilde{Q}_p^t)) + \beta, \\ \tilde{Q}_c^t &= \gamma \cdot LN(Q_c^t) + \beta \end{aligned} \tag{11}$$

where ψ is a multi-layer perceptron (MLP) that converted the 3D sampled points $\tilde{Q}_p^t$ into position encoding $\tilde{Q}_{pe}^t$. For the sake of unification, the MLN is also adopted into current object queries. The velocity v and time interval $\triangle t$ of the current frame are zero-initialized.

Hybrid Attention layer. The self-attention in DETR [4] contributes to duplicated prediction removal. We replace it with hybrid attention, which additionally introduces temporal interaction. As shown in Fig. 4, all stored object queries in the memory queue are concatenated with current queries to obtain the hybrid queries. The hybrid queries are regard as the key and $value$ in multi-head attention. Since the number of hybrid queries is small (about $2k$, which is far less than image tokens in the cross-attention), the hybrid attention layer brings negligible computation cost.

Following PETR [30], the $query$ can be defined as a randomly initialized 3D anchor. To fully utilize the spatial and context priors in streaming video, some object queries in the memory queue are directly propagated into the current frame. In our implementation, queries from the last frame are concatenated with randomly initialized queries. For a fair comparison, the number of randomly initialized queries and propagated queries are set to 644 and 256 respectively.

5. Experiments

5.1. Dataset and Metrics

We evaluate our approach on the large-scale NuScenes dataset [1] and Waymo Open dataset [42].

The nuScenes Dataset includes 1000 scenes, which are 20 seconds in length and annotated at 2Hz. The camera rig covers the full 360° field of view (FOV). The annotations contain up to 1.4M 3D bounding boxes, and 10 common classes are used for evaluation: car, truck, bus, trailer, construction vehicle, pedestrian, motorcycle, bicycle, barrier, and traffic cone. We compare the methods with the following metrics, the nuScenes Detection Score (NDS), mean Average Precision (mAP), and 5 kinds of True Positive (TP) metrics including average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE), average attribute error (AAE). Following the standard evaluation metrics, we report the average multi-object tracking accuracy (AMOTA), average multi-object tracking precision (AMOTP), recall (RECALL), multi-object tracking accuracy (MOTA) and ID switch (IDS) for 3D object tracking task.

Waymo Open Dataset collects camera data only spanning a horizontal FOV of 230 degrees. The ground truth bounding boxes are annotated to a maximum range of 75 meters. The longitudinal error tolerant metrics LET-3D-AP, LET-3D-AP-H and LET-3D-APL are used for evaluation. Noting that we only use 20% of training data for fair comparison according to common practice.

Table 2. Comparison on the nuScenes `test` set. TTA is test time augmentaion.

Methods	Modality	Backbone	Image / Voxel	TTA	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
CenterPoint [53]	L	-	0.075 × 0.075 × 0.2	✘	0.603	0.673	**0.262**	**0.239**	0.361	0.288	0.136
FCOS3D [46]	C	R101-DCN	900 × 1600	✔	0.358	0.428	0.690	0.249	0.452	1.434	0.124
DETR3D [47]	C	V2-99	900 × 1600	✔	0.412	0.479	0.641	0.255	0.394	0.845	0.133
MV2D [48]	C	V2-99	640 × 1600	✘	0.463	0.514	0.542	0.247	0.403	0.857	0.127
UVTR [24]	C	V2-99	900 × 1600	✘	0.472	0.551	0.577	0.253	0.391	0.508	0.123
BEVFormer [27]	C	V2-99	900 × 1600	✘	0.481	0.569	0.582	0.256	0.375	0.378	0.126
PETRv2 [31]	C	V2-99	640 × 1600	✘	0.490	0.582	0.561	0.243	0.361	0.343	0.120
PolarFormer [19]	C	V2-99	900 × 1600	✘	0.493	0.572	0.556	0.256	0.364	0.439	0.127
BEVStereo [23]	C	V2-99	640 × 1600	✘	0.525	0.610	0.431	0.246	0.358	0.357	0.138
StreamPETR	C	V2-99	640 × 1600	✘	0.550	0.636	0.479	**0.239**	0.317	0.241	**0.119**
BEVDet4D [16]	C	Swin-B [32]	900 × 1600	✔	0.451	0.569	0.511	0.241	0.386	0.301	0.121
BEVDepth [25]	C	ConvNeXt-B	640 × 1600	✘	0.520	0.609	0.445	0.243	0.352	0.347	0.127
AeDet [10]	C	ConvNeXt-B	640 × 1600	✔	0.531	0.620	0.439	0.247	0.344	0.292	0.130
PETRv2	C	RevCol-L [3]	640 × 1600	✘	0.512	0.592	0.547	0.242	0.360	0.367	0.126
SOLOFusion [39]	C	ConvNeXt-B	640 × 1600	✘	0.540	0.619	0.453	0.257	0.376	0.276	0.148
StreamPETR	C	ViT-L	800 × 1600	✘	**0.620**	**0.676**	0.470	0.241	**0.258**	**0.236**	0.134

Table 3. Comparison of 3D object tracking on nuScenes test set.

Methods	AMOTA↑	AMOTP↓	RECALL↑	IDS↓
CenterPoint [53]	0.638	0.555	67.5%	760
SimpleTrack [37]	0.668	0.550	70.3%	575
QD3DT [14]	0.217	1.550	37.5%	6856
MUTR3D [56]	0.270	1.494	41.1%	6018
CC-3DT [11]	0.410	1.274	57.8%	3334
PolarDETR [5]	0.273	1.185	40.4%	2170
UVTR [24]	0.519	1.125	59.9%	2204
QTrack [51]	0.480	1.100	59.7%	1484
Sparse4D [29]	0.519	1.078	63.3%	1090
ByteTrackv2 [58]	0.564	1.005	63.5%	704
PF-Track [36]	0.434	1.252	53.8%	**249**
StreamPETR	**0.653**	**0.876**	**73.3%**	1037

Table 4. Comparison on the Waymo `val` set. $*$ The saving interval τ is set to 5 during testing. ‡ The saving interval τ is set to 1.

Methods	Backbone	mAPL↑	mAP↑	mAPH↑
BEVFormer++ [26]	ResNet101-DCN	0.361	0.522	0.481
MV-FCOS3D++ [45]	ResNet101-DCN	0.379	0.522	0.484
PETR-DN [30]	ResNet101	0.358	0.502	0.462
PETRv2 [31]	ResNet101	0.366	0.519	0.479
StreamPETR*	ResNet101	**0.399**	**0.553**	0.517
StreamPETR‡	ResNet101	0.395	0.551	**0.518**

5.2. Implementation Details

We conduct experiments with ResNet50 [13], ResNet101, V2-99 [21] and ViT [8] backbones under different pre-training. Following previous methods [27, 30, 39], the performance of ResNet50 and ResNet101 models with pre-trained weights ImageNet [7] and nuImages [1] are provided on the nuScenes val set. To scale up our method, we also report results on the nuScenes test set with V2-99 initialized from DD3D [38] checkpoint and ViT-Large [8]. Following BEVFormerv2 [50], the ViT-Large [8] is pre-trained on Objects365 [41] and COCO [28] dataset.

StreamPETR is trained by AdamW [33] optimizer with a batch size of 16. The base learning rate is set to 4e-4 and the cosine annealing policy is employed. Only key frames are used during both training and inference. All experiments are conducted without CBGS [59] strategy. Our implementation is mainly based on Focal-PETR [44], which introduces auxiliary 2D supervision. The models in the ablation study are trained for 24 epochs, while trained for 60 epochs when compared with others. In particular, we only train 24 epochs for ViT-L [8] to prevent over-fitting. For image and BEV data augmentation, we adopt the same methods as PETR [18, 30]. We randomly skip 1 frame during the training sequence for temporal data augmentation [27].

5.3. Main Results

NuScenes Dataset. We compare the proposed StreamPETR with previous state-of-the-art vision-based 3D detectors on the nuScenes val and test set. As shown in Tab. 1, StreamPETR shows superior performance on mAP, NDS, mASE, and mAOE metrics when adopting ResNet101 backbone with nuImages pretraining. Compared with the single frame baseline Focal-PETR, StreamPETR has considerable improvements of 11.4% mAP and 13.1% NDS. The mATE of StreamPETR is 10.9% better than Focal-PETR, indicating that our object-centric temporal modeling is able to improve both the accuracy of localization. With image resolutions of 256×704 and adopting ResNet50 backbone, StreamPETR exceeds the state-of-the-art method (SOLOFusion) by 0.5 % mAP and 0.6 % NDS. When we reduce the number of queries and apply nuImages pretraining, our method has 2.3 % and 1.6 % advantages in mAP and NDS. At the same time, the inference speed of StreamPETR is 1.8× faster.

When we compare the performance on the test set in Tab. 2 and adopt a smaller V2-99 backbone, StreamPETR can surpass SOLOFusion with ConvNext-Base backbone by 1.0% mAP and 1.7% NDS. Scaling up the backbone to ViT-Large [8], StreamPETR achieves 62.0% of mAP,

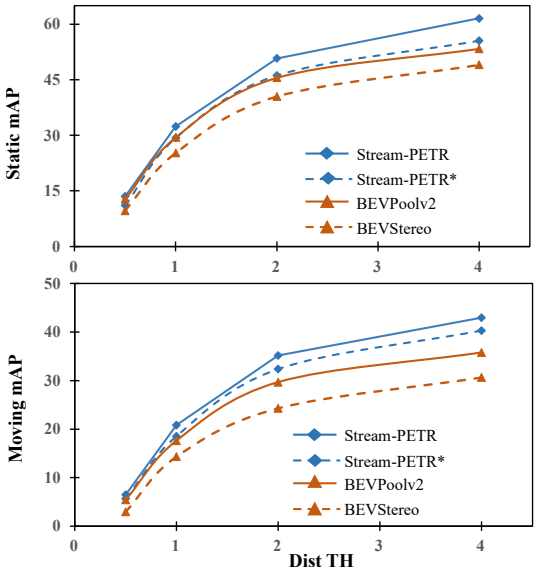

Figure 5. The mAP results with different distance thresholds (Dist TH) on the nuScenes val set. $*$ indicates StreamPETR without the proposed motion-aware layer normalization. Top: Boxes with a velocity lower than 1m/s are maintained for analysis. Down: Boxes with a velocity higher than 1m/s are maintained for analysis.

67.6% of NDS, and 25.8% of mAOE. Note that it is the first online multi-view method that achieves comparable performance with CenterPoint.

For 3D multi-object tracking task, we simply extend the multi-object tracking of CenterPoint [53] to the multi-view 3D setting. Owing to the exceptional detection and velocity estimation performance, StreamPETR significantly outperforms ByteTrackv2 [58] with an impressive margin of +8.9% AMOTA in Tab. 3. Furthermore, StreamPETR excels over CenterPoint [53] in AMOTA, and demonstrates superior benefits in RECALL.

Waymo Open Dataset. In this section, we provide experimental results on the Waymo val set, as shown in Tab. 4. Our model has trained 24 epochs and the saving interval of the memory queue is set to 5. It can be seen that our method shows superiority in official metrics compared with the dense BEV methods, *e.g.* BEVFormer++ [26] and MV-FCOS3D++ [45]. The Waymo open dataset has a larger evaluation range than nuScenes, our object-centric modeling method still shows obvious advantages in localization capability and longitudinal prediction. We also re-implemented PETR-DN and PETRv2 (all with query denoising [22]) as baseline models. StreamPETR outperforms the single-frame PETR-DN with a margin of 4.1% mAPL, 5.1% mAP, and 5.5% mAP-H. The Waymo open dataset covers part of the horizontal FOV, while object-centric temporal modeling still brings significant improvement. When we adopt the checkpoint and adjust saving interval τ to 1 during testing, StreamPETR has slight performance degradation, proving the adaptability on sensor frequency.

Table 5. Training frames for long-term fusion. W indicates testing in the sliding window, and V indicates testing in online video.

Training frames	Test	mAP↑	NDS↑	mATE↓	mAVE↓
1	-	0.317	0.372	0.770	0.885
2	W	0.328	0.410	0.742	0.726
2	V	0.315	0.401	0.738	0.767
4	W	0.377	0.483	0.683	0.385
4	V	0.366	0.475	0.685	0.392
8	W	0.396	0.501	0.664	0.324
8	V	0.402	0.505	0.660	**0.316**
12	W	**0.403**	0.507	0.649	0.325
12	V	0.402	**0.509**	**0.645**	**0.316**

Table 6. Ablation of motion-aware layer normalization. MC is explicit motion compensation. LN is layer normalization.

MC	LN	Ego Pose	Time	Velocity	mAP↑	NDS↑	mATE↓	mAVE↓
					0.378	0.483	0.697	0.354
✔					0.380	0.481	0.693	0.379
	✔				0.375	0.481	0.702	0.370
	✔	✔			0.398	0.501	0.667	**0.316**
	✔		✔		0.381	0.488	0.697	0.354
	✔			✔	0.386	0.489	0.690	0.373
	✔	✔	✔	✔	**0.402**	**0.505**	**0.660**	**0.316**

5.4. Ablation Study & Analysis

Impact of Training Sequence Length. StreamPETR is trained in local sliding windows and tested in online streaming video. To analyze the inconsistency between training and testing, we conduct experiments with varying numbers of training frames and show results in Tab. 5. When adding more training frames, the performance of StreamPETR continues to grow, and the performance gap between sliding windows and online video decreases obviously. It is worth noting that when the number of training frames increases to 8, video testing (40.2% mAP, 50.5% NDS) shows superior performance than the sliding window (39.6% mAP, 50.1% NDS), which proves that our method has a good potential to build long-term temporal dependency. Expanding to 12 frames brings limited performance improvement, so we train our models on 8 frames for experimental efficiency.

Effect of Motion-aware Layer Normalization. We compare the different designs for decoupling the ego vehicle and moving objects in Tab. 6. It can be seen that the performance does not improve when adopting explicit motion compensation (MC). We argue that the explicit way may cause error propagation in the early training phase. The MLN implicitly encodes and decouples the movements of the ego vehicle and moving objects. Specifically, implicit encoding of ego poses has achieved significant improvements, among which mAP increases by 2.0% and NDS increases by 1.8%. Besides, the encoding of relative time offset $\triangle t$ and object velocity v can further boost the performance. Both mAP and NDS are increased by 0.4%, which indicates that dynamic properties have a beneficial effect on the temporal interaction between object queries.

Number of Frames for Long-term Fusion. In Tab. 7,

Figure 6. Visualization results of StreamPETR. On the BEV plane (right), the groud-truth and predictions are drawn in green and blue rectangles respectively. The failure cases are marked by red circles.

Table 7. Number of frames (N) for long-term fusion.

number frames	mAP↑	NDS↑	mATE↓	mAVE↓	FPS↑
0	0.317	0.372	0.770	0.885	**27.7**
1	0.394	0.501	0.669	0.324	**27.7**
2	0.401	**0.505**	**0.660**	**0.314**	27.4
3	0.400	0.504	0.663	0.322	27.3
4	**0.402**	**0.505**	**0.660**	0.316	27.1

we analyze the impacts of memory size on hybrid attention. We can find that the mAP and NDS are improved with the increase of the memory size and begin to saturate when reaching 2 frames (nearly 1 second). The object query in StreamPETR is propagated and updated recursively, so even without a large-capacity memory queue, our method can still build a long-term spatial-temporal dependency. Since increasing the memory queue brings negligible computing costs, we use 4 frames to alleviate forgetting and obtain more stable results.

Perspective v.s. Object-Centric. StreamPETR achieves efficient temporal modeling through the interaction of sparse object queries. An alternative solution is to build temporal interaction via the perspective memory [31]. As shown in Tab. 8, the query-based temporal modeling has superior performance than perspective-based both on speed and accuracy. The combination of the query and perspective memory does not further improve the performance, implying that the temporal propagation of global query interaction is sufficient to achieve leading performance. Besides, concatenating current object queries with the queries of the last frame improves 0.7% mAP and 0.9% NDS.

Analysis of Moving Objects. In this section, we detailed analyze the performance of StreamPETR on perceiving static and moving objects respectively. For fair comparisons, all models are trained with 24 epochs without CBGS [59] and evaluated on the nuScenes [1] val set. The detection performance of moving objects still lags behind that of static objects to a large margin even with temporal modeling. Compared with dense BEV paradigms [23, 17], StreamPETR∗ has reached promising performance on both static and moving objects. This proves the superiority of object-centric temporal modeling, which has global temporal and spatial receptive fields. Applying the implicit encoding for motion information, the performance of StreamPETR can be further improved.

Table 8. From of the temporal propagation. 'Perspective and Object' mean propagating temporal information via image features and object queries respectively. 'Propagated' indicates concatenating the propagated queries from last frame.

Perspective	Object	Propagated	mAP↑	NDS↑	mATE↓	mAVE↓	FPS↑
			0.317	0.372	0.770	0.885	**27.7**
✔		✔	0.361	0.459	0.731	0.374	18.9
	✔		0.395	0.496	0.703	0.363	27.1
	✔	✔	**0.402**	**0.505**	**0.660**	**0.316**	27.1
✔	✔	✔	**0.402**	0.503	0.662	0.341	18.6

5.5. Failure Cases

We show the detection results of a challenging scene in Fig. 6. StreamPETR shows impressive results on crowded objects within the detection range of 30m. However, our method has many False Positives on remote objects. It is a common phenomenon of camera-based methods. In a complex urban scene, the duplicated predictions on remote objects can be tolerable and cause relatively little impact.

6. Conclusion

In this paper, we propose StreamPETR, an effective long-sequence 3D object detector. Different from the previous works, our method explores an object-centric paradigm that propagates temporal information through object queries frame by frame. In addition, a motion-aware layer normalization is adopted to introduce the motion information. StreamPETR achieves leading performance improvements while introducing negligible storage and computation cost. It is the first online multi-view method that achieves comparable performance with lidar-based methods. We hope StreamPETR can provide some new insights into long-sequence modeling for the community.

A. Appendix

A.1. Algorithm Workflow

Algorithm 1 Propagation Transformer

Input: Multi-view 2D features from streaming video $\boldsymbol{F}_{2d} = \{F_{2d}^1, F_{2d}^2, \cdots, F_{2d}^T\}$. A set of learnable reference points Q_p^{init}.

Memory Queue: Memory queue of N historical frames $\boldsymbol{X} = \{X^{t-N}, \cdots, X^{t-2}, X^{t-1}\}$. For each time stamp $t-k$, we maintain the query states with motion information, $X^{t-k} = \{\triangle t^{t-k}, Q_c^{t-k}, Q_p^{t-k}, v^{t-k}, E^{t-k}\}$.

Output: 3D bounding boxes prediction with classification scores $\boldsymbol{b}^t = (x, y, z, l, w, h, \theta, v_x, v_y, cls)$.

$\boldsymbol{X} \leftarrow \emptyset$
for $t \in \boldsymbol{T}$ **do**
  (1) Motion compensation
  $[\tilde{Q}_p^{t-N:t-1}, \tilde{E}_{t-N:t-1}^t] \leftarrow Ego([Q_p^{t-N:t-1}, E^{t-N:t-1}])$
  $\tilde{Q}_{pe}^{t-N:t-1} \leftarrow \psi(\tilde{Q}_p^{t-N:t-1})$
  $M \leftarrow [\tilde{E}_{t-N:t-1}^t, \triangle t^{t-N:t-1}, v^{t-N:t-1}]$
  $\tilde{Q}_{pe}^{t-N:t-1} \leftarrow MLN(\tilde{Q}_{pe}^{t-N:t-1}, M)$
  $\tilde{Q}_c^{t-N:t-1} \leftarrow MLN(Q_c^{t-N:t-1}, M)$
  (2) Propagate query
  $Q_{pe}^{init} \leftarrow \psi(Q_p^{init})$
  $Q_c^{init} \leftarrow Zero\_Like(Q_{pe}^{init})$
  $Q_{pe}^0 \leftarrow Concat([Q_{pe}^{init}, \tilde{Q}_{pe}^{t-1}])$
  $Q_c^0 \leftarrow Concat[Q_c^{init}, \tilde{Q}_c^{t-1}]$
  (3) Spatial-temporal interaction
  for $i \in L$ **do**
    $\tilde{Q}_{pe}^{hybrid} \leftarrow Concat([Q_{pe}^i, \tilde{Q}_{pe}^{t-N:t-1}])$
    $\tilde{Q}_c^{hybrid} \leftarrow Concat[Q_c^i, \tilde{Q}_c^{t-N:t-1}]$
    $Q_c^i \leftarrow Hybrid\_Attn([Q_c^i, Q_{pe}, \tilde{Q}_c^{hybrid}, \tilde{Q}_{pe}^{hybrid}])$
    $Q_c^{i+1} \leftarrow Cross\_Attn([Q_c^i, Q_{pe}, F_{2d}^t, F_{3d\_pe}^t])$
    $Q_{pe}^{i+1} \leftarrow Q_{pe}^i$
  end for
  (4) Update memory queue
  $\boldsymbol{b}^t \leftarrow Head(Q_c^L)$
  $index \leftarrow TopK(\boldsymbol{b}^t)$
  $X^t \leftarrow Gather(index)$
  $\boldsymbol{X} \leftarrow \{X^{t-N+1}, \cdots, X^{t-1}, X^t\}$
end for

The workflow of our proposed Propagation Transformer is shown in Alg. 1, which is divided into four stages:

(1) The motion compensation takes the information of the memory queue as input (including the relative time interval $\triangle t$, context embedding Q_c, object center Q_p, velocity v, and ego-pose matrix E). The object 3D centers $Q_p^{t-N:t-1}$ in the memory queue are explicitly aligned to the current frame according to the ego pose Ego. Then the aligned centers $\tilde{Q}_p^{t-N:t-1}$ are used to generate position encoding of object query by a single MLP layer ψ. Afterward, we apply the Motion aware Layer Normalization (MLN) to encode motion information M into the historical queries.

(2) The generation of the object queries is mainly based on the learnable query embedding $[Q_{pe}^{init}, Q_c^{init}]$ and the obtained propagated query embedding $[\tilde{Q}_{pe}^{t-1}, \tilde{Q}_c^{t-1}]$ from the last frame $t-1$.

(3) The spatial-temporal interaction of the Propagation Transformer has stacked L layers of the hybrid attention ($Hybrid\_Attn$) and cross attention ($Cross\_Attn$). For each layer, the hybrid attention performs the interaction of current queries $[Q_c^i, Q_{pe}^i]$ and historical queries $[\tilde{Q}_c^{hybrid}, \tilde{Q}_{pe}^{hybrid}]$, and the cross attention performs the interaction of current queries and image tokens $[F_{2d}^t, F_{3d\_pe}^t]$. $F_{3d\_pe}^t$ is the 3D position encoding proposed in PETR [30].

(4) After the layer-by-layer refinement, a 3D Head ($Head$) are conducted to generate the predictions. Then we select top-K foreground objects according to the classification scores and push the information of the selected objects to the memory queue $\boldsymbol{X}$.

Table 9. Flash Attention for efficient training (V2-99 [21] backbone with input resolution of 1600×640).

Flash Attn	A100 Training Time (s/iter) ↓	GPU Memory (G)↓
✔	**1.51**	**27G**
✘	1.68	61G

Table 10. Applicability of our method. We extend object-centric temporal modeling to DETR3D.

Method	mAP↑	NDS↑	mATE↓	mAVE↓	FPS↑
DETR3D	0.347	0.422	0.765	0.876	**6.3**
Stream-DETR3D	**0.396**	**0.490**	**0.723**	**0.487**	6.2

A.2. Additional Details

Qualitative results of StreamPETR are provided in video.

Here we provide more details for reproducing the results. First, we detach the gradient of the first 6 frames and compute the gradient and losses of the last 2 frames, which can accelerate the convergence. We additionally adopt Flash Attention [6] to further save the GPU memory, as shown in Tab. 9. The query denoising [22] is also conducted following PETRv2 [31]. When measuring the inference speed, we close the Flash Attention.

A.3. Extension of Our Method

To verify the extensibility of our method, we conduct experiments on another sparse query base model DETR3D [47]. We use ResNet101-DCN as the backbone, without additional augmentation and CBGS [59]. Results in Tab. 10 show that StreamDETR3D brings 4.9% and 6.8% improvements on mAP and NDS, while the inference speed is little impacted. Compared with the PETR paradigm, the improvement of DETR3D is relatively small. One possible reason is that the local spatial attention adopted by DETR3D limits the performance.

References

[1] 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. 5, 6, 8

[2] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8100, 2022. 2

[3] Yuxuan Cai, Yizhuang Zhou, Qi Han, Jianjian Sun, Xiangwen Kong, Jun Li, and Xiangyu Zhang. Reversible column networks. *arXiv preprint arXiv:2212.11696*, 2022. 6

[4] 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. 1, 2, 5

[5] Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Chang Huang, and Wenyu Liu. Polar parametrization for vision-based surround-view 3d detection. *arXiv preprint arXiv:2206.10965*, 2022. 2, 5, 6

[6] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *arXiv preprint arXiv:2205.14135*, 2022. 9

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[9] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016. 4

[10] Chengjian Feng, Zequn Jie, Yujie Zhong, Xiangxiang Chu, and Lin Ma. Aedet: Azimuth-invariant multi-view 3d object detection. *arXiv preprint arXiv:2211.12501*, 2022. 6

[11] Tobias Fischer, Yung-Hsu Yang, Suryansh Kumar, Min Sun, and Fisher Yu. Cc-3dt: Panoramic 3d object tracking via cross-camera fusion. *arXiv preprint arXiv:2212.01247*, 2022. 6

[12] Fei He, Naiyu Gao, Jian Jia, Xin Zhao, and Kaiqi Huang. Queryprop: Object query propagation for high-performance video object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 834–842, 2022. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[14] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular quasi-dense 3d object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1992–2008, 2022. 6

[15] Bin Huang, Yangguang Li, Enze Xie, Feng Liang, Luya Wang, Mingzhu Shen, Fenggang Liu, Tianqi Wang, Ping Luo, and Jing Shao. Fast-bev: Towards real-time on-vehicle bird's-eye view perception. *arXiv preprint arXiv:2301.07870*, 2023. 2

[16] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:/2203.17054*, 2021. 1, 2, 3, 5, 6

[17] Junjie Huang and Guan Huang. Bevpoolv2: A cutting-edge implementation of bevdet toward deployment. *arXiv preprint arXiv:2211.17111*, 2022. 5, 8

[18] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2, 5, 6

[19] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformers. *arXiv preprint arXiv:2206.15398*, 2022. 2, 6

[20] Zhengkai Jiang, Peng Gao, Chaoxu Guo, Qian Zhang, Shiming Xiang, and Chunhong Pan. Video object detection with locally-weighted deformable neighbors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8529–8536, 2019. 2

[21] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 6, 9

[22] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 7, 9

[23] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo. *arXiv preprint arXiv:2209.10248*, 2022. 2, 5, 6, 8

[24] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *arXiv preprint arXiv:2206.00630*, 2022. 6

[25] 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. *arXiv preprint arXiv:2206.10092*, 2022. 1, 2, 3, 5, 6

[26] Zhiqi Li, Hanming Deng, Tianyu Li, Yangyi Huang, Conghao Sima, Xiangwei Geng, Yulu Gao, Wenhai Wang, Yang Li, and Lewei Lu. Bevformer ++ : Improving bevformer for 3d camera-only object detection: 1st place solution for waymo open dataset challenge 2022. 2023. 6, 7

[27] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer:

Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 1, 2, 3, 5, 6

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[29] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. *arXiv preprint arXiv:2211.10581*, 2022. 1, 2, 3, 5, 6

[30] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 2, 4, 5, 6, 9

[31] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 1, 2, 3, 5, 6, 8, 9

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 6

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[34] Zhipeng Luo, Changqing Zhou, Gongjie Zhang, and Shijian Lu. Detr4d: Direct multi-view 3d object detection with sparse attention. *arXiv preprint arXiv:2212.07849*, 2022. 2, 3

[35] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8844–8854, 2022. 2

[36] Ziqi Pang, Jie Li, Pavel Tokmakov, Dian Chen, Sergey Zagoruyko, and Yu-Xiong Wang. Standing between past and future: Spatio-temporal modeling for multi-camera 3d multi-object tracking. *arXiv preprint arXiv:2302.03802*, 2023. 2, 6

[37] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pages 680–696. Springer, 2023. 6

[38] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021. 6

[39] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. *arXiv preprint arXiv:2210.02443*, 2022. 1, 2, 5, 6

[40] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Gaugan: semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH 2019 Real-Time Live!*, pages 1–1. 2019. 4

[41] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 6

[42] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 5

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[44] Shihao Wang, Xiaohui Jiang, and Ying Li. Focal-petr: Embracing foreground for efficient multi-camera 3d object detection. *arXiv preprint arXiv:2212.05505*, 2022. 5, 6

[45] Tai Wang, Qing Lian, Chenming Zhu, Xinge Zhu, and Wenwei Zhang. Mv-fcos3d++: Multi-view camera-only 4d object detection with pretrained monocular backbones. *arXiv preprint arXiv:2207.12716*, 2022. 6, 7

[46] 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. 6

[47] Yue Wang, Guizilini Vitor Campagnolo, Tianyuan Zhang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *In Conference on Robot Learning*, pages 180–191, 2022. 2, 4, 5, 6, 9

[48] Zitian Wang, Zehao Huang, Jiahui Fu, Naiyan Wang, and Si Liu. Object as query: Equipping any 2d object detector with 3d detection ability. *arXiv preprint arXiv:2301.02364*, 2023. 2, 6

[49] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. Mˆ 2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 2

[50] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. *arXiv preprint arXiv:2211.10439*, 2022. 5, 6

[51] Jinrong Yang, En Yu, Zeming Li, Xiaoping Li, and Wenbing Tao. Quality matters: Embracing quality clues for robust 3d multi-object tracking. *arXiv preprint arXiv:2208.10976*, 2022. 6

[52] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3d-man: 3d multi-frame attention network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1863–1872, 2021. 2

[53] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 6, 7

[54] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 659–675. Springer, 2022. 2, 3

[55] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 4

[56] Tianyuan Zhang, Xuanyao Chen, Yue Wang, Yilun Wang, and Hang Zhao. Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4537–4546, 2022. 2, 6

[57] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pre-trained object detectors. *arXiv preprint arXiv:2211.09791*, 2022. 2

[58] Yifu Zhang, Xinggang Wang, Xiaoqing Ye, Wei Zhang, Jincheng Lu, Xiao Tan, Errui Ding, Peize Sun, and Jingdong Wang. Bytetrackv2: 2d and 3d multi-object tracking by associating every detection box. *arXiv preprint arXiv:2303.15334*, 2023. 6, 7

[59] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 6, 8, 9

[60] 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. 1