Lift-Attend-Splat: Bird's-eye-view camera-lidar fusion using transformers

James Gunn* Zygmunt Lenyk* Anuj Sharma Andrea Donati
Alexandru Buburuzan John Redford Romain Mueller
FiveAI
<first>.<last>@five.ai

Abstract

Combining complementary sensor modalities is crucial to providing robust perception for safety-critical robotics applications such as autonomous driving (AD). Recent state-of-the-art camera-lidar fusion methods for AD rely on monocular depth estimation which is a notoriously difficult task compared to using depth information from the lidar directly. Here, we find that this approach does not leverage depth as expected and show that naively improving depth estimation does not lead to improvements in object detection performance. Strikingly, we also find that removing depth estimation altogether does not degrade object detection performance substantially, suggesting that relying on monocular depth could be an unnecessary architectural bottleneck during camera-lidar fusion. In this work, we introduce a novel fusion method that bypasses monocular depth estimation altogether and instead selects and fuses camera and lidar features in a bird's-eye-view grid using a simple attention mechanism. We show that our model can modulate its use of camera features based on the availability of lidar features and that it yields better 3D object detection on the nuScenes dataset than baselines relying on monocular depth estimation.

1. Introduction

Integrating information from different modalities efficiently and effectively is especially important in safety-critical applications such as autonomous driving, where different sensor modalities are complementary and combining them adequately is crucial to guarantee safety. For example, cameras capture rich semantic information of objects up to far away distances, while lidars provide extremely accurate depth information but are sparse at large distances. For this reason, many modern self-driving platforms have a large number of different sensors which must be combined in order to provide accurate and reliable perception of the surrounding scene and allow safe deployment of these vehicles in the real world.

Multimodal sensor fusion — learning a unified representation of a scene derived from multiple sensors — offers a plausible solution to this problem. However, training such multimodal models can be challenging, especially when modalities are as different as cameras (RGB images) and lidars (3D point clouds). For instance, it is known that different modalities overfit and generalise at different rates [57] and that training all modalities jointly can lead to underutilisation of the weaker modalities and even to inferior results compared to unimodal models in some situations [40].

In the context of autonomous driving, many of the recent state-of-the-art methods for camera-lidar fusion [16, 31, 36] are based on the Lift-Splat (LS) paradigm [41][1]. In this approach, the camera features are projected in bird's-eye-view (BEV) — or top-down space — using monocular depth before being fused with the lidar features. As a result, the location of the camera features in BEV is highly dependent on the quality of the monocular depth prediction and it has been argued that its accuracy is critical [16, 31]. In this work, we reconsider these claims and show that the monocular depth prediction inside these models is of poor quality and cannot account for their success. In particular, we present results showing that methods based on Lift-Splat perform equally well when the monocular depth prediction is replaced by direct depth estimation from the lidar point cloud or removed completely. This leads us to suggest that relying on monocular depth when fusing camera and lidar features is an unnecessary architectural bottleneck and that Lift-Splat could be replaced by a more effective projection.

We introduce a novel approach for camera-lidar fusion called "Lift-Attend-Splat" that bypasses monocular depth estimation altogether and instead selects and fuses camera and lidar features in BEV using a simple transformer. We present evidence that our method shows better camera utilisation compared to the methods based on monocular depth estimation and that it improves object detection per-

*Equal contribution

[1] The "shoot" component of "Lift, Splat, Shoot" [41] relates to trajectory prediction and is not considered here.

formance. Our contributions are as follows:

- We show that camera-lidar fusion methods based on the Lift-Splat paradigm are not leveraging depth as expected. In particular, we show that they perform equivalently or better if monocular depth prediction is removed completely.
- We introduce a novel camera-lidar fusion method that fuses camera and lidar features in BEV using a simple attention mechanism. We show that it leads to better camera utilisation and improves 3D object detection compared to models based on the Lift-Splat paradigm.

2. Related work

3D object detection for autonomous driving Most 3D object detection benchmarks are dominated by methods using lidar point clouds due to their highly accurate range measurement allowing for better placement of objects in 3D compared to methods using cameras or radars only. Deep learning methods for classification on point clouds were pioneered in the seminal works of [43, 44] and early works have been applying similar ideas to 3D object detection [45, 48]. A more recent family of methods is based on direct voxelisation of the 3D space [64, 70] or compression of the lidar representation along the z-direction into "pillars" [25, 65]. These approaches have been very successful and are the basis of many follow-up works [17, 22, 67].

The task of 3D object detection has also been tackled from multiple cameras alone. Early works have mostly been based on various two-stage approaches [4, 23, 45, 58], while recent methods have been leveraging monocular depth estimation directly [3, 24, 46]. This task is difficult when lidar is absent because 3D information must be estimated using images only, which is a challenging problem. However, recent works have shown impressive performance by borrowing ideas from lidar detection pipelines [8, 12, 18], by improving position embeddings [34] and 3D queries [20], as well as by leveraging temporal aggregation [13, 28, 33, 35, 56, 72] or 2D semantic segmentation [69].

Camera-lidar fusion Perception quality can be improved by jointly leveraging cameras and lidars when available. Recent fusion methods can be broadly classified into three categories: point decoration methods, methods that leverage task-specific object queries and architectures, and projection based methods. Point decoration methods augment the lidar point cloud using semantic segmentation data [52, 61], camera features [54], or even create new 3D points using object detections in the image plane [68]. Such methods are relatively easy to implement but suffer from the fact that they require lidar points to fuse camera features. TransFusion [1] is a recent example of a method that leverages task-specific object queries generated using the lidar point cloud. Final detections are made directly without explicit projection of camera features into BEV space. Similar methods utilising two-way modality interactions perform even better [62, 66]. Fusion can also be performed earlier in the model, for example at the level of the 3D voxels [6, 7] or lidar features [26], or by sharing information between the camera and lidar backbones [19, 29, 42]. MSMDFusion [21] fuses camera and lidar features at multiple scales using lidar points to estimate the 3D position of camera features, while UniTR [55] pre-assigns depth to each camera feature. FUTR3D [5] fuses features by selecting modality-agnostic 3D reference points. Finally, projection-based methods project camera features into 3D before fusing them with the lidar (see below).

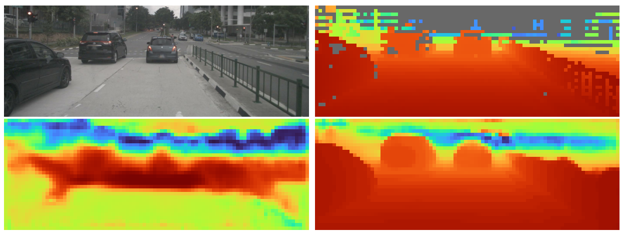
Projection based methods Of most interest to us are camera-lidar fusion methods based on projecting camera features into 3D. Many recent state-of-the-art camera-lidar fusion methods [16, 31, 36] project camera features in 3D using monocular depth estimation even though depth information is available from the lidar. It has also been shown that the projection method is less important than other aspects of training in the camera-only setting [14]. An alternative approach is to project camera features directly into BEV space using the known correspondence between lidar points and camera features [9, 26, 59]. However, the sparsity of the lidar point cloud can limit which camera features are projected, as described in [36]. Finally, learning to project camera features in BEV without explicit depth can be achieved when lidar is absent using a transformer, as shown in [28, 47]. Here, we extend this line of work to the case of camera-lidar fusion and leverage cross-attention to generate a dense BEV grid of fused lidar features.

3. Monocular depth prediction in Lift-Splat

Recent camera-lidar fusion methods based on the Lift-Splat paradigm [31, 36] learn a unified representation in the form of a BEV grid by projecting camera features in BEV space using an estimated depth distribution as

$$\text{Proj}_{\text{Lift-Splat}} = \text{Splat}\left(F'^{\text{cam}} \otimes D\right), \quad (1)$$

where $F'^{\text{cam}} \in \mathbb{R}^{C'_c \times H \times W}$ is a context vector obtained from the camera features $F^{\text{cam}} \in \mathbb{R}^{C_c \times H \times W}$, $D \in \mathbb{R}^{N_D \times H \times W}$ is a normalised distribution over predetermined depth bins and Splat denotes the operation of projecting each point downwards into the $z = 0$ plane, see [31, 36, 41] for details. The resulting feature map is then merged with the lidar features using concatenation [36] or gated attention [31]. In this paradigm, the monocular depth distribution prediction is learned indirectly from the downstream task without explicit depth supervision.



	Abs. Rel. ↓	RMSE ↓	mAP ↑
BEVFusion [36]	2.75	17.40	**68.5**
BEVFusion [36] w/ Eq. (2):			
$\lambda = 0$	2.83	18.54	68.4
$\lambda = 0.01$	0.76	8.09	68.0
$\lambda = 1$	0.22	4.77	68.1
$\lambda = 100$	0.16	4.55	64.6
Lidar	0.04	0.29	68.4
Pretrained	0.64	7.87	67.4
Uniform depth	–	–	**68.5**

Figure 1. Impact of the quality of the monocular depth prediction on the object detection performance of BEVFusion [36] on the nuScenes validation set. We compare BEVFusion and four different variants: adding depth supervision using Eq. (2) with various weights λ, using lidar depth maps instead of monocular depth estimation (lidar), using a pretrained and frozen depth classifier (pretrained), and finally removing depth estimation altogether by projecting camera features at all depths uniformly using Eq. (3) (uniform depth). In our experiments, more accurate depth does not translate to better detection performance and the original model is on-par with using the lidar points directly as a source of depth. Equivalent detection performance was achieved using the *uniform depth* model, clearly indicating that accurate monocular depth is not necessary for BEVFusion [36] to achieve its performance, see main text and Sec. A.4 for details.

Lift-Splat depth prediction is generally poor We analyse the quality of the depth distribution predicted by BEVFusion [36] by comparing its mean value to lidar depth maps, both qualitatively and quantitatively using the absolute relative (Abs. Rel.) and root mean squared errors (RMSE) [10, 27]. As shown on Fig. 1, the mean depth prediction does not accurately reflect the structure of the scene and is markedly different from the lidar depth map which suggests that monocular depth is not leveraged as expected in [36]. See Secs. A.2 and A.4 for details.

Improving depth prediction does not improve detection performance We next investigate whether improving the depth prediction quality can boost object detection performance. To do so, we retrain the model from [36] with the following loss:

$$L_{\text{total}} = L_{\text{sup}} + \lambda L_{\text{depth}}, \tag{2}$$

where L_{sup} is the original 3D object detection loss and L_{depth} is a simple cross-entropy loss for the depth estimation that uses one-hot encoded lidar depth as a target, see Sec. A.3 for more details. By changing the hyper-parameter λ, we can control the quality of the depth prediction and explore how it impacts detection performance. In Fig. 1, we see that while depth supervision indeed leads to much more accurate depth maps both visually and quantitatively, detection performance — measured using mean average precision (mAP) — degrades from the baseline as the weight of the depth supervision is increased. This suggests that the method is unable to take advantage of more accurate depth prediction. Since training on the multi-task loss Eq. (7) is likely to degrade object detection performance at high values of λ, we also experiment with two more variants: (i) pretraining the depth supervision module separately and (ii) using the lidar point cloud directly to bypass the depth supervision module altogether. Pretraining leads to more accurate depth prediction but degrades detection performance relative to the baseline, while using the lidar directly does not change the detection performance compared to the baseline, even though all depth metrics are close to zero[2].

Removing depth prediction altogether does not affect object detection performance The results above lead us to hypothesise that accurate monocular depth is not leveraged in camera-lidar fusion methods based on the Lift-Splat projection. To test this, we remove the monocular depth prediction completely and replace the projection (1) by

$$\text{Proj}_{\text{no-depth}} = \text{Splat}\left(F'^{\text{cam}} \otimes 1\right), \tag{3}$$

where we denote by 1 the tensor of the same shape as D with all entries equal to 1. This projects the camera features to all depths uniformly. Strikingly, we see in Fig. 1 (right) that *removing monocular depth estimation does not lead to a degradation in object detection performance*, suggesting that accurate depth estimation is not a key component of this method. We hypothesise that the importance of monocular depth is greatly diminished when lidar features are available since lidar is a much more precise source of depth information and that the model is able to easily suppress camera features projected at the wrong location. This suggests that relying on monocular depth estimation could be an unnecessary architectural bottleneck and lead to underutilisation of the camera.

[2]They are not exactly zero because of the depth quantisation introduced by the one-hot encoding of the lidar depth, see Sec. A.1.

4. Camera-lidar fusion without monocular depth estimation

In this section, we present a camera-lidar fusion method that bypasses monocular depth estimation altogether and instead fuses camera and lidar features in bird's-eye-view using a simple transformer [51]. A naive application of the transformer architecture to the problem of camera-lidar fusion is difficult, however, due to the large number of camera and lidar features and the quadratic nature of attention. As shown in [47], it is possible to use the geometry of the problem to drastically restrict the scope of the attention when projecting camera features in BEV, since camera features should only contribute to locations along their corresponding rays. We adapt this idea to the case of camera-lidar fusion and introduce a simple method that uses cross-attention between columns in the camera plane and polar rays in the lidar BEV grid. Instead of predicting monocular depth, cross-attention learns which camera features are the most salient given context provided by the lidar features along its ray.

Except for the projection of the camera features in BEV, our model shares a similar overall architecture to methods based on the Lift-Splat paradigm [16, 31, 36] and is depicted on Fig. 2 left. It consists of the following modules: the camera and lidar backbones which produce features for each modality independently, a projection and fusion module that embeds the camera features into BEV and fuses them with the lidar, and finally a detection head. When considering object detection, the final output of the model is the property of objects in the scene represented as 3D bounding boxes with position, dimension, orientation, velocity and classification information. In what follows we explain in detail the architecture of our projection and fusion modules.

Projected horizon For each camera, we consider the horizontal line passing through the centre of the image and the plane corresponding to its projection in 3D. We call this plane the *projected horizon* of the camera. It can easily be described using homogeneous coordinates as the set of points $\mathbf{x} \in \mathbb{R}^4$ for which there exists a $u \in \mathbb{R}$ such that

$$\mathbf{C}\mathbf{x} \sim (u, h/2, 1), \tag{4}$$

where $\mathbf{C}$ is the 3×4 camera projection matrix (intrinsic and extrinsic), and h is the height of the image. Note that this plane is not in general parallel to the BEV grid, its relative orientation being defined by the camera's extrinsic parameters. We define a regular grid on each camera's projected horizon that is aligned with the 2D grid of features in its image plane by tracing out rays from the intersection of the horizontal line with the edges of the feature columns in the image plane, and then separating these rays into a set of predetermined depth bins (similarly to [31]). Features on this grid can be represented by a matrix $G \in \mathbb{R}^{N_D \times W}$, where each row corresponds to a specific column in the camera feature map $F^{\text{cam}} \in \mathbb{R}^{H \times W \times C}$. The geometry of a projected horizon can be seen in Fig. 2 (left, inset). The projected horizon allows for a consistent definition of depth between differently pitched cameras.

Correspondence between projected horizons and BEV grid We can easily define a correspondence between points on a projected horizon and points on the BEV plane by projecting them along the z-direction in 3D space. As cameras are in general tilted with respect to the ground, this correspondence depends on each camera's extrinsic parameters. We transfer lidar features from the BEV grid to a camera's projected horizon through bi-linear sampling of the BEV grid at the locations of the down-projected cell-centers of the projected horizon. We call this process "lifting" and denote it as Lift_i for the projected horizon of camera i. Similarly, camera features can be transferred in the opposite direction, from a projected horizon to the BEV grid, by bi-linearly sampling the projected horizon at the locations of the projected cell-centers of the BEV grid. We denote this operation as Splat_i, similarly to [31, 36, 41]. Fusion of lidar features with splatted camera features takes place in BEV space, as is common [31, 36].

Lift-Attend-Splat Our projection module is depicted in Fig. 2 (right) and can be broken down into three simple steps: (i) we first *lift* the BEV lidar features B^{lid} onto the projected horizon of camera i, producing "lifted" lidar features $\tilde{B}_i^{\text{lid}}$, (ii) we then let the "lifted" lidar features *attend* to the camera features in the corresponding column using a simple transformer encoder-decoder, producing fused features $\tilde{B}_i^{\text{fus}}$ on the i^{th} projected horizon, and finally (iii) we *splat* these features back onto the BEV grid to produce B_i^{fus}. During the attend step, the camera features in each column are encoded by a transformer encoder E and passed as keys and values to a transformer decoder D which uses the frustum lidar features as queries. The result of these three steps can be written as

$$B_i^{\text{fus}} = \text{Splat}_i\left(D\left(\text{Lift}_i\left(B^{\text{lid}}\right), E\left(F_i^{\text{cam}}\right)\right)\right), \tag{5}$$

where Lift_i and Splat_i project the BEV features onto the projected horizon of camera i (and vice versa) as described above. Finally, we apply a simple fusion module where we sum the projected features from different cameras together, concatenate them with the lidar features and apply a convolutional block to obtain the final features in BEV. This simple architecture allows the camera features to be projected from the image plane onto the BEV grid without requiring monocular depth estimation. We share a single set of transformer weights across all column-frustum pairs and cameras. For simplicity, we use here a single transformer

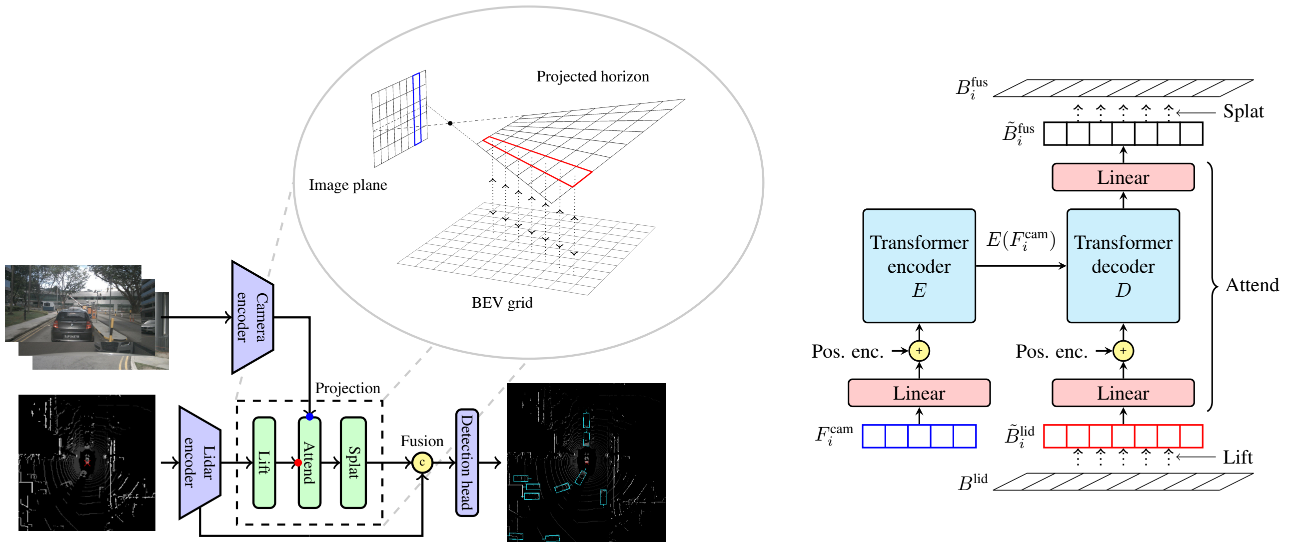


Figure 2. Lift-Attend-Splat camera-lidar fusion architecture. (left) Overall architecture: features from the camera and lidar backbones are fused together and merged before being passed to a detection head. (inset) Geometry of our 3D projection: the "Lift" step embeds the lidar BEV features into the projected horizon by lifting the lidar features along the z-direction using bilinear sampling. The "Splat" step corresponds to the inverse transformation in that it projects features from the projected horizon back onto the BEV grid using bilinear sampling, again along the z-direction. (right) Details of the projection module: the "Attend" step in our method lets the lifted lidar features $\tilde{B}_i^{\text{lid}}$ attend to the camera features F_i^{cam} in the corresponding column using a simple encoder-decoder transformer architecture to produce fused features $D(\tilde{B}_i^{\text{lid}}, E(F_i^{\text{cam}}))$ in frustum space.

encoder and decoder but show in Sec. 5.4 that adding more can be beneficial. All camera features participate in our attention, not just a small number of reference points as is the case in [28]. This learnt set of salient features initialises our object detection queries, rather than the fixed maxpool of [1].

Attention vs depth prediction It is worth discussing how our approach differs from predicting monocular depth directly. When using monocular depth, each feature in the camera feature map is projected into BEV at multiple locations weighted by a normalised depth distribution. This normalisation limits each feature to be projected either into a single location or smeared with lower intensity across multiple depths. However, in our approach, the attention between camera and lidar is such that the same camera feature can contribute fully to multiple locations in the BEV grid. This is possible because attention is normalised over keys, which correspond to different heights in the camera feature map, rather than queries, which correspond to different distances along the ray. Furthermore, our model has access to lidar features in BEV when choosing where to project camera features, which gives it greater flexibility. Finally, our projection requires fewer parameters than competing methods: 0.9M for our attention-based module compared to 1.6M in the equivalent component of [36].

5. Experiments

We measure the effectiveness of our approach against recent camera-lidar fusion methods that use the Lift-Splat paradigm [31, 36]. In all of our experiments, we concentrate on 3D object detection using the nuScenes dataset [2], which is a large-scale dataset for autonomous driving. We use the nuScenes detection score (NDS) and mean average precision (mAP) as evaluation metrics. We do not consider the extension of [31, 36] presented in [16] as it introduces two supplementary dense depth supervision losses on the camera path to significantly boost the performance of the underlying methods. In this work, we use solely the 3D object detection losses present in [1, 31, 36] and leave applying the framework of [16] to our method for future work.

Overall architecture We use Dual-Swin-Tiny [30] with a feature pyramid network [32] and VoxelNet [70] as our camera and lidar encoders respectively. Our object detection head is the transformer-decoder-based module from TransFusion-L [1]. We use our Lift-Attend-Splat method, described in Sec. 4, to project camera features into BEV space. We then fuse camera and lidar features using simple concatenation and convolution. Following [36] the RPN part of VoxelNet is applied to the merged feature. We ablate alternative choices for the fusion architecture in Sec. 5.4.

Implementation details Inputs to the camera encoder have resolution 800x448, which it downsamples by 8x into per-camera feature maps of shape 100x56. For VoxelNet, we follow the settings of [31]. We set a maximum of 90k non-empty voxels during training, increased to 180k for inference. We use an ego-centric BEV grid with dimensions 108m $\times$ 108m and 0.075m cell size. This is downsampled 8x by the lidar encoder to the 180×180 grid into which the camera features are projected. We construct the intermediate projected horizon with 143 uniformly spaced depth bins ranging from 1m to 72m. For the projection of camera features into BEV, we use the original transformer [51] as our encoder-decoder architecture, with one encoder layer, one decoder layer, $d_{\text{model}} = 256$ and $d_{\text{ff}} = 512$. We replace the ReLU [39] activation function with GeLU [15], use learnable position embeddings [11] in place of sinusoidal encodings and normalise features before each sublayer [60]. We tie the parameters in each of the 8 heads of our attention blocks. For the object detection head, we use 200 and 300 queries during training and inference respectively.

Training details Our lidar backbone is pretrained on 8 GPUs with batch-size of 1/GPU following the schedule presented in [1], with CBGS [71] and copy-paste augmentation [63]. We initialise the camera backbone with weights pretrained on nuImages [2] by [31]. We freeze the lidar backbone and train the camera backbone, projection, fusion and detection head for 20 epochs using 8 GPUs with a batch size of 4/GPU. We use the AdamW optimiser [37] with a maximum learning rate of 5×10^{-5} for the camera backbone and 1×10^{-3} for all other components. We apply the following augmentations: mirroring in the Y dimension, global rotation and scale, and camera-lidar copy-paste [53].

5.1. 3D object detection

We present results for the task of 3D object detection on Tab. 1. Compared to baselines based on the Lift-Splat projection [31, 36], our method shows improvements on the validation and test splits of the nuScenes dataset. In particular, we show substantial improvements in both mAP (+1.1) and NDS (+0.4) on the test split. The lidar backbone is frozen and similar in those approaches, showing that our model is better able to leverage camera features. Our method outperforms the more recent fusion algorithms TransFusion, DeepInteraction, and FUTR3D by a substantial margin, performs similarly to UniTR and MSMDFusion, and slightly underperforms CMT. These results show that our simple modification of BEVFusion is successful in raising its performance on par with recent SOTA methods. We leave extending our method with multi-scale feature fusion (used by MSMDFusion) and query denoising (used by CMT) to future work. Per-class object detection results and comparisons can be found in Tab. S2 and Sec. B.1.

	val.		test	
	mAP	NDS	mAP	NDS
BEVFusion [36]	68.5	71.4	70.2	72.9
BEVFusion [31]	69.6	72.1	71.3	73.3
FUTR3D [5]	64.2	68.0	69.4	72.1
TransFusion [1]	67.5	71.3	68.9	71.6
DeepInteraction [66]	69.9	72.6	70.8	73.4
MSMDFusion [21]	-	-	71.5	74.0
CMT [62]	70.3	72.9	72.0	74.1
UniTR [55]	70.5	73.3	70.9	74.5
Ours	71.2	72.7	71.5	73.6
Ours w/ TFA	72.1	73.8	-	-
BEVFusion‡ [36]	73.7	74.9	75.0	76.1
Ours‡	74.6	75.1	-	-
Ours w/ TFA‡	75.7	76.0	75.5	74.9

Table 1. Object detection performance on the validation and test splits of the nuScenes dataset. TFA: Temporal Feature Aggregation. ‡ denotes test-time augmentations and model ensembling.

We can analyse further the performance of our model by clustering objects together depending on their distances from the ego and on their sizes, see Fig. 3. We see that the bulk of the improvements comes from objects located at large distances and of small sizes. These are situations for which monocular depth estimation is particularly difficult which could explain why our model fares better in these cases. Note that even though far-away and small objects contain fewer lidar points, our model is still able to leverage camera features effectively even though the context given by the lidar is weaker.

We show results that use test-time-augmentations (TTA) and model ensembling at the bottom of Tab. 1. We perform TTA over a combination of mirror and rotation augmentations and ensemble models with cell resolutions of 0.05m, 0.075m and 0.10m. We first apply TTA at each cell resolution and then merge the resulting boxes using Weighted Boxes Fusion (WBF) [50]. Unsurprisingly, our method shows excellent scaling for these techniques and outperforms BEVFusion [36] on the nuScenes validation set. More details can be found in Sec. B.3.

5.2. Qualitative analysis

We visualise where camera features are projected onto the BEV grid and compare our method to BEVFusion [36]. For our method, we examine the attention map of the final cross-attention block in the transformer, averaged over all attention heads. For BEVFusion, we use the the monocular depth estimate to establish the strength of correspondence between positions in camera and BEV space. We consider only the pixels corresponding to ground-truth objects when calculating the total weight of projected camera

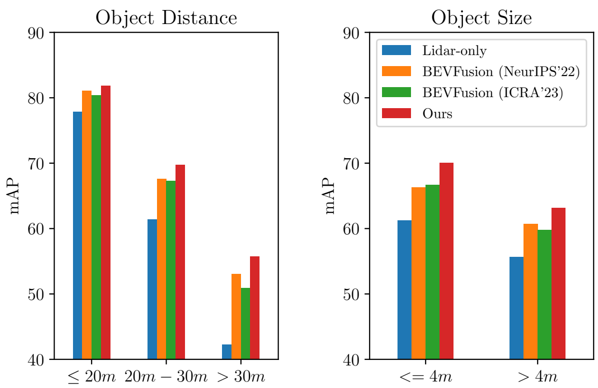


Figure 3. Object detection performance measured using mAP for objects at different distances from the ego and of different sizes. Our model consistently outperforms baselines based on Lift-Splat, especially at large distances and for small objects.

		mAP	NDS
Fusion module	Cat+Conv	**70.43**	71.9
	Gated sigmoid [31]	70.12	71.9
	Add	70.32	**72.1**
# decoder blocks*	1 block	70.29	71.9
	2 blocks	70.40	**72.0**
	4 blocks	**70.49**	71.9
# TFA frames	1 frame (no TFA)	71.2	72.8
	2 frames	72.1	73.3
	3 frames	**72.1**	**73.8**

Table 2. Impact of model modifications on 3D object detection performance: (i) feature fusion module, (ii) number of transformer decoder blocks in the "Attend" stage, (iii) number of frames in Temporal Feature Aggregation (TFA). * frozen camera backbone.

features in BEV, see Sec. B.2 for details. As can be seen in Fig. 4a, our method places camera features predominantly in regions where ground-truth bounding boxes are present. This indicates that it can effectively leverage the lidar point cloud as a context to project camera features at the relevant location in BEV. Compared to BEVFusion shown in Fig. 4b, the distribution of features appears more narrowly localised and stronger around objects. This could be because our projection mechanism does not require the weights of the camera features to be normalised along their ray, giving our model more flexibility to place features at the desired location. Note that, even though our method also projects camera features outside of ground-truth boxes in BEV, the strength of the activation in these regions is suppressed by the fusion module, see Fig. S4. This is consistent with our findings in Sec. 3, where we showed that the latter part of the model can effectively suppress camera features at the wrong location. More examples can be found in Sec. B.2.

We further explore which pixels in the camera images are most attended to using saliency maps [49]. These are derived by computing the gradient of the maximum class logit with respect to a camera image I_j, given object query index i and probabilities z, as $\partial z_{i,\hat{c}}/\partial I|_{I_j}$ where $\hat{c} = \arg\max_c z_{i,c}$. They allow us to visualise the contribution of individual pixels to the final detection for a selected object, see Fig. 4c. Interestingly, we observe that when trained with both camera and lidar, our model tends to select camera features at different locations than when trained with cameras only. In the absence of lidar, our method selects camera features across the entirety of the object, while in the presence of both lidar and cameras, the model selects camera features mainly from the upper part of the object. We observe that this pattern is mostly prevalent for objects close to the ego which are well-represented by lidar point clouds but fades away for far-away objects or objects with few lidar points such as pedestrians, see Fig. S2 for more examples. We hypothesise that our projection architecture enables the model to select camera features that best complement the information encoded in lidar, resulting in differing attention patterns between camera-only and fusion settings. This pattern is less present in BEVFusion [36], which attends to the broader neighbourhood of pixels surrounding the selected object in both cases.

5.3. Temporal feature aggregation

Because our method fuses camera and lidar features onto a BEV grid, we can easily leverage past information using temporal feature aggregation (TFA). To achieve this, we implement the simple autoregressive procedure of VideoBEV [13] but aggregate the fused BEV features $B^{\text{fus.}}$ instead of the camera features. Our method is as follows: (i) save the fused BEV features from the previous timestep, (ii) apply ego motion compensation to align these features with the current timestep, using bilinear sampling to construct the new feature grid, (iii) concatenate these features with the fused BEV features of the current timestep and merge them using a simple 3×3 convolutional block.

We train TFA models on sequences of 3 frames for 10 epochs starting from our single frame model's object detection head, lidar and camera backbones. During training, the lidar and camera backbones are frozen. For inference, we accumulate BEV features for the entire length of a run, yielding detections at each time step. Table 1 shows that temporal feature aggregation boosts object detection performance significantly in all configurations.

5.4. Ablation experiments

We ablate some design choices for our method and show their impact on object detection performance on Tab. 2. All ablation experiments use a simpler training schedule with 10 epochs, batch accumulation instead of full batch training

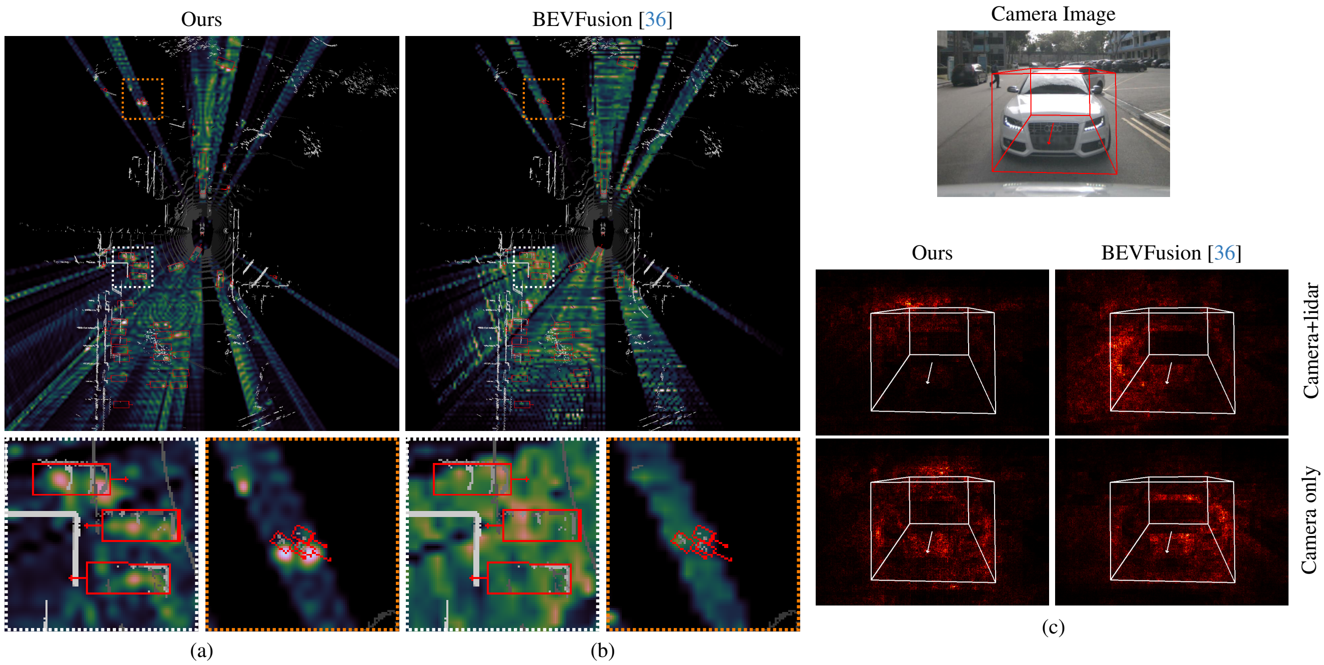


Figure 4. **(a, b)** Visualisation of where camera features of ground-truth objects are projected onto the BEV grid for our method compared to BEVFusion [36]. We observe that our method is able to place camera features around objects more narrowly than BEVFusion, which is based on monocular depth estimation. **(c)** Comparison of saliency maps, cropped to aid visualisation, given the camera image (top) for models trained with camera-lidar (middle) and camera only (bottom). When trained with both camera and lidar, our model selects camera features in an area that is different than when trained with camera only, while [36] behaves similarly in both settings.

and no camera augmentations. We first analyse the effect of different fusion modules: we compare a simple skip connection (add), a small concatenation and convolution layer (Cat+Conv as in [36]) and a gated sigmoid block [31]. We find all perform similarly, with Cat+Conv achieving the best mAP, contrary to findings of [31]. We also ablate the number of transformer decoder blocks in the "Attend" stage of our projection and show that increasing their number leads to a small improvement in mAP. This suggests that our method's performance scales with compute. We use a single decoder block in our main experiments to balance quality and performance. Finally, we see good improvement in NDS with an increasing number of frames in TFA during training.

6. Conclusion

In this work, we analysed the role of monocular depth prediction in recent state-of-the-art camera-lidar fusion methods and showed that, surprisingly, improvements in depth estimation did not lead to better object detection performance. Strikingly, we also showed that removing depth estimation altogether did not worsen performance significantly. This led us to hypothesise that relying on monocular depth estimation could be an unnecessary architectural bottleneck when fusing camera and lidar, and prompted us to introduce a novel fusion method that directly combines camera and lidar features using a simple attention mechanism. Compared to projecting camera features using monocular depth, our method allows camera features to contribute to multiple locations in BEV space and gives our model greater flexibility to select complementary camera and lidar features. Finally, we validated the effectiveness of our method on the nuScenes dataset and showed that it improves object detection performance over baselines based on monocular depth estimation and showcased the role of attention as a key contributor to these improvements. We leave detailed investigation of our model in the camera-only setting and inclusion of radar as future work. We hope that our findings will motivate discussions around the role of monocular depth prediction in camera-lidar fusion and prompt further developments in multimodal perception.

Acknowledgements

We thank our FiveAI and Bosch colleagues, especially Tom Joy and Anthony Knittel, for their valuable feedback. We are grateful to Bhavesh Garg for his initial experiments into depth supervision and to Blaine Rogers for initiating work on multimodal fusion at FiveAI.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1090–1099, 2022. 2, 5, 6

[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, 6

[3] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2147–2156, 2016. 2

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2

[5] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 172–181, 2023. 2, 6

[6] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5428–5437, 2022. 2

[7] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinghong Jiang, Feng Zhao, Bolei Zhou, and Hang Zhao. Autoalign: pixel-instance feature aggregation for multimodal 3d object detection. *arXiv preprint arXiv:2201.06493*, 2022. 2

[8] Zhiyu Chong, Xinzhu Ma, Hong Zhang, Yuxin Yue, Haojie Li, Zhihui Wang, and Wanli Ouyang. Monodistill: Learning spatial features for monocular 3d object detection. *arXiv preprint arXiv:2201.10830*, 2022. 2

[9] Florian Drews, Di Feng, Florian Faion, Lars Rosenbaum, Michael Ulrich, and Claudius Gläser. Deepfusion: A robust and modular 3d object detector for lidars, cameras and radars. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 560–567. IEEE, 2022. 2

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 3, 12

[11] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017. 6

[12] Xiaoyang Guo, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3153–3163, 2021. 2

[13] Chunrui Han, Jianjian Sun, Zheng Ge, Jinrong Yang, Runpei Dong, Hongyu Zhou, Weixin Mao, Yuang Peng, and Xiangyu Zhang. Exploring recurrent long-term temporal fusion for multi-view 3d perception. *arXiv preprint arXiv:2303.05970*, 2023. 2, 7

[14] Adam W Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-bev: What really matters for multi-sensor bev perception? In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2759–2765. IEEE, 2023. 2

[15] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 6

[16] Haotian Hu, Fanyi Wang, Jingwen Su, Yaonong Wang, Laifeng Hu, Weiye Fang, Jingwei Xu, and Zhiwang Zhang. Ea-lss: Edge-aware lift-splat-shot framework for 3d bev object detection, 2023. 1, 2, 4, 5

[17] Dihe Huang, Ying Chen, Yikang Ding, Jinli Liao, Jianlin Liu, Kai Wu, Qiang Nie, Yong Liu, Chengjie Wang, and Zhiheng Li. Rethinking dimensionality reduction in grid-based 3d object detection. *arXiv preprint arXiv:2209.09464*, 2022. 2

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

[19] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 35–52. Springer, 2020. 2

[20] Xiaohui Jiang, Shuailin Li, Yingfei Liu, Shihao Wang, Fan Jia, Tiancai Wang, Lijin Han, and Xiangyu Zhang. Far3d: Expanding the horizon for surround-view 3d object detection. *arXiv preprint arXiv:2308.09616*, 2023. 2

[21] Yang Jiao, Zequn Jie, Shaoxiang Chen, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. Msmdfusion: Fusing lidar and camera at multiple scales with multi-depth seeds for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21643–21652, 2023. 2, 6

[22] Junho Koh, Junhyung Lee, Youngwoo Lee, Jaekyum Kim, and Jun Won Choi. Mgtanet: Encoding sequential lidar points using long short-term motion-guided temporal attention for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1179–1187, 2023. 2

[23] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2

[24] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11867–11876, 2019. 2

[25] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 2

[26] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17182–17191, 2022. 2

[27] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1477–1485, 2023. 3, 12

[28] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022. 2, 5

[29] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 641–656, 2018. 2

[30] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and Haibin Ling. CBNet: A composite backbone network architecture for object detection. *IEEE Transactions on Image Processing*, 31: 6893–6906, 2022. 5

[31] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, 35:10421–10434, 2022. 1, 2, 4, 5, 6, 7, 8, 13

[32] 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. 5

[33] 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. 2

[34] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, pages 531–548. Springer, 2022. 2

[35] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Aqi Gao, Tiancai Wang, and Xiangyu Zhang. Petrv2: A unified framework for 3d perception from multi-camera images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3262–3272, 2023. 2

[36] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781. IEEE, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14, 15, 17

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

[38] Anton Mikhailov. Turbo, an improved rainbow colormap for visualization. `https://blog.research.google/2019/08/turbo-improved-rainbow-colormap-for.html`, 2019. Accessed: 20th October 2023. 12

[39] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 6

[40] Kranti Parida, Neeraj Matiyali, Tanaya Guha, and Gaurav Sharma. Coordinated joint multimodal embeddings for generalized audio-visual zero-shot classification and retrieval of videos. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3251–3260, 2020. 1

[41] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. 1, 2, 4

[42] AJ Piergiovanni, Vincent Casser, Michael S Ryoo, and Anelia Angelova. 4d-net for learned multi-modal alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15435–15445, 2021. 2

[43] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[44] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[45] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2

[46] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5881–5890, 2020. 2

[47] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Translating images into maps. In *2022 International conference on robotics and automation (ICRA)*, pages 9200–9206. IEEE, 2022. 2, 4

[48] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 2

[49] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 7

[50] Roman A. Solovyev and Weimin Wang. Weighted boxes fusion: ensembling boxes for object detection models. *CoRR*, abs/1910.13302, 2019. 6, 13

[51] 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, 6

[52] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020. 2

[53] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11794–11803, 2021. 6

[54] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11794–11803, 2021. 2

[55] Haiyang Wang, Hao Tang, Shaoshuai Shi, Aoxue Li, Zhenguo Li, Bernt Schiele, and Liwei Wang. Unitr: A unified and efficient multi-modal transformer for bird's-eye-view representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6792–6802, 2023. 2, 6

[56] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. *arXiv preprint arXiv:2303.11926*, 2023. 2

[57] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12695–12705, 2020. 1

[58] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1742–1749. IEEE, 2019. 2

[59] Zining Wang, Wei Zhan, and Masayoshi Tomizuka. Fusing bird view lidar point cloud and front view camera image for deep object detection, 2018. 2

[60] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020. 6

[61] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Zhou Bin, and Liangjun Zhang. Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3047–3054. IEEE, 2021. 2

[62] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross modal transformer: Towards fast and robust 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18268–18278, 2023. 2, 6

[63] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors (Basel, Switzerland)*, 18, 2018. 6

[64] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2

[65] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 2

[66] Zeyu Yang, Jiaqi Chen, Zhenwei Miao, Wei Li, Xiatian Zhu, and Li Zhang. Deepinteraction: 3d object detection via modality interaction. *Advances in Neural Information Processing Systems*, 35:1992–2005, 2022. 2, 6, 13

[67] 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. 2

[68] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. *Advances in Neural Information Processing Systems*, 34:16494–16507, 2021. 2

[69] Jinqing Zhang, Yanan Zhang, Qingjie Liu, and Yunhong Wang. Sa-bev: Generating semantic-aware bird's-eye-view feature for multi-view 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3348–3357, 2023. 2

[70] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 2, 5

[71] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection, 2019. 6

[72] Zhuofan Zong, Dongzhi Jiang, Guanglu Song, Zeyue Xue, Jingyong Su, Hongsheng Li, and Yu Liu. Temporal enhanced training of multi-view 3d object detector via historical object prediction. *arXiv preprint arXiv:2304.00967*, 2023. 2