SRCN3D: Sparse R-CNN 3D for Compact Convolutional Multi-View 3D Object Detection and Tracking

Yining Shi[1], Jingyan Shen[1], Yifan Sun[1], Yunlong Wang[1],
Jiaxin Li[1], Shiqi Sun[1], Kun Jiang[1†], Diange Yang[1†]
[1] Tsinghua University

Abstract

Detection and tracking of moving objects is an essential component in environmental perception for autonomous driving. In the flourishing field of multi-view 3D camera-based detectors, different transformer-based pipelines are designed to learn queries in 3D space from 2D feature maps of perspective views, but the dominant dense BEV query mechanism is computationally inefficient. This paper proposes Sparse R-CNN 3D (SRCN3D), a novel two-stage fully-sparse detector that incorporates sparse queries, sparse attention with box-wise sampling, and sparse prediction. SRCN3D adopts a cascade structure with the twin-track update of both a fixed number of query boxes and latent query features. Our novel sparse feature sampling module only utilizes local 2D region of interest (RoI) features calculated by the projection of 3D query boxes for further box refinement, leading to a fully-convolutional and deployment-friendly pipeline. For multi-object tracking, motion features, query features and RoI features are comprehensively utilized in multi-hypotheses data association. Extensive experiments on nuScenes dataset demonstrate that SRCN3D achieves competitive performance in both 3D object detection and multi-object tracking tasks, while also exhibiting superior efficiency compared to transformer-based methods. Code and models are available at https://github.com/synsin0/SRCN3D.

1. Introduction

Environmental perception is an essential task in the field of autonomous driving. 3D object detection and tracking are responsible for identifying and localizing objects of interest, as well as recording their unique labels and past trajectories. While LiDAR is commonly used for this purpose, on-board cameras offer certain advantages such as lower cost, wider detection range, higher angular resolution, and richer semantic cues. However, vision-centric detectors face two long-standing challenges. Firstly, since cameras lack geometric or depth cues, 3D reconstruction can be an ill-conditioned problem. Secondly, there is the issue of cross-view fusion, which refers to the challenge of detecting an object as a whole when two adjacent cameras only capture parts of it.

Vision-centric detectors rise rapidly in accuracy thanks to latest innovation of data-driven view transformation from perspective view to 3D world space. LSS [24] proposes a geometry-based explicit pipeline that includes depth estimation, point-cloud lifting, voxelization, and splatting, while DETR3D [28] introduces a network-based implicit pipeline learning queries from projected reference points to obtain values without explicit depth estimation and post-processing. As illustrated in Fig. 1, previous arts require either dense queries or dense interaction between queries and values. Our proposed method enjoys the merit of implicit pipeline and avoids dense feature sampling at the same time.

Our proposed method, SRCN3D provides a simple and elegant cascade pipeline and set prediction approach highly inspired by Sparse R-CNN [25]. We adopt commonly used backbones and Feature Pyramid Network (FPN) [17] neck, but a novel SRCN3D head that iteratively updates both 3D query boxes and query features at the same time. Specifically, 3D query boxes are projected to six views to aggregate local RoI features. The resulting features are used to refine query features via a sparse interaction head, which produces classification and the offsets that are applied to the original bounding boxes. After three to six stages, the final refined query boxes serve as the direct detection outputs, eliminating the need for complicated post-processing steps or regression processes from latent features. Our framework does not rely on transformer-style operations such as masking operations or positional embeddings.

Besides, in a mainstream tracking-by-detection (TBD) pipeline, multi-object tracking is challenging in the data association process given possible missed or false detection. This problem is more common for camera-based detectors than LiDAR-based detectors. Therefore, conventional deterministic data association suffers from poor track conti-

†Corresponding author: Kun Jiang, Diange Yang

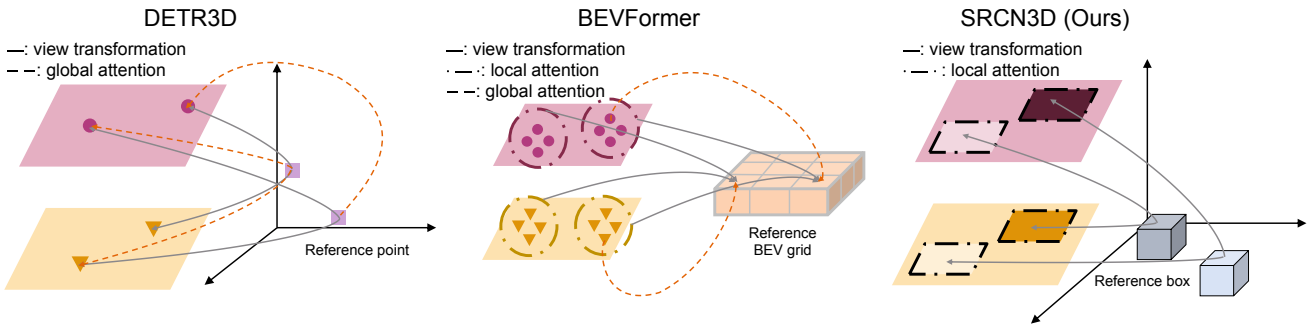

Figure 1. Comparison between typical query-based multi-view 3D (MV3D) detectors. DETR3D [28] applies sparse queries and dense global attention. BEVFormer [16] applies dense queries and dense deformable attention. SRCN3D steps further to sparser queries and totally sparse local RoI attention, which pioneers a fully-sparse box-wise sampling paradigm.

nuity. Recent trackers, such as QD-3DT [12], incorporate dense image features, while re-identification (Re-ID) from dense region queries is inefficient and inaccurate. In light of the instability of detection results, we find out that a simplified multi-hypotheses Random Finite Set (RFS) approach for probabilistic matching reduces failed tracking. Our proposed tracker is also the first approach to incorporate RoI features and query features in the RFS framework.

In summary, this paper makes the following contributions:

- To the best of our knowledge, **the first transformer-less two-stage MV3D approach with box-wise sampling.** The pipeline is straightforward, lightweight, and faster than other transformer-based detectors.
- **A novel sparse cross-attention module to refine 3D queries from 2D feature maps**, which replaces dense attention with a local sparse interaction module. Consequently, a lower computation cost is achieved.
- Extensive experiments on nuScenes dataset demonstrate the effectiveness of SRCN3D for 3D object detection and tracking.

2. Related Work

2.1. Multi-camera 3D Object Detection

The development of MV3D has progressed rapidly due to the use of data-driven view transformation. Currently, there are three primary paradigms for state-of-the-art multi-view camera 3D object detection: perspective-view-based (PV-based), geometry-based, and network-based view transformation. PV-based view transformation involves predicting 3D boxes from 2D images in perspective view and performing BEV aggregation [23, 26, 27, 31], which presents challenges in cross-view fusion. Geometry-based view transformation involves depth estimation, followed by transformation to world coordinates and aggregation of point-clouds in a bottom-up manner. However, this approach is sensitive to depth errors. Network-based view transformation, on the other hand, involves randomly proposing 3D queries and refining them in a cascade [16, 19, 28], without depth estimation. Our proposed method adopts the network-based paradigm and introduces a novel two-stage pipeline that differs from previous single-stage approaches.

2.2. Transformer-based Object Detection

Vision transformers have demonstrated excellent performance in object detection. In 2D domain, DETR [3] presents a set-prediction paradigm with regard to a fixed set of objects. Deformable DETR [35] invents deformable attention for faster convergence. Furthermore, Sparse R-CNN [25] puts forward a completely sparse schema, where each query box interacts only with its specific query feature. In 3D domain, transformers serve as cross-attention between 3D queries and 2D feature maps [16, 19, 20, 28]. DETR3D [28] is the first to apply a top-down framework, which projects reference points on feature maps and performs cross-attention to refine query features. BEVFormer [16] leverages cross-attention on bird's eye view (BEV) grid features, and employs spatial deformable attention for BEV grids and temporal alignment of BEV features. PETR [19] and PETRv2 [20] adopt ideas from implicit neural representation and project 2D features maps to 3D space so as to interact with 3D queries. Compared to dense attention arts, our method explores a sparse feature sampling module without global attention and makes itself a purely sparse approach.

2.3. 3D Multi-Object Tracking

3D Multi-Object Tracking (3D MOT) is another challenging task right after object detection, aiming to temporally associate trajectories of each same object and record

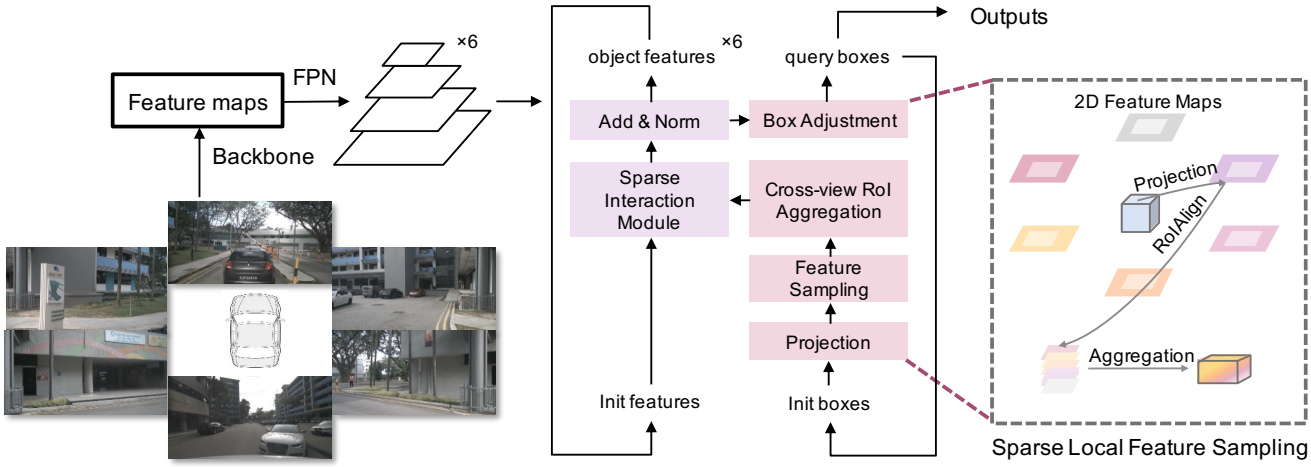

Figure 2. The framework of SRCN3D. Taken camera images as inputs, SRCN3D contains a backbone network with FPN to extract 2D feature maps and a twin-track detection module. A sparse feature sampling module is designed to extract local RoI features and refine the query boxes.

its unique label. Data association is the core issue of MOT, where the dominant matching approach is Global Nearest Neighbor (GNN), carried out in the form of Hungarian or Greedy algorithm used in AB3DMOT [29]. Another approach is RFS [9], an online multi-hypotheses paradigm circumventing deterministic one-to-one pair matching. Current MOT researches, e.g. CenterTrack [33], QD-3DT [12] and MUTR3D [32], focus on utilizing implicit features to express matching similarity in the embedded space, empowering the network to identify the same object based on appearance features through Re-ID process. Our tracker explores a novel multi-hypotheses probabilistic mode of data associated with hybrid feature embedding.

3. Methodology

3.1. Overview

The overall framework of SRCN3D is illustrated in Fig. 2. SRCN3D has a novel two-stage fully-convolutional 3D object detection architecture. We build our architecture based on the following consideration:

- We estimate 3D bounding boxes directly in 3D world space without depth supervision and post-processing like non-maximum suppression (NMS), while 2D images serve as implicit cues in our network.
- Following a sparse paradigm, each query box serves as a filter to focus on a sparse local region of 2D feature maps.
- We design a fully-convolutional pipeline without mask operations, positional embeddings and attention weights in typical vision transformers.

The architecture of SRCN3D consists of a common backbone with FPN and a novel SRCN3D head. First of all, we feed the RGB images into a backbone network (e.g. ResNet-101 [11]) with FPN [17] to generate multi-level multi-camera feature maps $\{F_1^i, F_2^i, F_3^i, F_4^i\}_{i=1}^{N_{cam}}$ for each view. SRCN3D heads start from a fixed set of 3D learnable query boxes and query features, and the initial parameters for boxes and features are randomly initialized and learnable during training process. In a cascade structure, the query boxes and query features compose strict pairs while non-pairs do not interact. Boxes and features are updated in a twin-track approach. Query boxes are updated in a box adjustment step and query features are updated with a sparse feature sampling module. The last stage of query boxes and class regression of query features without post-processing techniques comprise the final results.

3.2. SRCN3D Head

As shown in Fig. 3, SRCN3D head composes of three key modules, that is, a sparse feature sampling module for feature extraction, a sparse interaction module for feature refinement and a box adjustment module for box refinement. For each iteration round, there are two inputs: a fixed number of learnable query boxes and query features. The forward process for each round includes the following steps:

- Restore query boxes to world scale, retrieve eight corner points and project corner points to images using corresponding intrinsic and extrinsic camera parameters.
- Collect four borderline corner points from each image to form a RoI candidate, sample RoI features using RoIAlign and aggregate cross-view RoI features.

- Feed cross-view RoI features into sparse interaction head to refine query features and output cues (fine-tuning values) for box adjustment.

- Refine location, dimension, rotation and velocity with box adjustment module, normalize the 3D positions of the boxes and obtain the input for the next round.

3D query boxes are defined as a fixed number of boxes parameterized to the same dimension as 3D bounding box (e.g. $\{B_i\}_{i=1}^{N} \subset R^{10}, N = 300$). The 10 dimensions are defined as $[c_x, c_y, h, w, c_z, l, cos\theta, sin\theta, v_x, v_y]$, where c_x, c_y, c_z are center coordinates of the box, h, w, l are height, weight and length, θ is the yaw angle and v_x, v_y are velocities.

Query features are represented by sets of high-dimensional latent vectors (e.g. $\{f_i\}_{i=1}^{N} \subset R^{256}$), strictly corresponding to 3D query boxes.

Sparse interaction head. The sparse interaction head implements local interaction by applying 1×1 convolutional kernels on RoI features extracted from query boxes and generating corresponding parameters from query features via linear transformation. The inputs are passed through two 1×1 convolutional layers for interaction, followed by a Feed-Forward Network (FFN) block with layer normalization and a linear projection block to output classification and regression predictions.

3.3. Sparse Feature Sampling Module

The 3D query boxes, which are learnable, serve as sparse candidates that are updated iteratively. We decode 3D query boxes from center points to box corners through simple geometric transformations. For simplicity, we refer the i th decoded box $\{C_{il}\}_{l=1}^{8} \subset \mathbb{R}^3$ with coordinates of eight corners. Through a standard camera model, these query boxes are projected into visible regions of cameras as follows:

$$C_{il}^{*} = C_{il} \oplus 1, C_{mil} = T_m C_{il}^{*}, \tag{1}$$

where $l = 1, \ldots, 8, m = 1, \ldots, N_{cam}, C_{mil} = (c_{milx}, c_{mily}, 1)$ and T_m denotes the camera transformation matrix. Then the projected boxes on each camera can be obtained as follows:

$$\tilde{B}_{im} = (\min_{l} c_{imlx}, \min_{l} c_{imly}, \max_{l} c_{imlx}, \max_{l} c_{imly}), \tag{2}$$

where $i = 1, \ldots, N, m = 1, \ldots, N_{cam}$.

As shown in Fig .3, given the projected boxes, we use RoI Align operation to extract features of interest. The projection of box corner points may result in three cases. The normal case indicates a projected 2D box on images. If the projected points have a negative depth, the box locates behind the camera, which is naturally invisible. If the projected corners are outside or partially outside the pixel space, the box is (partially) invisible. The latter two abnormal feature sampling cases result in empty or partially empty RoI features via RoIAlign, so that masking operations and positional encoding techniques are deftly avoided.

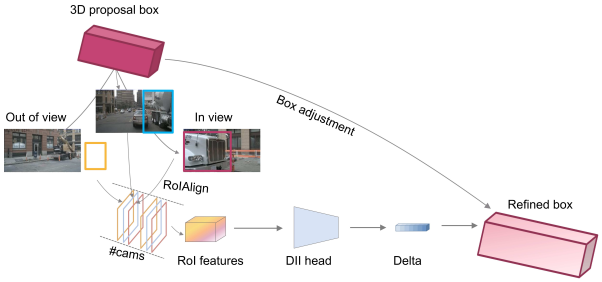

Figure 3. A graphical illustration of sparse feature sampling module. 3D query boxes are initialized and projected into feature maps in the camera plane for RoI extraction. After aggregation, RoI features interact with query features in DII head which produces cues for refinement. Finally, box adjustment module refines the boxes.

Cross-view fusion. Before entering the prediction head, corresponding RoI features on multi-view camera images are aggregated to guarantee cross-view fusion learning. In this way, RoI features maintain a fixed expression, no matter how many cameras capture one query box.

3.4. Box Adjustment

We define the predicted boxes and cues in the t th stage as $\{b_i^t\}_{i=1}^{N}$ and $\{\Delta b_i^t\}_{i=1}^{N}$ respectively. Then the box adjustment operation for location and dimension parameters can be formulated as follows:

$$\begin{cases} b_{ix}^{t+1} = \Delta b_{ix}^{t+1} \times b_{iw}^{t} + b_{ix}^{t} \\ b_{iy}^{t+1} = \Delta b_{iy}^{t+1} \times b_{il}^{t} + b_{iy}^{t} \\ b_{iz}^{t+1} = \Delta b_{iz}^{t+1} \times b_{ih}^{t} + b_{iz}^{t} \end{cases}. \tag{3}$$

Considering non-negative constraints, dimension parameters are usually in the form of logarithms. Therefore, its adjustment is formulated as follows:

$$\begin{cases} b_{iw}^{t+1} = e^{\Delta b_{iw}^{t+1}} \times b_{iw}^{t} \\ b_{il}^{t+1} = e^{\Delta b_{il}^{t+1}} \times b_{il}^{t} \\ b_{ih}^{t+1} = e^{\Delta b_{ih}^{t+1}} \times b_{ih}^{t} \end{cases}. \tag{4}$$

As for rotation and velocity, we directly take the values in $\{\Delta b_i^t\}_{i=1}^{N}$ as the predicted results of the t th stage.

3.5. Loss Design

Generally, the loss function of SRCN3D is a linear combination of a Focal Loss [18] for category classification and a L1 norm loss for 3D bounding box regression, which is as follows:

$$\mathcal{L} = \omega_{cls} \times \mathcal{L}_{cls} + \omega_{l_{reg}} \times \mathcal{L}_1. \tag{5}$$

SRCN3D employs set prediction loss following DETR [3]. Details of set prediction loss are presented in the supplementary material.

3.6. Multi-object Tracking

Multi-object Tracking mainly handles data association of detected objects between past and current frames. Given nonideal detection results, we adopt a hypothesis-oriented probabilistic approach, Multi-Bernoulli Mixture (MBM) to deal with uncertain data association. MBM treats data association into global hypotheses and single target hypotheses. The MBM density is defined as the sum of j global hypotheses as follows:

$$f^{\mathrm{mbm}}(X) \propto \sum_{j} \sum_{X_1 \uplus \ldots \uplus X_n = X} \prod_{i=1}^{n} w_{j,i} f_{j,i}\left(X_i\right), \quad (6)$$

where $f^{\mathrm{mbm}}(X)$ is the posterior of MBM intensity, X is the whole set of detected objects, $w_{j,i}$ indicates the weight of a Bernoulli component X_i in global hypothesis j, and $f_{j,i}\left(X_i\right)$ is the probability intensity of single Bernoulli component X_i in global hypothesis j, defined as follows:

$$f_{j,i}(X_i) = \begin{cases} 1 - r_{j,i} & \text{if } X_i = \oslash \\ r_{j,i} p_{j,i}(x) & \text{if } X_i = x \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

where $r_{j,i}$ denotes the existence probability of object x in single target hypothesis (STH) X and $p_{j,i}$ is the probability density function considering the log likelihood of the hypothesis target. Each STH denotes one object detected or a missed detection. Temporal prediction and update of states follow a standard unscented kalman filter (UKF). An important criterion for matching is to compute likelihood between measurements and states. In SRCN3D tracker, each measurement includes explicit properties of 3D bounding boxes and two kinds of implicit features, namely, RoI features and query features. The likelihood of 3D bounding boxes is computed by Mahanalobis distance introduced in [7], and latent RoI features and query features follow a cosine similarity. The overall likelihood is calculated as

$$l = l_{box} + \alpha \times l_{RoI} + \beta \times l_{prop}. \quad (8)$$

In practice, we set $\alpha = \beta = 0.5$ to achieve a balance among box attributes and appearance.

4. Experiments

4.1. Dataset

We report experiment results on large-scale public nuScenes dataset [2], which includes 1000 driving scenes of about 20 seconds duration. RGB images are collected from 6 cameras with known intrinsic and extrinsic camera parameters. NuScenes dataset provides 28130, 6019 and 6008 samples for training, validation and testing, respectively. Only key frames at 2Hz are annotated and used.

4.2. Metrics

We adopt the nuScenes [2] official evaluation protocol. As for detection metrics, we adopt mean average precision (mAP) and nuScenes Detection Score (NDS) as primary metrics, and true positive metrics (TP metrics) including average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE), and average attribute error (AAE). Metrics of MOT are based on CLEARMOT [1], including average multi object tracking accuracy (AMOTA) as the primary metric, and average multi-object tracking precision (AMOTP) and recall rate (RECALL). Reports of SRCN3D on all detection and tracking metrics are publicly available on nuScenes leaderboard.

4.3. Training and Inference

Training. Our code is built upon the MMDetection3D [22]. We employ two types of backbone network: (1) ResNet101 [11] pretrained on FCOS3D [27] and PGD [26], (2) VoVNetV2-99 [15] pretrained on DD3D [23]. We use a bipartite loss via 3D Hungarian assigner, which consists of a Focal Loss [18] with weight 2.0 and a L1 loss with weight 0.25. The model is trained using AdamW [21] optimizer with weight decay of 0.01. The learning rate is initialized with $2e^{-4}$ and decays with cosine annealing policy. We set the detection region to $[-61.2m, 61.2m]$ for the X and Y axis, and $[-5m, 3m]$ for the Z axis. Experiments are trained for 24 epochs on 8 2080TI GPUs, the training hours is around 24 hours.

Inference. In the inference process, SRCN3D makes simple predictions without any post-processing step such as NMS and test-time augmentation (TTA). Our model infers within 3.2 FPS on a single RTX3090 GPU with ResNet 101 as the backbone and the original image size 1600×900 as shown in Table. 1, faster than other transformer-based detectors like DETR3D [28] with sparse queries (2.7 FPS), BEVFormer [16] (2.1 FPS) with dense queries and PETR [19] with hybrid queries.

4.4. Comparison with State-of-the-art

nuScenes detection benchmark. In Table. 1, we present the performance comparison with state-of-the-art methods on nuScenes validation set. SRCN3D gains 42.8% NDS and 33.8% mAP for ResNet101 [11] backbone and image size of 900×1600. Compared to Monocular 3D detectors, SRCN3D surpasses CenterNet [8] and FCOS3D [27] in NDS by 10.1% and 1.3%. Compared to MV3D detectors, SRCN3D also shows competitive results. Un-

Method	Size	Backbone	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	FPS↑
CenterNet [8]	-	DLA	0.328	0.306	0.716	0.264	0.609	1.426	0.658	-
FCOS3D ‡# [27]	1600×900	Res-101	0.415	0.343	0.725	**0.263**	0.422	1.292	**0.153**	2.0
DETR3D ¶ [28]	1600×900	Res-101	0.425	0.346	0.773	0.268	0.383	0.842	0.216	2.7
BEVDet § [14]	1056×384	Res-101	0.396	0.330	**0.702**	0.272	0.534	0.932	0.251	**16.7**
BEVFormer-S ¶ [16]	1600×900	Res-101	**0.448**	**0.375**	0.725	0.272	0.391	0.802	0.200	2.1
PETR §¶ [19]	1600×900	Res-101	0.442	0.370	0.711	0.267	0.383	0.865	0.201	2.5
SRCN3D (Ours)¶	1600×900	Res-101	0.428	0.337	0.779	0.287	**0.367**	**0.781**	0.188	3.2
SRCN3D (Ours)¶	1600×900	V2-99	0.475	0.396	0.737	0.294	0.278	0.728	0.197	2.5

Table 1. Comparison of state-of-the-art detectors on nuScenes detection val set. ‡: with test-time augmentation. §: trained with CBGS [34]. ¶: initialized from pretrained FCOS3D [27] backbone. #: with model ensemble.

Method	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
CenterNet [8]	0.400	0.338	0.658	0.255	0.629	1.629	0.142
EPro-PnP-Det [5]	0.453	0.373	0.605	0.243	**0.359**	1.067	**0.124**
M2BEV [30]	0.451	0.398	0.577	0.245	0.500	1.227	0.154
FCOS3D ‡ [27]	0.428	0.358	0.690	0.249	0.452	1.434	**0.124**
BEVFormer-S [16]	0.462	0.409	0.650	0.261	0.439	0.925	0.147
DETR3D † [28]	0.479	0.412	0.641	0.255	0.394	0.845	0.133
DD3D †‡ [23]	0.477	0.418	0.572	0.249	0.368	1.014	0.124
BEVDet † [14]	0.488	0.424	**0.524**	**0.242**	0.373	0.950	0.148
PETR † [19]	**0.504**	**0.441**	0.593	0.249	0.383	**0.808**	0.132
SRCN3D †(Ours)	0.463	0.396	0.673	0.269	0.403	0.875	0.129

Table 2. Comparison of state-of-the-art detectors on nuScenes detection test set. Detection methods using temporal aggregation are not included. †: trained using extra data. ‡: with test time augmentation.

der the same backbone and image size settings, our method outperforms DETR3D [28] by 0.3% in NDS. In terms of TP metrics, it shows that SRCN3D works well in predicting orientation, velocity and attributes, achieving the best performance in mAOE and mAVE. However, SRCN3D still suffer from limitations for translation and scale predictions. Table. 2 shows the performance comparison on nuScenes detection test set. Our method achieves competitive performance on NDS, mAP and other true positive metrics. Overall, the experimental results demonstrate the effectiveness of our method on 3D object detection tasks.

nuScenes tracking benchmark. Table. 3 reports nuScenes tracking benchmark on both validation and test split. SRCN3D achieves state-of-the-art performance in camera track and exceeds other competitors by a large margin. On validation set, compared with camera based methods, SRCN3D achieves the best performance in all reported metrics. On test set, our method achieves 0.398 in terms of AMOTA metrics on nuScenes test set, more than 12 points of accuracy improvement over recent state-of-the-art camera-only trackers. The test results also show a moderate AMOTP error and the highest recall rate.

4.5. Ablation Study

In this section, we perform ablations on several important components or properties of SRCN3D on nuScenes validation set.

Ablation on key modules. We have identified four key modules that distinguish SRCN3D from previous approaches: learnable boxes, sparse interaction module, box adjustment, and initialization. To evaluate the importance of these modules, we perform ablation experiments by replacing them with similar parts in DETR3D. Table 4 shows the results. In the first case, we construct a two-stage DETR3D, showing that query features alone are not sufficient for refining query boxes. The second case regresses boxes directly using a regression branch of query features at the beginning. The third case removes the box adjustment step and directly inputs object features into the regression branch, resulting in a slightly reduced accuracy. Additionally, random initialization of query boxes outperforms fixed initialization. We also test the dynamic instance interaction head introduced in Sparse R-CNN, which has an additional self-attention module. Our sparse interaction without self-attention is equally effective, demonstrating that we do not need self-attention to distinguish different queries.

Ablation on number of queries, objects and stages.

Method	Modality	Split	AMOTA↑	AMOTP↓	RECALL↑
QD3DT [12]	C	val	0.242	1.518	0.399
MUTR3D [32]	C	val	0.294	1.498	0.427
ViP3D [10]	C	val	0.216	1.616	0.358
UniAD [13]	C	val	0.359	1.320	0.467
SRCN3D (Ours)	C	val	**0.439**	**1.280**	**0.545**
CenterTrack-Open [33]	L + C	test	0.108	**0.989**	0.412
QD-3DT [12]	C	test	0.217	1.550	0.375
PolarDETR [6]	C	test	0.273	1.185	0.404
DEFT [4]	C	test	0.177	1.564	0.338
MUTR3D [32]	C	test	0.270	1.494	0.411
SRCN3D (Ours)	C	test	**0.398**	1.317	**0.538**

Table 3. Comparison of state-of-the-art trackers on nuScenes tracking benchmark. The quantitative results for the validation set are obtained from the original paper, while the results for the test set are obtained from the nuScenes leaderboard. For modalities, "L" denotes LiDAR and "C" denotes camera.

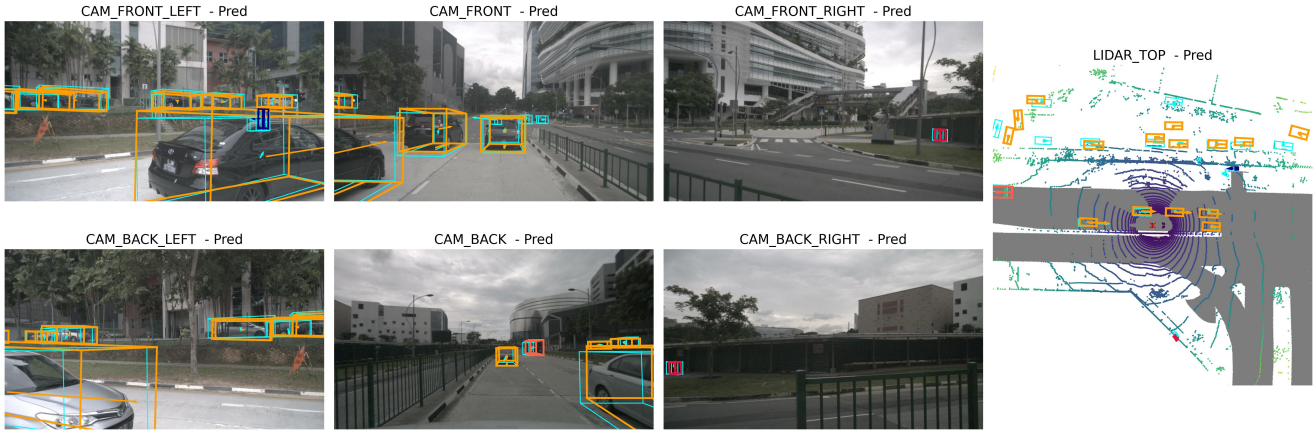

Figure 4. Visualization of final predictions and ground truth boxes on nuScenes val set. Ground truth boxes are coloured in light green. Boxes coloured in orange, red and blue correspond to vehicles, bicycles and pedestrians, respectively.

Box	Head	Delta	Init	Others	NDS↑	mAP↑
✓			Random	-	0.310	0.249
	✓		Random	-	0.341	0.258
✓	✓		Random	-	0.409	0.333
✓	✓	✓	Fixed	-	0.414	0.329
✓	✓	✓	Random	-	**0.428**	0.337
✓	✓	✓	Random	Self-attn	0.427	**0.338**

Table 4. Ablation on key modules. "Box" denotes the two-stage query boxes module. "Head" denotes the sparse interaction head. "Delta" denotes the box adjustment module. "Init" denotes the initialization method for query boxes. "Self-attn" means we add a self-attention module to the feature sampling module.

Modules	AMOTA↑	AMOTP↓	RECALL↑
DE	0.277	1.519	0.506
PR	0.405	1.361	0.508
PR + R	0.436	1.287	0.539
PR + H	**0.439**	**1.280**	**0.545**

Table 5. Ablation of the tracking module. Deterministic method is our adaption of AB3DMOT [29], The other lines shows utilization of different features in data association process. "DE" denotes deterministic matching method. "PR" denotes probabilistic matching method. "R", and "H" denotes RoI features and hybrid features, respectively.

The number of query boxes and features is related to GPU memory cost and inference speed. Number of objects determines the output of NMS-free box coder. Number of stages refers to how many times the query boxes are refined.

Table 6 shows that reducing the number of queries has only a slight impact on accuracy ($< 3\%$), while reducing the number of candidate objects leads to a greater loss of accuracy. Additionally, the first three stages in the cascade structure contribute significantly to improving accu-

Backbone	Queries	Objects	Stages	NDS↑	mAP↑
R-101	900	300	6	0.428	0.337
R-101	500	300	6	0.408	0.332
R-101	300	300	6	0.418	0.327
V2-99	900	300	6	**0.475**	**0.396**
V2-99	500	300	6	0.474	0.378
R-101	300	300	6	**0.418**	**0.327**
R-101	300	200	6	0.410	0.324
R-101	900	300	6	**0.428**	**0.337**
R-101	900	300	3	0.421	0.335
R-101	900	300	1	0.385	0.297

Table 6. Ablation on the number of queries, candidate objects and stages.

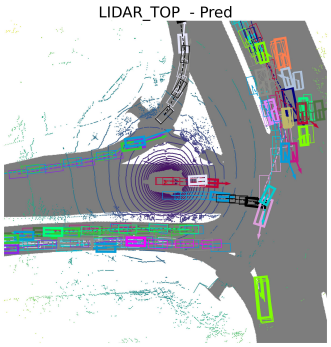

Figure 5. Visualization of tracking results on nuScenes val set. We visualize the tracked objects in the past five key frames in the same scene. Boxes in different colors refer to different tracked instances.

racy, while later stages provide diminishing returns. This suggests that the refinement process efficiently converges by the third stage. Our ablation study demonstrates that our sparse feature sampling module can capture regions of interest using only local features in a lightweight manner.

Ablation on tracking module. As shown in Table. 5, for camera-only tracking, probabilistic association greatly outperforms deterministic counterparts. We further demonstrate that as RoI features and query features represent kinds of affinity cues for appearance and classification respectively, these latent features facilitate the matching process and improve tracking precision.

4.6. Visualization

Fig. 4 visualizes final detection results in the camera front view and bird's-eye view on the validation set with ground truth annotation. Overall, as shown in the bird's-eye view, the predicted boxes are close to the ground truth ones. The bus detected as a whole both in front and left-front camera illustrates the effectiveness of cross-view fusion in overlapped regions. Small objects (e.g. pedestrians in the front-left camera view) are also detected precisely. These results indicate a satisfactory performance of SRCN3D and related modules. More visualizations of final results and intermediate-stage query boxes are available in the supplementary materials.

Fig. 5 presents a BEV example of tracking on nuScenes validation set. We visualize past five key frames of unique objects in a crowded intersection to demonstrate satisfactory tracking accuracy and continuity.

Visualization also exposes current limitations of SRCN3D. In areas where features of targeted objects are dense, the predicted boxes overlap with each other, which is unreasonable for real-world objects. It shows that there remains a few duplicates in predicted boxes.

4.7. Discussion

Sparse R-CNN 3D provides the first attempt for box-wise sampling and refinement approach to conduct 3D object detection and downstream multi-object tracking. The RoI features are commonly used in two-stage object detection pipelines, serving as a downstream refinement of region proposal network (RPN). In contrast to grid-sampled pixel-level feature sampling, RoI features offer a comprehensive perspective of the region, making them better suited for tasks such as classification, orientation, and velocity estimation. In contrast, RoI features are not as effective as grid sampling in object localization tasks in 3D space, which leads to larger errors in object center and scale estimation.

5. Conclusion

This paper proposes a novel innovative architecture, SRCN3D, aiming at detecting and tracking objects of interest. It possesses the traits of sparse queries, sparse attention and sparse prediction, and is able to efficiently extract and fuse cross-view features. Our insight is that box-feature twin-track queries and cascade-style refinement process with only local RoI attention enable 3D object detection and cross-view fusion. We hope that this architecture can serve as a foundation for fully-sparse surround-view 3D object detection. In the future, the authors will investigate deeper in combining segmentation and temporal information to enhance the accuracy and robustness of SRCN3D.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grants U22A20104, and Beijing Municipal Science and Technology Commission (Grant No.Z221100008122011).

References

[1] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *J. Image*

Video Process., 2008, Jan. 2008. 5

[2] 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

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2, 5

[4] Mohamed Chaabane, Peter Zhang, J Ross Beveridge, and Stephen O'Hara. Deft: Detection embeddings for tracking. *arXiv preprint arXiv:2102.02267*, 2021. 7

[5] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2781–2790, 2022. 6

[6] Shaoyu Chen, , Xinggang Wang, Tianheng Cheng, Qian Zhang, Chang Huang, and Wenyu Liu. Polar parametrization for vision-based surround-view 3d detection. *arXiv:2206.10965*, 2022. 7

[7] Hsu-Kuang Chiu, Jie Li, Rareş Ambruş, and Jeannette Bohg. Probabilistic 3d multi-modal, multi-object tracking for autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14227–14233, 2021. 5

[8] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019. 5, 6

[9] Angel García-Fernández, Jason Williams, Karl Granström, and Lennart Svensson. Poisson multi-bernoulli mixture filter: direct derivation and implementation. *IEEE Transactions on Aerospace and Electronic Systems*, PP, 03 2017. 3

[10] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. *arXiv preprint arXiv:2208.01582*, 2022. 7

[11] 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. 3, 5

[12] 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*, 2022. 2, 3, 7

[13] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving, 2023. 7

[14] 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. 6

[15] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020. 5

[16] 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. 2, 5, 6

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 3

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4, 5

[19] 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, 5, 6

[20] 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. 2

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 5

[22] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 5

[23] 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. 2, 5, 6

[24] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proceedings of the European Conference on Computer Vision*, 2020. 1

[25] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021. 1, 2

[26] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning*, pages 1475–1485. PMLR, 2022. 2, 5

[27] 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. 2, 5, 6

[28] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. DETR3d: 3d object detection from multi-view images via 3d-to-2d queries. In *5th Annual Conference on Robot Learning*, 2021. 1, 2, 5, 6

[29] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics. *arXiv preprint arXiv:2008.08063*, 2020. 3, 7

[30] 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. 6

[31] Renrui Zhang, Han Qiu, Tai Wang, Xuanzhuo Xu, Ziyu Guo, Yu Qiao, Peng Gao, and Hongsheng Li. Monodetr: Depth-aware transformer for monocular 3d object detection. *arXiv preprint arXiv:2203.13310*, 2022. 2

[32] 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. 3, 7

[33] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020. 3, 7

[34] 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

[35] 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. 2