SpatialDETR: Robust Scalable Transformer-Based 3D Object Detection from Multi-View Camera Images with Global Cross-Sensor Attention

Simon Doll[1,3], Richard Schulz[1], Lukas Schneider[1], Viviane Benzin[1], Markus Enzweiler[2], and Hendrik P.A. Lensch[3]

[1] Mercedes-Benz, simon.doll@mercedes-benz.com
[2] Esslingen University of Applied Sciences,
[3] University of Tübingen simon.doll@student.uni-tuebingen.de

Abstract. Based on the key idea of DETR this paper introduces an object-centric 3D object detection framework that operates on a limited number of 3D object queries instead of dense bounding box proposals followed by non-maximum suppression. After image feature extraction a decoder-only transformer architecture is trained on a set-based loss. SpatialDETR infers the classification and bounding box estimates based on attention both spatially within each image and across the different views. To fuse the multi-view information in the attention block we introduce a novel geometric positional encoding that incorporates the view ray geometry to explicitly consider the extrinsic and intrinsic camera setup. This way, the spatially-aware cross-view attention exploits arbitrary receptive fields to integrate cross-sensor data and therefore global context. Extensive experiments on the nuScenes benchmark demonstrate the potential of global attention and result in state-of-the-art performance. Code available at https://github.com/cgtuebingen/SpatialDETR.

Keywords: 3D object detection, cross-sensor attention, autonomous driving

1 Introduction

To achieve the goal of autonomous driving it is necessary to perform 3D scene understanding and to detect objects such as other cars or pedestrians. The required 3D object detection task is typically either performed using a LiDAR sensor [10], [30], multi-view cameras [27], [29] or by using multiple sensor modalities [31], [25]. While LiDAR-based methods have the advantage of precise information about the 3D structure of objects, camera-only approaches come with higher sensor-refresh rates and scale also to low-cost autonomous vehicles.

The problem of multi-view 3D object detection solely from camera images is challenging due to the fact that the images only contain a 2D projection of the 3D scene. Furthermore, the position of the camera sensors needs to be taken into

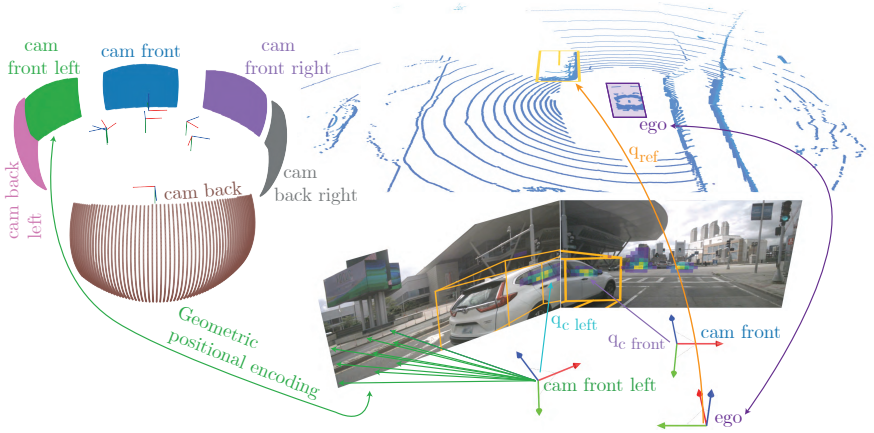


Fig. 1. Visualization of our spatially-aware global cross-sensor attention. Fusing the 3D information from the individual images is supported by a positional encoding (left) that incorporates the global 3D view direction of each pixel

account, since some cameras might have overlapping field of views. FCOS3D [27] tackles this issue by extending concepts from the 2D object detection to the 3D domain. This typically results in the need for post-processing steps to filter out duplicate detections in overlapping regions since the 3D structure and the positioning of the sensors is not taken into account. Pseudo-LiDAR-based methods [28] solve this by converting the images to a 3D point cloud [16] and achieve state-of-the-art performance on the nuScenes benchmark [1]. Using multi-head dot-product attention as introduced in [24], DETR [2] proposed a new object-centric approach to 2D object detection. Recently, DETR3D [29] pioneered in extending this concept to the multi-view 3D object detection domain. But in contrast to DETR where global attention between a query and each feature patch is used to select pixels corresponding to an object, DETR3D only uses pixels that correspond to the query center and therefore supposably misses feature patches corresponding to a large object in the scene.

To allow for global, cross-sensor attention and to therefore enable an object query to select all feature patches that correspond to an object in the scene the DETR3D formulation is not sufficient. Global, cross-sensor attention requires a way to compare an object query with keys that correspond to feature patches in the different camera images.

Motivated by the findings in [16] that the success of pseudo-LiDAR methods results from the explicit modelling of the sensor positions, we propose a new geometric positional encoding to exploit the spatial and geometric structure of object detection in 3D. We furthermore introduce a spatially-aware cross-attention block to account for different coordinate spaces and to model query-key similarity in a sensor-specific fashion. The resulting combination enables global cross-sensor attention and an implicit conversion from 2D image patches

to a unified, latent 3D representation. Our approach called SpatialDETR is naturally scalable to other sensor modalities such as LiDAR or Radar since the direction vector formulation of our positional encoding models the way how sensors of all modalities capture data. We hope to encourage future research to focus on multi-modal 3D object detection by introducing a unified spatially-aware representation for global cross-sensor attention. See Fig. 1 for an overview.

Summarizing, our main contributions are:

- Geometric positional encoding
- Cross-sensor global attention for arbitrary sensor setups
- Explicit modelling of sensor calibrations

2 Related Work

2.1 Transformer-Based Object Detection

Modern 2D object detectors often predict a high amount of *oriented bounding boxes* (OBB)s [20], [12], [23] by using anchor-boxes or a dense prediction scheme. As a result, such methods rely on post-processing steps such as non-maximum suppression to filter out highly overlapping OBB proposals. DETR [2] mitigates this issue by viewing object detection as a set-prediction problem. Using a transformer encoder-decoder architecture and learnable object queries as input to the decoder results in a small number of object proposals which allow for a set-based loss via bipartite matching with the ground truth. To improve the slow convergence of DETR various methods have been proposed such as introducing sparsity to keys [33] or queries [21] or by using an encoder-only version of DETR [22].

To perform object detection in 3D from monocular camera images, two-stage methods convert the images to a pseudo-LiDAR point cloud [28], [8] using dense depth-estimation. Single stage methods either leverage an object detection network that was pretrained on a large-scale monocular depth estimation task [18] or multi-task learning [27].

DETR3D [29] proposes a DETR-based decoder-only approach that is capable of 3D object detection via 3D to 2D queries without dense depth prediction. For each object query the object center is extracted via a *feed-forward network* (FFN) and projected to the different image planes to extract multi-view features from the backbone. Similar to DeformableDETR, the attention weights are not image-feature dependent via query-key similarity as in the original attention formulation [24] but are instead directly inferred from the object queries with a FFN. As a result, the attended feature patches correspond to the object centers only, which requires deformable convolutions in the backbone to allow for larger receptive fields since no global attention is performed. In contrast to this, we propose a geometric positional encoding that explicitly models the sensor calibration and allows to incorporate global context across multiple sensors via global cross-attention.

2.2 Prior-Guided Attention

As stated in [33] attention-based methods like DETR require long training schedules since they do not introduce inductive biases like convolutional architectures and learn a fully flexible receptive field over a sequence of inputs [24]. Deformable DETR and DETR3D mitigate this issue by attending to a small number of patches only or by even using only the object center. Spatial-Gaussian priors are introduced in [6] to increase the attention weights of feature patches close to the objects and to therefore account for the local nature of objects. In TransFusion [15] the epipolar-line and epipolar field are used in a dual camera setup as geometric loss function to guide the attention weights during training. In our work we avoid hard constraints, guide attention through explicit modeling of the scene geometry and model geometric dependencies between an object query and a feature patch directly in the cross-attention blocks using sensor-relative query projections. Due to the geometric interpretation of our encoding it is still possible to introduce task or sensor-set specific biases, e.g. by only using queries that lie in front of the image plane to speed up the convergence.

2.3 Positional Encodings

Since transformers are invariant to input permutations, positional encodings as introduced in [24] and [5] are used to incorporate positional information to the feature patches. Typically, either learned or fixed encodings are used, whilst the latter lead to a slightly higher performance in DETR [2]. Fixed encodings consisting of alternating sine and cosine functions [2] might have the benefit of better extrapolation behavior. Although those encodings are well-suited for grid-based data like images the encoding of multi-view data from cameras mounted at arbitrary positions results in a complex graph of pixel directions. Additionally, small calibration changes, e.g. due to different vehicles in a fleet of cars might result in changed pixel directions. Our proposed encoding models pixel directions as a function of the sensor extrinsics and is scalable to varying sensor-modalities and arbitrary sensor calibrations.

3 Method

3.1 Overall Architecture

An overview of the proposed architecture is presented in Fig. 2. The input consists of multi-view, RGB camera images which are utilized to predict a set of 3D bounding boxes in the scene.

A shared backbone for all cameras is applied to compress the images to a latent representation. Following the design of DETR and DETR3D a transformer decoder is used to generate OBB proposals. Our proposed geometric positional encoding is used to incorporate the extrinsics and intrinsics of the camera setup and scene geometry into the sequence of input features supporting global attention across all cameras. The so-called *object queries* are learnable vectors that are

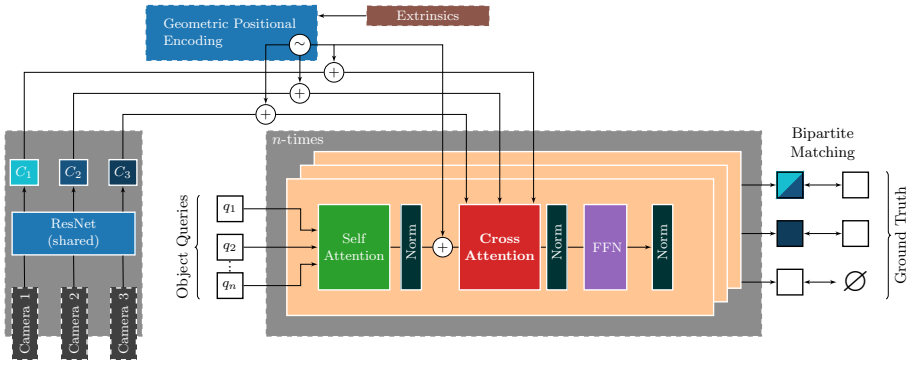


Fig. 2. SpatialDETR architecture. The camera images are encoded using a shared backbone. Afterwards object proposals are iteratively refined with self- and cross-attention in n decoder layers. The newly proposed geometric positional encoding is used in each spatially-aware cross-attention block for query-key similarity

refined iteratively using a stack of decoder layers. Each decoder layer consists of a self-attention block between the object queries and a cross-attention block between object queries and image feature maps. In contrast to DETR our decoder-only approach scales linearly with the image resolution since no self-attention between feature patches is performed as motivated in [9]. As in DETR3D [29], the final bounding box predictions are generated by a FFN converting each latent object query to an OBB described as $\left(x, y, z, w, h, l, \theta_{cos}, \theta_{sin}, v_x, v_y\right)$ where $\left(x, y, z\right)$ correspond to the object center, $\left(w, h, l\right)$ to the extend (width, length and height) of the bounding box, v_x and v_y to the object velocity and $\theta = \arctan\left(\frac{\theta_{sin}}{\theta_{cos}}\right)$ to the object heading. Following [29], the bounding box center coordinates are predicted in normalized coordinates using a sigmoid function and are afterwards scaled to the desired object detection range to increase the numeric stability. During training each decoder layer computes bounding box predictions to improve the gradient flow, during inference only the output of the last decoder layer is used.

After each decoder layer the output object centers $\mathbf{o}_c^{n+1}$ are interpreted as relative offsets to the object centers of the last decoder layer before the sigmoid which results in the following update of a query center $\mathbf{q}_c^{n+1}$ in decoder layer $n+1$ as introduced in [29]

$$\mathbf{q}_c^{n+1} = \text{sigmoid}\left(\text{sigmoid}^{-1}(\mathbf{q}_c^n) + \mathbf{o}_c^{n+1}\right) \tag{1}$$

3.2 Revisiting Attention in DETR

Vaswani et al. introduced the concept of dot-product attention in [24] as

$$Attn(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_k}}) \cdot \mathbf{V} \tag{2}$$

For each value $\mathbf{v} \in \mathbf{V}$, an attention weight is computed as the scaled dot product between a query $\mathbf{q} \in \mathbf{Q}$ and the corresponding key vector $\mathbf{k}$ (see [24]). This query-key similarity is responsible for selecting the relevant values in the input. Afterwards, the resulting weighted sum of values is the output of an attention head. In the case of cross-attention for object detection the values and keys correspond to linearly projected feature patches while the queries correspond to objects in the scene. To adapt this concept to multi-sensor cross-attention, we need to solve two main challenges in our framework:

- **Position information:** The spatial location of each feature patch from all cameras needs to be incorporated, e.g. by using a positional encoding.
- **Query-key similarity:** The query-key similarity via the dot-product in Eq. (2) needs to take the different coordinate spaces into account, since object queries are defined with respect to a reference system while feature patches encode camera-relative information.

As a solution we propose a novel geometric positional encoding in Section 3.3 to model spatial dependencies between feature pixels. Furthermore, we introduce a spatially-aware cross-attention block that resolves the issues related with different coordinate spaces in Section 3.4.

3.3 Geometric Positional Encoding

As shown in DETR [2] global spatial attention between a query that represents an object and feature patches from the images is a simple yet effective way to allow for flexible receptive fields and therefore global context. The selected image patches are used to update the query to allow for a precise localization and classification of the object. Furthermore, object detection based on attention removes the need for post-processing steps like non-maximum suppression or prior knowledge such as anchor boxes while still achieving state of the art performance since the learnable object queries specialize to learned priors during training [2].

To utilize dot-product attention as defined in Eq. (2) the keys need a positional encoding to incorporate position information of an image patch, since the general attention mechanism is permutation invariant with respect to the input sequence [24], [2]. Modelling this with a 1D or 2D fixed *sine-cosine* encoding [5] or a learnable positional encoding [2] loses the advantage of explicitly modelling the sensor extrinsics as discussed beforehand. Additionally, in contrast to encodings for a single image in which the pixels follow a grid-like structure, the introduction of multi-view images results in a complex graph of pixel directions (see Fig. 1). A simple 1D or 2D enumeration of pixel positions as done in [5] or [2] is therefore not applicable. Taking the positioning of the different sensors with respect to a reference frame into account is required to preserve robustness and scalability to a fleet of cars with varying extrinsic calibrations or even different sensor sets. Our approach is in contrast to DETR3D where the 3D query centers are projected into the individual images via the extrinsic and intrinsic calibration, followed by a simple lookup of the image feature at that position without any

spatial attention. The subsequent cross-attention across the different views does therefore not need image patch information. As a result, a query in DETR3D only aggregates features corresponding to object centers across multiple-feature levels and different views without a flexible receptive field and global context or key to query similarity.

Having the possibility to use global attention is required to improve the detection of large objects such as trucks or busses and is the key step towards a unified representation for multi-task learning since tasks like behavior prediction or planning rely on global context as shown at the TESLA-AI-Day[4] and in [19].

To allow for robust global attention we propose a sensor-relative, spatially-aware encoding. From the camera intrinsic parameters for a camera c which are given from the dataset we can directly compute normalized direction vectors $\mathbf{d}_c^p$ per pixel p. This view-ray vectors are then element-wise added to the latent key vectors via a fully learnable network *dir2latent* that maps the three dimensional inputs to the latent feature dimension which is shared for all key and query vectors (see Fig 3).

This results in the following update for the latent key vectors:

$$\hat{\mathbf{k}}_c^p = \mathbf{k}_c^p + \mathit{dir2latent}(\mathbf{d}_c^p) \tag{3}$$

Therefore, no hyper parameter for scaling the encoding, to control the amount of shift that is applied to the values, is required and the model learns how to merge semantic and geometric information during training.

It is noteworthy that this formulation results in sensor-relative encodings, since the direction vectors are defined with respect to the corresponding camera frames. A transformation of the key encodings to reference coordinates is not applicable without an explicit depth estimate per feature pixel due to the translation component. Similar to fixed sine-cosine encodings our geometric positional encoding can model relative offsets with a linear operator [24] by using a three-dimensional rotation matrix as operator.

3.4 Spatially-Aware Attention

To use the sensor-relative encodings and to perform similarity matching within the same coordinate space while allowing to explicitly model the sensor extrinsics, the queries in a cross-attention block need to be camera specific as they are used in the dot-product with sensor specific keys (see Eq. (2)).

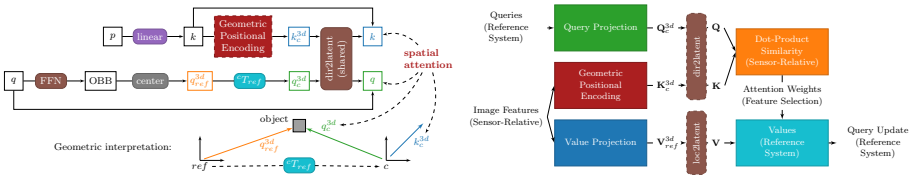
To tackle this each object query $\mathbf{q}$ is projected to all camera frames using

$$\mathbf{q}_{\text{ref}}^{3d} = \text{center}(\mathit{query2box}(\mathbf{q})) \tag{4}$$

$$\mathbf{q}_c^{3d} = {}^{c}\mathbf{T}_{\text{ref}} \cdot \mathbf{q}_{\text{ref}}^{3d} \tag{5}$$

$$\mathbf{q}_c = \mathbf{q} + \mathit{dir2latent}\left(\mathbf{q}_c^{3d} \,/\, ||\mathbf{q}_c^{3d}||_2\right) \tag{6}$$

[4] TESLA-AI-Day: `https://www.youtube.com/watch?v=j0z4FweCy4M`



(a) Spatially-aware query-key similarity (b) Similarity computation

Fig. 3. Visualization of the proposed spatial attention module. Each query $\mathbf{q}$ gets a sensor-relative update and each key $\mathbf{k}$ a geometric update. The resulting spatially-aware latent vectors can then be compared using a dot-product to perform spatial attention

The FFN *query2box* projects the latent query vector to an OBB as defined in Section 3.1. The query center is then projected to all camera frames. Transformations ${}^{b}\mathbf{T}_{a}$ from one frame a to another frame b are modelled via homogenous matrices

$$ {}^{b}\mathbf{T}_{a} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}. \tag{7} $$

A visualization of the camera-relative queries and their geometric similarity with keys from a camera image is given in Fig. 3.

As a result, the queries get a camera-relative geometric update that enforces a latent encoding which is similar to the geometric-aware keys by sharing the network for direction to latent projection. The resulting queries $\mathbf{q}_c$ and keys $\hat{\mathbf{k}}_c^p$ are then used for the attention weight computation.

In contrast to this sensor-relative space, the resulting sum of weighted values is used to perform a query update in reference coordinates. This is due to the fact, that patches from different cameras may contribute to the same object query. This results in the need for values that are encoded with respect to the reference frame to decouple sensor extrinsics from the feature encodings. Analogously to the query projection this can be done by computing a latent update to transform a value to the reference system. This is achieved by computing an explicit depth estimate v_{depth} for each latent value vector $\mathbf{v}_c^p$ utilizing a FFN:

$$ v_{\text{depth}} = \mathit{FFN}(\mathbf{v}_c^p) \tag{8} $$

$$ \mathbf{v}_c^{3d} = v_{depth} \cdot \mathbf{d}_c^p \tag{9} $$

$$ \mathbf{v}_{\text{ref}}^{3d} = {}^{\text{ref}}\mathbf{T}_{\text{c}} \cdot \mathbf{v}_{\text{c}}^{3d} \tag{10} $$

$$ \mathbf{v}_{\text{ref}}^{p} = \mathbf{v}_c^p + \mathit{loc2latent}(\mathbf{v}_{\text{ref}}^{3d}). \tag{11} $$

The direction vector $\mathbf{d}_c^p$ is scaled according to the computed factor which results in a sensor-relative depth estimate. Finally, the latent value update is applied using the FFN *loc2latent* to convert the 3D global position $\mathbf{v}_{\text{ref}}^{3d}$ to a latent vector, resulting in the spatially-aware latent value vector $\mathbf{v}_{\text{ref}}^{p}$.

This formulation also directly integrates multi-task learning into the framework by introducing a depth loss for all values with an attention weight greater

than zero if ground truth depth data is available. In comparison to a dense depth prediction scheme, the proposed approach produces depth gradients only for patches that contributed to an object. As a result, depth estimates for regions without well-defined depth like the sky are not contributing to the loss and the depth estimates are focused on the objects in the scene.

As a consequence, the proposed cross-attention block does perform the attention weight computation in a sensor specific manner while the resulting query updates from weighted values are performed in reference coordinates. We expect this formulation to help the model to decouple the sensor positioning from the feature encoding and to scale to small calibration changes or online calibration.

4 Experiments

We evaluate the performance of SpatialDETR on the well-established nuScenes benchmark in the setting of camera-only single-shot 3D object detection. In addition, we provide qualitative results and several ablation studies to evaluate the effects of the proposed encoding, the spatial attention block and decoder layer configuration.

4.1 Experimental Setup

Dataset All experiments are performed on the nuScenes dataset [1] for autonomous driving. Data was collected using cars that feature a setup of six cameras mounted on the roof of the car to capture images in different directions as well as other sensors. No radar or LiDAR data is used during training or evaluation. The dataset consists of 1000 scenes with a length of 20s each, containing annotations of 23 classes from which 10 are used to compute benchmark metrics [17]. Training is performed utilizing the officially defined `train` split. Since ground-truth annotations are only provided for the training and validation set, all ablation experiments are performed on the validation set.

Metrics For evaluation we report metrics according to the official nuScenes standard [1] which include the mean Average Precision (mAP), true-positive metrics such as Average Translation Error (ATE), Average Scale Error (ASE) and Average Orientation Error (AOE), Average Velocity Error (AVE), Average Attribute Error (AAE) and the summarizing nuScenes Detection Score (NDS). For a detailed metric definition, the mAP computation and class-specific detection ranges we refer the reader to [1].

Model Configuration All model configurations use a shared ResNet-101 [7] backbone which is initialized from a FCOS3D [27] checkpoint as in [4]. The attached SpatialDETR head uses the output of the last backbone stage as input. As done in DETR3D [29] a stack of $d = 6$ decoder layers is then applied to produce the object bounding box proposals for $q = 900$ object queries. We

closely follow the decoder configuration proposed in DETR3D [4] to increase the comparability. Each decoder layer uses a latent dimension $d_e = 256$ for the attention computation which is then split to $h = 8$ heads of parallel multi-head-attention modules with a latent dimension $d_h = \frac{d_e}{h} = 32$ in sequential query-to-query self-attention and query-to-patch cross-attention blocks. All FFNs used in the spatial attention blocks utilize two hidden layers consisting of a linear projection of dimensionality d_e, Layer-norm and a ReLU activation function. The two sub-networks that predict the OBB and a corresponding class label are chosen as proposed in [29]. The benchmark configurations are trained with shared decoder weights and attend only to camera images in which the object center is visible, see Section 4.3 for details.

Training Parameters As proposed in [4] we train all models for 24 epochs, a total batch size of 8 on 8 NVIDIA A100 GPUs with 40GB RAM and a backbone in which the first block is frozen. The ablation models are trained with a batch size of 4 on 4 NVIDIA V100 GPUs with 16GB RAM and a fully frozen backbone to reduce the memory footprint. All models are trained using the settings proposed in DETR3D [4], consisting of an AdamW [14] optimizer and a cosine-annealing lr-schedule with an initial learning rate of $2e^{-4}$. We did not perform hyperparameter tuning for the used learning rate schedule, optimizer, batch size or used backbone and use no test-time augmentation or stochastic weight averaging. Our implementation is based on DETR3D [4] and MMDetection3D [3].

4.2 Comparison to Existing Works

We compare SpatialDETR to previous state-of-the-art methods for multi-view 3D object-detection. While FCOS3D [27], PGD [26] and BEVDet [8] follow a pseudo-LiDAR approach, DETR3D [29] and SpatialDETR use a transformer-based object-centric paradigm. As shown in Table 1 our method achieves the highest performance in terms of mean Average Precision and performs equally with DETR3D in the nuScenes detection score using the same training configuration as DETR3D.

It is noteworthy that the transformer-based methods gain a large performance increase when using a feature extractor that was initialized from a pseudo-LiDAR method. We suspect this to result from the dense-depth estimation and the depth-based loss in pseudo-LiDAR methods which might result in a faster convergence and spatially-aware backbone features while transformer-based methods, have a slower convergence behavior due to the uniformly distributed attention in early training stages [33]. Our model does not use multi-scale feature maps with a FPN [11] via multi-scale attention as proposed in [6] which might be the reason for the slightly less precise bounding boxes as compared to DETR3D.

Without any bells and whistles like test-time augmentation, model ensembling, finetuning or complex data augmentation as in BEVDet [8], SpatialDETR ranks within the Top-3 on the official nuScenes benchmark [17] and clearly outperforms DETR3D (see Table 2). As shown in Table 3 our global attention

Table 1. Comparison of state-of-the-art methods with different configurations on the nuScenes `val` set. PT flags models that use a pretrained backbone initialized from a FCOS3D checkpoint, MS indicates methods that utilize multi-scale features. ‡ marks methods using class-balanced grouping and sampling (*CBGS*) [32], + stands for methods using a Swin-Transformer-Tiny backbone [13], ¶ for results reported in [8] and † for results reported in [29]

Name	PT	MS	mAP↑	ATE↓	ASE↓	AOE↓	AVE↓	AAE↓	NDS↑
FCOS3D[27]†	✗	✓	0.299	0.785	0.268	0.557	1.396	**0.154**	0.373
FCOS3D[27]†	✓	✓	0.321	0.746	0.265	0.503	1.351	0.160	0.393
PGD [26]¶	✗	✓	0.335	0.732	**0.263**	0.423	1.285	0.172	0.409
BEVDET [8]‡	✗	✓	0.317	0.704	0.273	0.531	0.940	0.250	0.389
BEVDET [8]+‡	✗	✓	0.349	**0.637**	0.269	0.490	0.914	0.268	0.417
DETR3D [29]	✗	✓	0.303	0.860	0.278	0.437	0.967	0.235	0.374
DETR3D [29]	✓	✓	0.346	0.773	0.268	**0.383**	**0.842**	0.216	**0.425**
SpatialDETR	✗	✗	0.303	0.849	0.282	0.522	0.941	0.229	0.369
SpatialDETR	✓	✗	**0.351**	0.772	0.274	0.395	0.847	0.217	**0.425**

approach increases the performance by up to 3% on large objects such as trucks and busses in comparison to DETR3D. This supports the assumption that our attention allows for global context and arbitrary receptive fields that can capture large objects across the different cameras.

Since we do not use multi-scale feature maps, the performance on small objects such as pedestrians is slightly reduced due to the low-resolution backbone feature map which is in accordance to the findings in DETR [2]. As for the 2D case this can be tackled by using multi-scale attention as proposed in [33] or [6]. Additionally, our global attention formulation does not significantly increase the runtime since the entire global attention computation and feature projections can be computed in parallel.

4.3 Ablations and Analysis

Qualitative results for the performance of SpatialDETR on the nuScenes `val` set are shown in Fig. 4. The visualized attention maps in Fig. 5 show how the model

Table 2. Comparison of state-of-the-art methods with different configurations on the nuScenes `test` set. Our model was trained with CBGS [32] on the training and validation set as proposed by DETR3D [4]

Name	mAP↑	ATE↓	ASE↓	AOE↓	AVE↓	AAE↓	NDS↑
BEVDet [8]	**0.424**	**0.524**	**0.242**	**0.373**	0.950	0.148	**0.488**
DETR3D [29]	0.412	0.641	0.255	0.394	**0.845**	0.133	0.479
SpatialDETR	**0.424**	0.613	0.253	0.402	0.857	**0.131**	0.486

Table 3. Comparison of transformer-based methods in terms of average precision for different classes on the `val` set and runtime on a single NVIDIA V100 GPU. Both methods use a ResNet101 initialized from a FCOS3D checkpoint. Note how our spatially-aware attention improves the localization error particularly for extended objects

Method	truck↑	bus↑	trailer↑	pedestrian↑	traffic cone↑	FPS↑
DETR3D	0.286	0.347	0.167	**0.424**	**0.529**	**2.5**
SpatialDETR	**0.302**	**0.378**	**0.175**	0.418	0.514	2.4

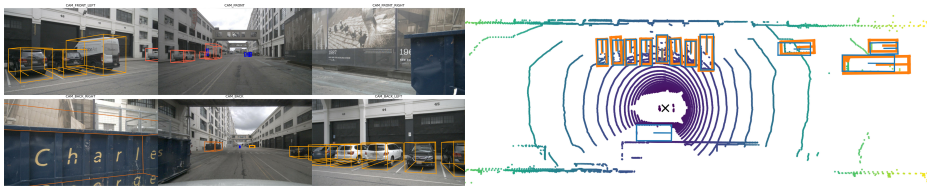
uses the global attention mechanism to focus on feature patches that belong to an object even across image borders as shown for query q^{93}.

The effect of different configurations of the proposed positional encoding and the proposed spatial-attention is analyzed in Table 4. Here, a configura-

Table 4. Effect of the different components of SpatialDETR. $P(Q)$ indicates the projection of queries in sensor-relative coordinates, $P(V)$ the projection of values to global coordinates. Models with $C(Q)$ only attend to feature patches of an image if the query center is visible within that image

$P(Q)$	$P(V)$	$C(Q)$	mAP↑	ATE↓	ASE↓	AOE↓	AVE↓	AAE↓	NDS↑
✓	✓	✗	0.313	0.850	0.274	0.494	0.814	0.213	0.392
✗	✗	✗	0.248	0.993	0.284	0.506	**0.771**	0.202	0.348
✓	✗	✗	0.317	**0.824**	0.276	0.470	0.797	0.200	**0.402**
✓	✗	✓	0.318	0.835	0.277	**0.460**	0.837	0.210	0.397
✓	✓	✓	**0.319**	**0.824**	**0.272**	0.502	0.813	**0.192**	0.399

tion without sensor-specific queries reduces the baseline performance where the queries use a sensor specific, geometric update $P(Q)$ as defined in Eq. (6). This is likely due to the fact that the translation of the cameras with respect to the



(a) Input images with GT-OBBs (b) Predictions plotted in BEV

Fig. 4. Qualitative detection results. a) input images and ground truth OBBs, b) detection results in BEV (blue) and ground truth (orange)

reference system is not accounted for in the global formulation. Introducing the geometric constraint that the predicted object center of an object query has to be in front of the image plane increases the performance which proves the effectiveness of our proposed geometric encoding since it allows to model various task-specific geometric constraints. A more complex computation that does not only take the object center but instead the full shape estimate of the bounding box into account might improve the performance even further.

Interpreting values directly as global features without the proposed explicit value projection update, increases the performance compared to the baseline but might result in a coupling between backbone features and sensor calibration. Since we directly predict a depth value per feature patch with a FFN this might cause numeric instabilities and could therefore require a customized scaling dependent on the desired object detection range. Furthermore, depth-supervision either by ground-truth depth maps or LiDAR hits might be needed to increase the performance. Table 5 shows the effect of different decoder settings. Changing the number of decoder layers as well as the number of used object queries has a similar effect to DETR [2] and DETR3D [4]. Decreasing the amount of object queries or decoder layers (see Table 5) decreases the performance, the performance saturates close to the default values of six decoder layers and 900 object queries as proposed in [29]. Sharing the weights between the decoder layers and therefore treating the full decoder like a RNN with the same input for each layer and a cross-attention block as proposed in [9] results in a performance gain, reduction of trainable parameters and more precise bounding box estimates.

5 Conclusion

This paper presents SpatialDETR, a novel spatially and geometrically motivated transformer-based 3D object detection framework for multi-view data. It explicitly models the sensor calibration and features linear complexity with respect to the input images size due to a decoder-only attention architecture. The introduced geometric positional encoding and spatially-aware attention block enable us to use the global, cross-sensor context for arbitrary sensor positions and

Table 5. Analysis of different decoder configurations. Models flagged as shared, share weights of the decoder layer 2-6. The first row indicates our baseline configuration

Queries	Layers	Shared	mAP↑	ATE↓	ASE↓	AOE↓	AVE↓	AAE↓	NDS↑
900	6	✗	0.313	0.850	0.274	0.494	0.814	0.213	0.392
600	6	✗	0.308	0.846	0.279	0.521	0.809	0.209	0.388
1200	6	✗	0.313	0.841	0.274	0.501	**0.798**	**0.207**	0.394
900	2	✗	0.308	0.846	0.278	0.493	0.899	0.230	0.379
900	10	✗	0.311	0.838	0.277	0.483	0.810	0.223	0.392
900	6	✓	**0.314**	**0.823**	**0.273**	**0.472**	0.822	0.224	**0.395**

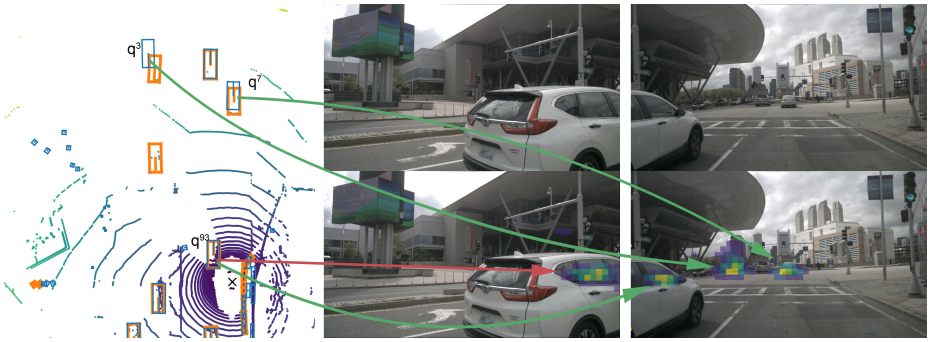


Fig. 5. Visualization of the spatially-aware global cross-sensor attention. Left: predictions (blue) and ground truth (orange). For three queries ($\mathbf{q}^3, \mathbf{q}^7, \mathbf{q}^{93}$) we jointly visualize the query-key attention maps for the sensor-relative queries (bottom images). Green arrows indicate queries relative to the front camera, red arrows queries relative to the front-left camera

achieve state-of-the-art performance while significantly improving the performance for large objects without task-specific, hand-crafted constraints or post-processing steps. Similar to DETR, the detection of small objects is challenging and might be tackled in future work as done in the 2D case already. To prove the effectiveness of spatially-aware features that are decoupled from the sensor calibration, new data sets that model a fleet of cars, contain varying sensor calibrations and ground truth depth maps might be required. We are looking forward to apply the presented approach to multi-modal sensor data and to investigate its unified latent cross-sensor representation with multi-task learning.

Acknowledgement The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Climate Action within the project "KI Delta Learning" (Förderkennzeichen 19A19013A). The authors would like to thank the consortium for the successful cooperation."

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11621–11631 (2020)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision (ECCV). pp. 213–229 (2020)
3. Contributors, M.: MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d` (2020), accessed: 07.03.22
4. DETR3D Github-Repository. `https://github.com/WangYueFt/detr3d`, accessed: 07.03.22
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
6. Gao, P., Zheng, M., Wang, X., Dai, J., Li, H.: Fast convergence of detr with spatially modulated co-attention. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3621–3630 (2021)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
8. Huang, J., Huang, G., Zhu, Z., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
9. Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., Carreira, J.: Perceiver: General perception with iterative attention. In: International Conference on Machine Learning (ICML). pp. 4651–4664 (2021)
10. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12697–12705 (2019)
11. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2117–2125 (2017)
12. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2980–2988 (2017)
13. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10012–10022 (2021)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
15. Ma, H., Chen, L., Kong, D., Wang, Z., Liu, X., Tang, H., Yan, X., Xie, Y., Lin, S.Y., Xie, X.: Transfusion: Cross-view fusion with transformer for 3d human pose estimation. arXiv preprint arXiv:2110.09554 (2021)
16. Ma, X., Liu, S., Xia, Z., Zhang, H., Zeng, X., Ouyang, W.: Rethinking pseudo-lidar representation. In: European Conference on Computer Vision (ECCV). pp. 311–327 (2020)

17. nuScenes Detection Task. https://www.nuscenes.org/object-detection/, accessed: 07.03.22
18. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3142–3152 (2021)
19. Prakash, A., Chitta, K., Geiger, A.: Multi-modal fusion transformer for end-to-end autonomous driving. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7077–7087 (2021)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
21. Roh, B., Shin, J., Shin, W., Kim, S.: Sparse detr: Efficient end-to-end object detection with learnable sparsity. arXiv preprint arXiv:2111.14330 (2021)
22. Sun, Z., Cao, S., Yang, Y., Kitani, K.M.: Rethinking transformer-based set prediction for object detection. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3611–3620 (2021)
23. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9627–9636 (2019)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
25. Wang, C., Ma, C., Zhu, M., Yang, X.: Pointaugmenting: Cross-modal augmentation for 3d object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11794–11803 (2021)
26. Wang, T., Xinge, Z., Pang, J., Lin, D.: Probabilistic and geometric depth: Detecting objects in perspective. In: Conference on Robot Learning (CORL). pp. 1475–1485 (2022)
27. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 913–922 (2021)
28. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8445–8453 (2019)
29. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning (CORL). pp. 180–191 (2022)
30. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11784–11793 (2021)
31. Zhang, W., Wang, Z., Change Loy, C.: Multi-modality cut and paste for 3d object detection. arXiv e-prints pp. arXiv–2012 (2020)
32. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. arXiv preprint arXiv:1908.09492 (2019)
33. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)