Availability-aware Sensor Fusion via Unified Canonical Space

Dong-Hee Paek **Seung-Hyun Kong***
CCS Graduate School of Mobility
KAIST
{donghee.paek, skong}@kaist.ac.kr

Abstract

Sensor fusion of camera, LiDAR, and 4-dimensional (4D) Radar has brought a significant performance improvement in autonomous driving. However, there still exist fundamental challenges: deeply coupled fusion methods assume continuous sensor availability, making them vulnerable to sensor degradation and failure, whereas sensor-wise cross-attention fusion methods struggle with computational cost and unified feature representation. This paper presents availability-aware sensor fusion (ASF), a novel method that employs unified canonical projection (UCP) to enable consistency in all sensor features for fusion and cross-attention across sensors along patches (CASAP) to enhance robustness of sensor fusion against sensor degradation and failure. As a result, the proposed ASF shows a superior object detection performance to the existing state-of-the-art fusion methods under various weather and sensor degradation (or failure) conditions; Extensive experiments on the K-Radar dataset demonstrate that ASF achieves improvements of 9.7% in AP_{BEV} (87.2%) and 20.1% in AP_{3D} (73.6%) in object detection at IoU=0.5, while requiring a low computational cost. All codes are available at https://github.com/kaist-avelab/k-radar.

1 Introduction

Autonomous driving technology has advanced rapidly, with multiple companies adopting multi-sensor fusion approaches that combine two or more sensors, such as cameras, LiDAR, and 4-dimensional (4D) Radar, to achieve more robust and reliable perception (Badue et al., 2021; Wang et al., 2020). Cameras uniquely provide color information but struggle with depth estimation; LiDAR delivers high-resolution 3-dimensional (3D) point cloud data but is less reliable in adverse weather conditions (Zheng et al., 2023a); and 4D Radar, despite its relatively low angular resolution, offers robustness in adverse weather and directly measures relative velocity (Paek et al., 2022; Palffy et al., 2022; Kong et al., 2024; Sun and Zhang, 2021). This complementary nature initiated sensor fusion between camera, LiDAR, and 4D Radar to improve perception performance and reliability compared to a single-sensor (Yan et al., 2023; Liang et al., 2024; Liu et al., 2023a; Zheng et al., 2023b).

Most multi-modal sensor fusion methods can be categorized into two methods. The first is deeply coupled fusion (DCF), which directly combines feature maps (FMs) extracted by sensor-specific encoder, as illustrated in Fig. 1-(a). While it is simple to implement and computationally efficient with excellent performance in various benchmarks (Chae et al., 2024; Liu et al., 2023a; Liang et al., 2024; Zhao et al., 2024a; Caesar et al., 2020; Geiger et al., 2013), it assumes all sensors are functioning properly and consistently. This makes DCF vulnerable to sensor degradation due to adverse weathers, surface-damages, and sensor failure. Moreover, DCF requires retraining the entire neural network when the number of sensors changes, as the size of the fused FM (i.e., the input to the detection

*corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

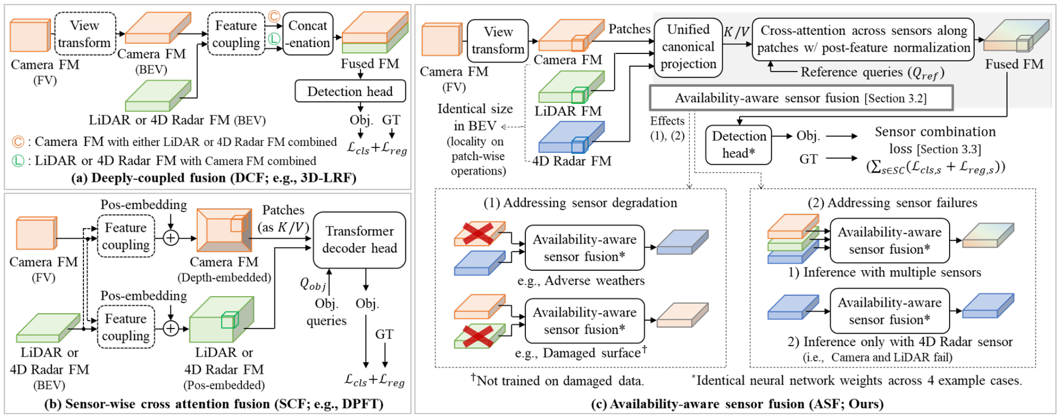
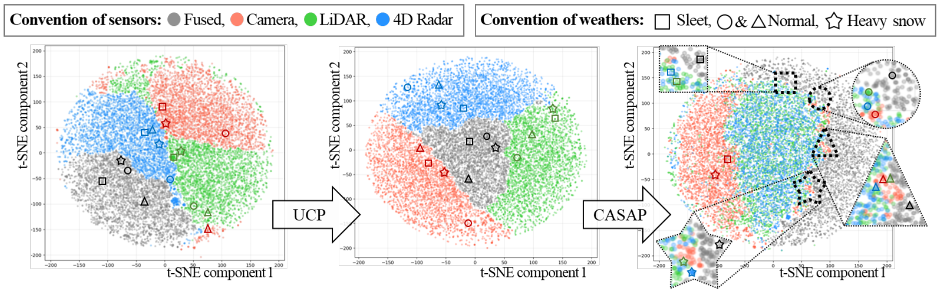


Figure 1: Comparison of sensor fusion methods: (a) DCF (e.g., 3D-LRF (Chae et al., 2024)), (b) SCF (e.g., CMT (Yan et al., 2023)), and (c) ASF. FV, BEV, Obj., GT, $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$ stand for 'front-view', 'bird's eye-view', 'objects', 'ground truths', 'classification loss', and 'regression loss', respectively. 'Feature coupling' refers to methods that combine features from multiple sensors to create new features. Optional components are in dashed lines; for example, (Vora et al., 2020) combines camera and LiDAR features without transforming the camera viewpoint, while (Yan et al., 2023) fuses features through a transformer decoder head (Carion et al., 2020) without explicit feature coupling. ASF does not apply feature coupling to ensure independence between sensors.

head) changes. The second method is sensor-wise cross-attention fusion (SCF), which divides features extracted from each sensor into patches with positional-embedding (e.g., depth information for camera (Liu et al., 2022; Wang et al., 2022)) and selectively combines available patches using cross-attention (Fig. 1-(b)), allowing it to handle cases where some sensors are degraded. However, SCF does not have sensor scalability, since the method does not project sensor-specific features into a standardized representation (Zheng et al., 2024; Liu et al., 2023b; Wang et al., 2016). In addition, SCF incurs computational complexity that scales with the number of patches, resulting in substantial computational overhead when processing multiple sensors with numerous patches (Fent et al., 2024; Yan et al., 2023; Bai et al., 2022).

One of the fundamental limitations of existing fusion methods stems from inconsistencies in feature representation across different sensors (Shashua, 2024; Yeong et al., 2025). Cameras produce 2D RGB images, whereas LiDAR generates 3D point clouds, and 4D Radar produces low-resolution tensors with power values. Therefore, each sensor extracts features with different representations for the same object, making consistent fusion challenging (as shown in Fig. 2-(a)). To address this inconsistency, an ideal strategy could project features from different sensors into a unified canonical space for fusion. The concept of 'True Redundancy (Shashua, 2024)', that ensures sensor independence while maintaining canonical feature representation for any sensor combination, suggests a promising direction for highly reliable and robust sensor fusion.

Therefore, we propose availability-aware sensor fusion (ASF) method (Fig. 1-(c)), in which each sensor performs independently while being complementarily fused through a projection to a unified canonical space. As a result, the proposed method addresses the limitations of both DCF and SCF simultaneously. The key innovation of ASF is in two sub-modules; First, unified canonical projection (UCP) projects features from each sensor into a unified space based on common criteria (i.e., canonical). Since UCP is optimized using the same reference query for all sensors, inconsistencies in sensor features are eliminated. While Fig. 2-(a) shows sensor features represented without clear patterns, Fig. 2-(b) demonstrates how UCP aligns the features from each sensor to the fused feature. Second, cross-attention across sensors along patches (CASAP) estimates the availability of sensors through patch-wise cross-attention on features projected into the unified canonical space, assigning higher weights to features from available sensors and lower weights to features from missing or degraded sensors. Unlike SCF that applies the cross-attention across all sensors (N_s) and patches (N_p) (i.e., $O(N_q N_s N_p)$ for N_q queries), ASF only applies the cross-attention across sensors along patches (i.e., $O(N_q N_s)$). Because of this, ASF eliminates complex positional-embedding and improves computationally efficiency. Additionally, it applies normalization to ensure that (camera, LiDAR,



(a) Output features of sensor-specific encoders (b) Output features of UCP (c) Output features of CASAP

Figure 2: Visualization of feature representation with t-SNE (van der Maaten and Hinton, 2008) at different stages of ASF for 'Sedan' class. Red, green, blue, and gray dots represent features from camera, LiDAR, 4D Radar, and fused features, respectively. Symbols in solid lines such as circle and triangle, square, and star indicate normal, sleet, and heavy snow conditions, respectively. (a) Initial output features from sensor-specific encoders show inconsistent distribution across sensors. (b) After unified canonical projection (UCP), features become better aligned to the fused feature. (c) After cross-attention across sensors along patches (CASAP), features from available sensors form cohesive clusters (in dashed symbols) based on weather conditions. Note that in adverse weather, camera features show larger deviation due to the degradation. Additional visualizations are in Appendix B.

and 4D Radar) features can be processed consistently by the detection head regardless of sensor combination. In contrast to Fig. 2-(b), where each sensor's features remain separate, Fig. 2-(c) shows how sensor features cluster together after CASAP (except for camera become useless in adverse weathers, as shown in Fig. 3). This enables ASF to flexibly handle sensor degradation or failure and to achieve the reliability embodied in the 'True Redundancy' concept (Shashua, 2024) for autonomous driving implementation.

To integrate the availability awareness into the detection head, we propose a sensor combination loss (SCL) that optimizes learning outcomes across all sensor combinations. SCL considers individual sensor unavailability during the training, enabling the system to maintain high performance in the presence of unexpected sensor failure or adverse weather conditions. The effectiveness of our proposed ASF method has been validated on the K-Radar dataset (Paek et al., 2022), demonstrating improvements of 9.7% in AP_{BEV} (87.2%) and 20.1% in AP_{3D} (73.6%) for detection performance at IoU=0.5 compared to state-of-the-art (SOTA) methods (Chae et al., 2024; Huang et al., 2025), which includes the performance in extreme situations such as sensor degradation or failure (i.e., unavailable).

The main contributions of this paper are as follows: (1) We propose ASF based on UCP and CASAP that achieves superior performance to SOTA methods and robust performance against sensor degradation and failure. (2) We propose SCL for the loss function to optimize the detection performance for all possible sensor combinations. (3) Through extensive experiments on the K-Radar dataset, we demonstrate that ASF achieves the high performance with low computational load.

The remainder of this paper is organized as follows: Section 2 introduces existing methods, and Section 3 describes the proposed ASF in detail. Section 4 presents experimental settings and results on the K-Radar dataset, analyzing performance in various scenarios including sensor degradation and failure. Section 5 concludes the paper with a summary and discusses the limitations. All codes and logs for ASF are available at `https://github.com/kaist-avelab/k-radar`.

2 Related Works

2.1 Deeply Coupled Fusion (DCF)

DCF constructs a fused feature map (FM) by concatenating FMs from each sensor. Most studies focus on fusing camera with LiDAR or 4D Radar (Liu et al., 2023a; Liang et al., 2024; Zheng et al., 2023b; Xiong et al., 2023), or combining LiDAR and 4D Radar (Chae et al., 2024; Huang et al., 2025). Implementations range from directly fusing the front-view (FV) camera image with LiDAR or 4D Radar without view transformation (Vora et al., 2020) to applying learnable BEV transforms

(Philion and Fidler, 2020) and concatenating at the BEV stage (Liu et al., 2023a; Liang et al., 2024; Zheng et al., 2023b). DCF method is straightforward to implement and computationally efficient compared to SCF, demonstrating strong performance across multiple benchmarks.

Recent DCF studies have improved performance by applying feature coupling, where the FMs of each sensor are enhanced with FMs of other sensors using multi-layer perception (MLP) or attention mechanisms (Vaswani et al., 2017). 3D-LRF (Chae et al., 2024) demonstrated superior performance to the conventional DCFs (Liang et al., 2024; Liu et al., 2023a) by applying attention between LiDAR and 4D Radar voxel features before concatenation. L4DR (Huang et al., 2025) achieved SOTA performance on the K-Radar dataset by weighting each sensor's FM using coupled BEV FMs with LiDAR and 4D Radar. However, DCF assumes all sensors are functioning normally, as they rely on the fused FM constructed by concatenating FMs from all sensors, making DCF vulnerable to sensor degradation and failure. This limitation arises because the training process does not expose the model to potential sensor degradation or failure scenarios that commonly occur during deployment.

2.2 Sensor-wise Cross-attention Fusion (SCF)

SCF divides sensor-specific FMs into patches and dynamically combines them through cross-attention in a transformer decoder head (Yan et al., 2023), inherently accommodating sensor availability. TransFusion (Bai et al., 2022) is the first SCF that addresses sensor availability, but its sequential fusion method (e.g., performing LiDAR detection first and then fusing with camera data) makes inference impossible when LiDAR is unavailable. CMT (Yan et al., 2023) presents an availability-aware sensor fusion of camera and LiDAR data using a transformer decoder head without applying feature coupling to individual sensors. However, without feature coupling for camera, CMT relies on positional embedding to incorporate depth information into camera FMs (Liu et al., 2022), which results in 3D patches ($H \times W \times D$). This leads to computational complexity of $O(N_q N_s N_p)$ (where N_q, N_s, and N_p are the number of queries, sensors, and patches, respectively), causing explosive growth in computational cost and memory usage. For instance, CMT requires 8 A100 GPUs with 80GB VRAM to train with a batch size of 16.

Recently, DPFT (Fent et al., 2024) creates independent FMs and projects sensor-agnostic query points onto different FMs to verify sensor availability, achieving 56.1% AP_{3D} at IoU=0.3 (using only 4D Radar and camera). Unlike methods that utilize entire FMs, DPFT achieves reasonable computational efficiency by employing deformable attention (Xia et al., 2022) that considers varying receptive fields using only a small number of key points. However, similar to CMT, DPFT performs object detection using variable object queries, which does not establish a common representation across different sensors (as illustrated in Fig. 2).

3 Proposed Methods

3.1 Sensor Fusion Framework

The overall sensor fusion framework consists of three stages: (1) sensor-specific encoders (i.e., backbones) that extract same-sized bird's eye-view (BEV) feature maps (FMs) from each sensor data (i.e., RGB image, LiDAR point cloud, and 4D Radar tensor), (2) the proposed availability-aware sensor fusion (ASF) network that is described in following subsection 3.2, and (3) a detection head that detects objects from the fused FM.

Focusing on our ASF contribution, we utilize established methods for the sensor-specific encoders and detection head. Specifically, we adopt BEVDepth (Li et al., 2023), SECOND (Yan et al., 2018), and RTNH (Paek et al., 2022) backbones for camera, LiDAR, and 4D Radar, respectively, along with a SSD detection head (Liu et al., 2016). Further specifications regarding the overall sensor fusion framework, such as sensor-specific encoders and detection head, are provided in Appendix A.

3.2 Availability-aware Sensor Fusion (ASF)

As illustrated in Fig. 1-(c), the proposed ASF consists of two key components: unified canonical projection (UCP) and cross-attention across sensors along patches (CASAP).

Unified Canonical Projection (UCP). One of the key challenge in multi-modal sensor fusion is the inherent inconsistency of features from different sensors (Shashua, 2024; Yeong et al., 2025), as visualized in Fig. 2-(a). To tackle this, we divide BEV FMs into an equal number of patches for all sensors and train projection functions to transform features from each sensor into a unified space based on the same criteria (i.e., reference query in CASAP). To formally define our methods, we first represent the same-sized FMs of each sensor as:

$$\mathbf{FM}^s \in \mathbb{R}^{C_s \times H \times W}, s \in \{S_C, S_L, S_R\}, \tag{1}$$

where C_s denotes the channel dimension for sensor s, H and W represent the identical height and width of BEV FMs for all sensors, respectively, and S_C, S_L, and S_R denote camera, LiDAR, and 4D Radar sensors, respectively. Each FM is then divided into patches $\mathbf{F}_p^s$ with height P_H and width P_W:

$$\mathcal{T}_p(\mathbf{FM}^s) = \{\mathbf{F}_{p,i}^s | \mathbf{F}_{p,i}^s \in \mathbb{R}^{C_s \times P_H \times P_W}, i = 1 : N_p\}, \tag{2}$$

where $\mathcal{T}_p(\cdot)$ is the operation that divides each FM into patches, $N_p = (H/P_H) \times (W/P_W)$ is the number of patches, which is identical across all sensors since each FM has the same spatial size, and '$1 : N_p$' denotes '$1, 2, ..., N_p$'. Note that since the patches are already spatially aligned (i.e., $\mathbf{F}_{p,i}^{S_C}$, $\mathbf{F}_{p,i}^{S_L}$, and $\mathbf{F}_{p,i}^{S_R}$ correspond to the same position), our method eliminates the use of computationally expensive positional-embedding (Liu et al., 2022) required for SCF (Yan et al., 2023; Fent et al., 2024). Then, we apply a parallel operation along patches that projects each patch to have the same channel dimension C_u. This is the UCP operation $\mathcal{U}^s(\cdot)$ that transforms sensor-specific patches into patches in a unified canonical space. The UCP-processed patch $\mathcal{P}_u^s$ for each sensor is expressed as:

$$\mathcal{P}_u^s = \{\mathbf{F}_{u,i}^s | \mathbf{F}_{u,i}^s = \mathcal{U}^s(\mathrm{LN}(\mathbf{F}_{p,i}^s)) \in \mathbb{R}^{C_u}, i = 1 : N_p\} \tag{3}$$

$$\mathcal{U}^s(\cdot) = \mathrm{LN}(\mathrm{Proj}^{(n_u)}(\cdot)), \mathrm{Proj}(\cdot) = \mathrm{GeLU}(\mathrm{MLP}(\cdot)), \tag{4}$$

where LN and n_u denote layer normalization (Ba et al., 2016) for training stability and the number of sequential projection functions incorporating MLP for transformation and GeLU (Hendrycks and Gimpel, 2016) for non-linearity, respectively. While our framework allows for repetition of the projection function to increase non-linearity, with 1 or 2 repetitions being sufficient (we use $n_u = 2$, which aligns features as demonstrated in Fig. 2-(b)). Note that $\mathcal{U}^s$ is trained separately for each sensor based on reference query, which results in alignment of features from all sensors with respect to the fused feature as shown in Fig. 2-(b).

Cross-attention Across Sensors Along Patches (CASAP). The patches $\mathbf{F}_u^s$ projected into the unified canonical space by UCP serve as keys (K) and values (V) for a trainable reference query $\mathbf{Q}_{ref} \in \mathbb{R}^{N_q \times C_u}$ (where N_q is the number of queries Q), and we perform cross-attention across sensors along patches as:

$$\mathbf{Q}'_{ref,i} = \mathrm{CrossAttn}(Q = \mathbf{Q}_{ref}, K\&V \in \{\mathbf{F}_{u,i}^{S_C}, \mathbf{F}_{u,i}^{S_L}, \mathbf{F}_{u,i}^{S_R}\}), i = 1 : N_p, \tag{5}$$

where $\mathbf{Q}'_{ref,i}$ is the output of the cross-attention applied across sensors for the i-th patch. Since $\mathbf{Q}_{ref}$ is trained primarily on features that are mostly available in the training data, it naturally develops high correlation (i.e., high attention scores) with patches from available sensors after the training. Consequently, during inference, $\mathbf{Q}'_{ref,i}$ is predominantly composed of available $\mathbf{F}_u^s$. The number of heads in cross-attention is a hyper-parameter whose impact is analyzed in subsection 4.4.

Compared to ASF, cross-attention in SCF is performed across all patches with respect to object queries $\mathbf{Q}_{obj}$ in the transformer decoder head. This can be mathematically expressed as:

$$\mathbf{Q}'_{obj} = \mathrm{CrossAttn}^{(n_{td})}(Q = \mathbf{Q}_{obj}, K\&V \in \{\mathbf{F}_{pe,1}^{S_C}, ..., \mathbf{F}_{pe,N_p}^{S_C}, \mathbf{F}_{pe,1}^{S_L}, ..., \mathbf{F}_{pe,N_p}^{S_L}, \mathbf{F}_{pe,1}^{S_R}, ..., \mathbf{F}_{pe,N_p}^{S_R}\}), \tag{6}$$

where $\mathbf{F}_{pe}^s$ represents patches with positional-embedding, and n_{td} (usually larger than 6 (Liu et al., 2022; Yan et al., 2023)) is the number of stacked transformer decoders. Eq. 6 shows the cross-attention across all sensors and all patches (i.e., the $K\&V$ set contains $N_s N_p$ patches, resulting in $O(N_q N_s N_p)$ computational complexity for N_q queries). In contrast, in Eq. 5, the cross-attention is applied across sensors and along patches, which requires only $O(N_q N_s)$ computational operations. This is a significant computational costs reduction as $N_s \ll N_p$. Moreover, as demonstrated in Fig. 3 and Tab. 1, ASF achieves better performance with only a single cross-attention layer than SCF utilizing stacked cross-attention layers (i.e., $n_{td} \geq 6$).

Sequentially, ASF applies post-feature normalization (PN) $\mathcal{N}$ that has a similar structure to $\mathcal{U}^s(\cdot)$ with LN, to ensure that features can be processed consistently by the detection head regardless of

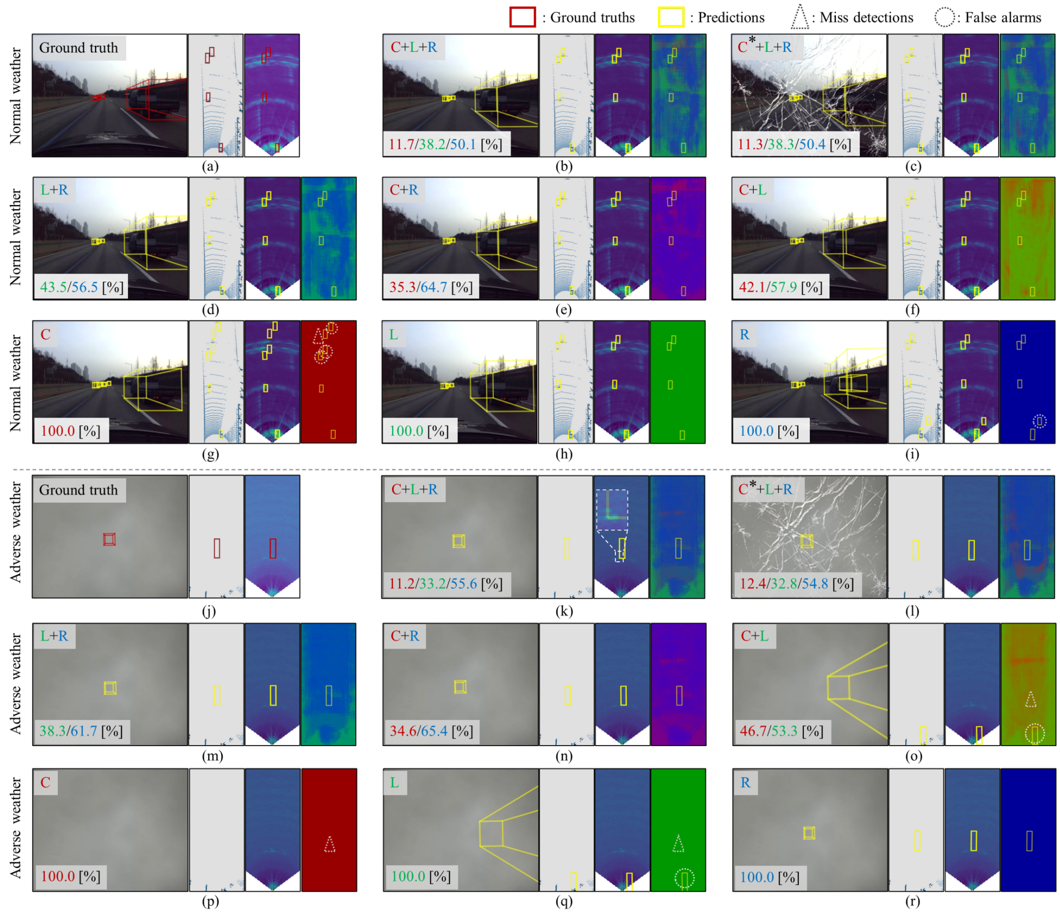


Figure 3: Qualitative results of ASF for various sensor combinations. We show results for normal and adverse weather conditions in (a-i) and (j-r), respectively, where employed sensors are noted in the top-left corner (C: Camera, L: LiDAR, R: 4D Radar, C*: damaged camera). Each subplot visualizes front-view camera image, LiDAR point cloud, 4D Radar tensor, and a sensor attention map (SAM) showing attention score distribution from cross-attention in CASAP. In the SAMs, red, green, and blue represent attention scores for Camera, LiDAR, and 4D Radar, respectively. For example, a predominantly blue SAM indicates that 4D Radar receives the highest attention scores, meaning that 4D Radar is used primarily for detection in the scene. The bottom-left corner of each subplot shows the proportion of attention scores in colored percentages (C/L/R[%]). Note that predictions are visualized on all sensor data, even when a sensor is not employed for detection (e.g., predictions from L+R are also visualized on the camera image).

the sensor combination. Therefore, PN enables camera, LiDAR, and 4D Radar features for the same object to be consistent. The set of patches $\mathcal{P}_n$ with PN is formulated as:

$$\mathcal{P}_n = \{\mathbf{F}_{n,i} | \mathbf{F}_{n,i} = \mathcal{N}(\mathrm{LN}(\mathbf{Q}'_{ref,i})) \in \mathbb{R}^{C_u}, i = 1 : N_p\} \tag{7}$$

$$\mathcal{N}(\cdot) = \mathrm{LN}(\mathrm{Proj}^{(n_n)}(\cdot)), \mathrm{Proj}(\cdot) = \mathrm{GeLU}(\mathrm{MLP}(\cdot)), \tag{8}$$

where n_n denotes the number of sequential projections, with 1 or 2 being sufficient to increase non-linearity. Unlike Fig. 2-(b) where features from different sensors occupy distinct regions, Fig. 2-(c) illustrates how PN causes sensor features to converge into unified clusters.

Finally, to transform $\mathcal{P}_n$ back to the original BEV size $(H \times W)$, we apply a reshape operation $\mathcal{T}_n(\cdot)$. Since the size of $\mathcal{P}_n$ is $C_u \times N_p = C_u \times N_H \times N_W = C_u \times (H/P_H) \times (W/P_W)$, the fused FM $\mathbf{FM}_{\text{fused}}$ can be obtained as:

$$\mathbf{FM}_{\text{fused}} = \mathcal{T}_n(\mathcal{P}_n \in \mathbb{R}^{C_u \times N_p}) \in \mathbb{R}^{C_q \times H \times W}, \tag{9}$$

Table 1: Performance comparison of 3D object detection on K-Radar (Paek et al., 2022) benchmark v1.0. C, L, and R represent Camera, LiDAR, and 4D Radar, respectively. The **bold** and underlined values indicate the best and the second-best, respectively. Note that the two ASF models for sensors L+R and C+L+R share the same neural network weights trained with C+L+R, which means that ASF for L+R represents a scenario where camera becomes unavailable. Nor., Ove., Sle., L.s., and H.s. denote 'Normal', 'Overcast', 'Sleet', 'Light snow', and 'Heavy snow', respectively.

Methods	Sensors	IoU	Metric	Total	Nor.	Ove.	Fog	Rain	Sle.	L.s.	H.s.
RTNH (Paek et al.)	R	0.3	BEV	41.1	41.0	44.6	45.4	32.9	50.6	81.5	56.3
			3D	37.4	37.6	42.0	41.2	29.2	49.1	63.9	43.1
		0.5	BEV	36.0	35.8	41.9	44.8	30.2	34.5	63.9	55.1
			3D	14.1	19.7	20.5	15.9	13.0	13.5	21.0	6.36
RTNH (Paek et al.)	L	0.3	BEV	76.5	76.5	88.2	86.3	77.3	55.3	81.1	59.5
			3D	72.7	73.1	76.5	84.8	64.5	53.4	80.3	52.9
		0.5	BEV	66.3	65.4	87.4	83.8	73.7	48.8	78.5	48.1
			3D	37.8	39.8	46.3	59.8	28.2	31.4	50.7	24.6
3D-LRF (Chae et al.)	L+R	0.3	BEV	84.0	83.7	89.2	95.4	78.3	60.7	88.9	74.9
			3D	74.8	81.2	87.2	86.1	73.8	49.5	87.9	67.2
		0.5	BEV	73.6	72.3	88.4	86.6	76.6	47.5	79.6	64.1
			3D	45.2	45.3	55.8	51.8	38.3	23.4	60.2	36.9
L4DR (Huang et al.)	L+R	0.3	BEV	79.5	86.0	89.6	89.9	81.1	62.3	89.1	61.3
			3D	78.0	77.7	80.0	88.6	79.2	60.1	78.9	51.9
		0.5	BEV	77.5	76.8	88.6	89.7	78.2	59.3	80.9	53.8
			3D	53.5	53.0	64.1	73.2	53.8	46.2	52.4	37.0
ASF (Proposed)	L+R	0.3	BEV	88.6	88.1	**90.3**	**99.0**	**89.1**	80.4	**89.4**	**78.7**
			3D	87.3	86.6	89.8	90.7	**88.6**	80.0	**88.8**	**77.5**
		0.5	BEV	87.0	86.2	**90.2**	90.8	**88.8**	78.2	**88.6**	**71.0**
			3D	72.9	64.6	86.6	**79.6**	**73.4**	**70.0**	77.6	**66.7**
	C+L+R	0.3	BEV	**88.6**	**88.2**	90.2	98.9	89.0	**80.4**	89.2	78.4
			3D	**87.4**	**87.0**	**90.1**	**90.7**	88.2	**80.0**	88.6	77.4
		0.5	BEV	**87.2**	**86.7**	90.1	**90.8**	88.7	**78.3**	88.3	70.9
			3D	**73.6**	**71.8**	**87.0**	79.4	73.0	67.5	**78.0**	66.4

where C_q is the quotient of $C_u/(P_H \times P_W)$ as we design $C_u = P_H \times P_W \times C_q$. Since the resulting channel dimension C_q may be insufficient for containing feature representation due to reduced channel dimensions after reshaping, in the implementation, we increase the number of patches by a factor of n_p (i.e., $N_p = N_H \times N_W \rightarrow N_p = n_p \times N_H \times N_W$). Consequently, the channel dimension of $\mathbf{FM}_{\text{fused}}$ increases from C_q to $n_p \times C_q$, and impact of this modification is evaluated in subsection 4.4.

3.3 Sensor Combination Loss (SCL)

Leveraging the consistent size of $\mathbf{FM}_{\text{fused}}$ (which serves as the input to the detection head) regardless of sensor combinations, we propose an SCL that enables simultaneous optimization across multiple sensor configurations. The proposed SCL is formalized as:

$$\mathcal{L}_{SCL} = \sum_{s \in \mathcal{SC}} (\mathcal{L}_{cls,s} + \mathcal{L}_{reg,s}), \tag{10}$$

where $\mathcal{SC}$ represents the set of 7 possible sensor combinations (S_C-only, S_L-only, S_R-only, S_L+S_R, S_C+S_R, S_C+S_L, and $S_C+S_L+S_R$), where S_C, S_L, and S_R denote camera, LiDAR, and 4D Radar, respectively. $\mathcal{L}_{cls,s}$ and $\mathcal{L}_{reg,s}$ are the classification and regression losses for each sensor combination. SCL explicitly prepares for the potential sensor unavailability by optimizing across all sensor combinations in the training, enabling the model to recognize that available sensors perform better than others (e.g., 4D Radar outperform camera in adverse weather). As demonstrated in Tab. 4, SCL enhances the performance of the proposed ASF method when compared to ASF without SCL.

Table 2: Comparison of VRAM and FPS evaluated on the K-Radar benchmark v1.0. The **bold** and underlined values indicate the best and the second-best, respectively. The unit of VRAM and FPS are GB and Hz, respectively.

Methods	Sensors	VRAM	FPS
3D-LRF (Chae et al.)	L+R	**1.2**	5.04
DPFT (Fent et al.)	C+R	4.0	11.5
ASF (Proposed)	L+R	1.5	**20.5**
	C+L+R	1.6	13.5

4 Experiments

4.1 Experimental Setup

Dataset and Metrics. K-Radar (Paek et al., 2022) is a large-scale autonomous driving dataset with a broad range of conditions including time (day, night), weather (normal, rain, fog, snow, sleet), road types (urban, highway, mountain), and sensors (4D Radar, LiDAR, camera, GPS). Notably, K-Radar is the only dataset with data captured in adverse weather conditions.

For comparison with SOTA methods, we utilize two K-Radar benchmark variants. Benchmark v1.0 (Paek et al., 2022; Chae et al., 2024; Huang et al., 2025) focuses on the 'Sedan' class within a driving corridor region of [0m, 72m] × [-6.4m, 6.4m] × [-2m, 6m] (X×Y×Z). For ablation studies and qualitative analysis, we use benchmark v2.0, which covers a wider area [0m, 72m] × [-16m, 16m] × [-2m, 7.6m] and includes both 'Sedan' and 'Bus or Truck' classes. We evaluate using AP_{3D} and AP_{BEV} at IoU thresholds of 0.3 and 0.5, while also reporting VRAM usage and FPS based on the same hardware setup.

Implementation Details. We implement the ASF on a single RTX3090 GPU with 24GB VRAM. ASF is trained for 11 epochs using AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate 0.001 and a batch size 2. The voxel size for the fused FM is set to 0.4m, consistent with (Paek et al., 2022).

4.2 Comparison of ASF to SOTA Methods

Following the benchmark v1.0 of the K-Radar (Paek et al., 2022), we compare the proposed ASF with SOTA methods including 3D-LRF (Chae et al., 2024) and L4DR (Huang et al., 2025), and we use RTNH (Paek et al., 2022) for single-sensor performance. In addition to the detection performance comparison with DCF methods (Chae et al., 2024; Huang et al., 2025), we evaluate computational efficiency against DPFT (Fent et al., 2024) which is the only open-sourced SCF method available for K-Radar.

Table 3: Performance comparison of ASF under various sensor combinations on K-Radar (Paek et al., 2022) benchmark v2.0. We indicate the employed sensors (C: Camera, L: LiDAR, R: 4D Radar) and report AP_{3D} at IoU = 0.3 for 'Sedan' and 'Bus or Truck' classes. 'Nor.', 'Ove.', 'Sle.' and 'H.s.' refer to 'Normal', 'Overcast', 'Sleet', and 'Heavy snow', respectively. $*$ denotes the sensor unavailability, as shown in Fig. 3. The **bold** and underlined values indicate the best and the second-best, respectively. All ten ASF models share the same neural network weights trained for R+L+C. Performance under additional weather conditions (Fog, Rain, and Light Snow) and with other evaluation metric AP_{BEV} is provided in Appendix D.

Class	Sensors	Total	Nor.	Ove.	Sle.	H.s.
Sedan	R	47.3	40.7	58.8	45.9	56.5
	L	73.0	73.0	86.1	64.9	54.5
	C	14.8	14.9	7.71	-	-
	C*	3.71	3.72	3.17	-	-
	L+R	77.3	77.7	87.3	74.4	65.4
	C+R	52.7	49.1	62.4	46.0	57.2
	C+L	76.4	78.3	86.5	64.2	57.1
	C+L+R	**79.3**	**78.8**	87.6	74.2	**65.8**
	C*+L+R	77.6	78.2	**87.7**	**74.4**	65.4
	C+L*+R	58.9	58.8	66.6	52.3	58.2
Bus or Truck	R	34.2	22.9	40.9	21.1	51.2
	L	54.9	**53.7**	74.8	69.1	37.8
	C	9.59	9.02	17.2	-	-
	C*	3.65	3.66	0.00	-	-
	L+R	59.9	52.5	71.6	68.2	68.9
	C+R	36.2	24.4	41.4	23.4	56.5
	C+L	53.0	49.1	60.1	**72.1**	39.6
	C+L+R	**60.4**	52.7	**77.4**	69.2	68.9
	C*+L+R	60.1	52.1	72.0	70.9	**69.1**
	C+L*+R	40.0	31.5	38.2	28.9	54.8

Detection Performance. As shown in Tab. 1, ASF significantly outperforms SOTA methods across various weather conditions. Compared to previous SOTA L4DR (Huang et al., 2025), ASF achieves substantial improvements of 9.7% in AP_{BEV} (87.2% vs. 77.5%) and 20.1% in AP_{3D} (73.6% vs. 53.5%) at IoU=0.5. These improvements are particularly remarkable in challenging conditions like sleet (67.5% vs. 46.2% AP_{3D}) and heavy snow (66.4% vs. 37.0% AP_{3D}). Notably, both ASF configurations (L+R and C+L+R) use identical neural network weights yet maintain comparable performance, demonstrating the system's ability to gracefully handle sensor degradation. Even with only LiDAR and Radar, ASF achieves 87.0% AP_{BEV} and 72.9% AP_{3D} at IoU=0.5, nearly matching the full sensor suite's performance.

Computational Efficiency. ASF achieves exceptional computational efficiency for real-time autonomous driving applications. As demonstrated in Tab. 2, ASF with LiDAR and 4D Radar processes at 20.5 Hz, approximately 4× faster than 3D-LRF (5.04 Hz) using identical sensors. Even with all three sensors, ASF maintains 13.5 Hz, exceeding the 10 Hz threshold typically required for autonomous driving systems (Zhao et al., 2024b). This efficiency results from our CASAP, which applies cross-attention across sensors along patches rather than across all sensors and patches. Furthermore, ASF maintains a compact memory footprint (1.5-1.6 GB), comparable to 3D-LRF (1.2 GB) and substantially lower than DPFT (4.0 GB).

4.3 Addressing Sensor Degradation and Failure

A key advantage of ASF is the robust performance under sensor degradation or failure. As shown in Tab. 3 and Fig. 3, ASF dynamically adapts to different sensor combinations without retraining. Under normal conditions (Fig. 3-(a-i)), ASF effectively utilizes all available sensors with attention weights distributed according to each sensor's reliability. However, ASF's true value emerges in challenging scenarios. In adverse weather (Fig. 3-(j-r)), camera and LiDAR measurements are significantly degraded or disappear completely. In these critical situations, ASF automatically redistributes attention toward the more reliable 4D Radar, as evidenced by the predominant blue coloration in the sensor attention maps (SAMs) and corresponding attention percentages. Even with damaged sensors (denoted by * in Tab. 3 and Fig. 3), ASF maintains near-optimal performance. For example, with a damaged camera (C*), C*+L+R shows 77.6% AP_{3D}, which is only 1.7% lower than with fully sensors (79.3%). This robustness stems from the unified canonical projection (which creates a common feature space) and the cross-attention mechanism (which estimates sensor reliability).

The qualitative results in Fig. 3 demonstrate that in adverse weather, when LiDAR measurements disappear and camera visibility severely degrades, reliable object detection is only possible with active 4D Radar and ASF is fully using 4D Radar. Note that all results shown are from the same ASF model with identical weights, illustrating how ASF dynamically adjusts attention to maintain detection performance across varying sensor availabilities.

4.4 Ablation Studies

Tab. 4 presents ablation studies of key ASF components, analyzing five factors: patch size (P), channel dimension (C_u), patches multiplier (n_p), number of attention heads (n_h), and sensor combination loss (SCL). Our findings reveal that smaller patch sizes (P=2) improve performance through finer feature extraction, while balancing reduced channel dimension (C_u=256) with increased patches multiplier (n_p=8) maintains or enhances results; furthermore, increasing attention heads (n_h = 16) benefits 'Bus or Truck' detection, and incorporating SCL consistently improves performance across configurations by enhancing robustness to varying sensor availability. The optimal configuration combines P=2, C_u=256, n_p=8, n_h=16 with SCL.

Table 4: Ablation study of ASF. ASF performance for different components and parameters: P (patch size), C_u (channel dimension in unified canonical space), n_p (number of patches multiplier), n_h (number of heads in CASAP) and SCL, using AP_{3D} at IoU=0.3 for both 'Sedan' and 'Bus or Truck' on the K-Radar benchmark v2.0.

Exp.	P	C_u	n_p	n_h	SCL	Sedan	Bus
(a)	5	512	1	8		76.1	45.7
(b)	5	512	4	8		76.4	47.4
(c)	2	512	4	8		77.2	49.7
(d)	2	256	8	8		77.6	57.9
(e)	2	256	8	8	✓	79.3	58.2
(f)	2	256	8	16		77.5	60.2
(g)	2	256	8	16	✓	79.3	60.4

5 Conclusion

This paper introduces availability-aware sensor fusion (ASF), which addresses sensor availability challenges in autonomous driving by transforming features into a unified canonical space through UCP and CASAP. Our approach maintains computational efficiency ($O(N_q N_s)$) while providing robust fusion for sensor degradation or failure. The proposed sensor combination loss further enhances robustness by optimizing across all possible sensor combinations. Experiments on the K-Radar dataset demonstrate significant improvements over SOTA methods (9.7% in AP_{BEV} and

20.1% in AP_{3D} at IoU=0.5), with consistent performance across various weather conditions and sensor combinations.

Limitations. Despite ASF's strong performance, the camera network's capabilities remain a limitation. As shown in Fig. 3-(g) and (p), camera-based object detection is less precise, particularly in adverse weather. In Fig. 2-(c), while LiDAR and 4D Radar features are well integrated, camera features remain more separated in feature space. Enhancing the camera backbone could further boost system performance, especially in favorable weather conditions, where visual information is valuable.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C3008370).

References

Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert systems with applications*, 165:113816, 2021.

Zhangjing Wang, Yu Wu, and Qingqing Niu. Multi-sensor fusion in automated driving: A survey. *IEEE Access*, 8:2847–2868, 2020. doi: 10.1109/ACCESS.2019.2962554.

Ziqiang Zheng, Yujie Cheng, Zhichao Xin, Zhibin Yu, and Bing Zheng. Robust perception under adverse conditions for autonomous driving based on data augmentation. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):13916–13929, 2023a. doi: 10.1109/TITS.2023.3297318.

Dong-Hee Paek, Seung-Hyun Kong, and Kevin Tirta Wijaya. K-radar: 4d radar object detection for autonomous driving in various weather conditions. *Advances in Neural Information Processing Systems*, 35:3819–3829, 2022.

Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian F. P. Kooij, and Dariu M. Gavrila. Multi-class road user detection with 3+1d radar in the view-of-delft dataset. *IEEE Robotics and Automation Letters*, 7(2):4961–4968, 2022. doi: 10.1109/LRA.2022.3147324.

Seung-Hyun Kong, Dong-Hee Paek, and Sangyeong Lee. Rtnh+: Enhanced 4d radar object detection network using two-level preprocessing and vertical encoding. *IEEE Transactions on Intelligent Vehicles*, pages 1–14, 2024. doi: 10.1109/TIV.2024.3428696.

Shunqiao Sun and Yimin D. Zhang. 4d automotive radar sensing for autonomous vehicles: A sparsity-oriented approach. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):879–891, 2021. doi: 10.1109/JSTSP.2021.3079626.

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.

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. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088.

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, 2023a.

Lianqing Zheng, Sen Li, Bin Tan, Long Yang, Sihan Chen, Libo Huang, Jie Bai, Xichan Zhu, and Zhixiong Ma. Rcfusion: Fusing 4-d radar and camera with bird's-eye view features for 3-d object detection. *IEEE Transactions on Instrumentation and Measurement*, 72:1–14, 2023b. doi: 10.1109/TIM.2023.3280525.

Yujeong Chae, Hyeonseong Kim, and Kuk-Jin Yoon. Towards robust 3d object detection with lidar and 4d radar fusion in various weather conditions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15162–15172, 2024. doi: 10.1109/CVPR52733.2024.01436.

Yun Zhao, Zhan Gong, Peiru Zheng, Hong Zhu, and Shaohua Wu. Simplebev: Improved lidar-camera fusion architecture for 3d object detection. *arXiv preprint arXiv:2411.05292*, 2024a.

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.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

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.

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

Xu Zheng, Yuanhuiyi Lyu, and Lin Wang. Learning modality-agnostic representation for semantic segmentation from any modalities. In *European Conference on Computer Vision*, pages 146–165. Springer, 2024.

Zhenghao Liu, Chenyan Xiong, Yuanhuiyi Lv, Zhiyuan Liu, and Ge Yu. Universal vision-language dense retrieval: Learning a unified representation space for multi-modal retrieval. In *International Conference on Learning Representations (ICLR)*, 2023b.

Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013, 2016.

Felix Fent, Andras Palffy, and Holger Caesar. DPFT: Dual perspective fusion transformer for camera-radar-based object detection. *IEEE Transactions on Intelligent Vehicles (T-IV)*, 2024.

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.

Amnon Shashua. True Redundancy, 2024. URL `https://www.mobileye.com/technology/true-redundancy/`.

De Jong Yeong, Krishna Panduru, and Joseph Walsh. Exploring the unseen: A survey of multi-sensor fusion and the role of explainable ai (xai) in autonomous vehicles. *Sensors*, 25(3):856, 2025. doi: 10.3390/s25030856.

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.

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

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Xun Huang, Ziyu Xu, Hai Wu, Jinlong Wang, Qiming Xia, Yan Xia, Jonathan Li, Kyle Gao, Chenglu Wen, and Cheng Wang. L4dr: Lidar-4dradar fusion for weather-robust 3d object detection. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

Weiyi Xiong, Jianan Liu, Tao Huang, Qing-Long Han, Yuxuan Xia, and Bing Zhu. Lxl: Lidar excluded lean 3d object detection with 4d imaging radar and camera fusion. *IEEE Transactions on Intelligent Vehicles*, 2023.

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.

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.

Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4794–4803, 2022.

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*, volume 37, pages 1477–1485, 2023.

Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. In *Sensors*, volume 18, page 3337. MDPI, 2018.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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

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

Jingyuan Zhao, Wenyi Zhao, Bo Deng, Zhenghong Wang, Feng Zhang, Wenxiang Zheng, Wanke Cao, Jinrui Nan, Yubo Lian, and Andrew F. Burke. Autonomous driving system: A comprehensive survey. *Expert Systems with Applications*, 242:122836, 2024b. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2023.122836.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's key contributions (ASF with UCP and CASAP, sensor combination loss).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper explicitly addresses limitations in conclusion, noting that camera network capabilities remain a limitation, particularly in adverse weather conditions, and acknowledging that camera features are less well-integrated than LiDAR and 4D Radar features in the unified feature space.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper provides complete mathematical formulations with clearly stated assumptions for all components of the ASF (UCP, CASAP). Specifically, the computational complexity analysis ($O(N_q N_s)$ vs. $O(N_q N_s N_p)$) is mathematically justified in subsection 3.2, with all relevant variables defined and assumptions explicitly stated.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides comprehensive details for reproducing the experimental results, including dataset specifications (K-Radar benchmarks v1.0 and v2.0), implementation details (GPU, epochs, optimizer, learning rate, batch size, voxel size), network architecture specifics, and evaluation metrics. Additionally, the authors explicitly state that the code will be made publicly available after the review process.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The K-Radar dataset referenced in the paper is a widely used dataset and benchmark in autonomous driving research, making it accessible to researchers in the field. Additionally, the authors explicitly state that the code will be made publicly available following the completion of the review process, ensuring full reproducibility.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides comprehensive experimental details in subsection 4.1, specifying dataset variants (benchmark v1.0 and v2.0), evaluation metrics, implementation hardware (RTX3090 GPU), training parameters (11 epochs, AdamW optimizer, learning rate 0.001, batch size 2), and architectural specifics.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper provides suitable information about statistical significance through comprehensive performance breakdowns across different weather conditions and sensor combinations in Tab. 1 and 3, demonstrating the consistency of results. Additionally, the authors have included experiments in the Appendix that specifically analyze the impact of random seed variations, confirming the statistical reliability of the reported performance improvements.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the computing resources used, including hardware (single RTX3090 GPU with 24GB VRAM) and provides computational efficiency metrics (VRAM usage and FPS) in Tab. 2. Training details including epochs, batch size, and optimizer settings are clearly stated, giving sufficient information to estimate resource requirements for reproduction.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS Code of Ethics by focusing on technical advancements in sensor fusion without involving human subjects or sensitive data.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses broader impacts in introduction, highlighting the positive societal implications of improved autonomous driving safety in adverse weather conditions. It addresses how the proposed ASF enhances reliability when sensors are degraded or compromised, which directly impacts safety during challenging environmental conditions.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not pose high risks for misuse as it focuses on a technical approach for sensor fusion in autonomous driving systems. It does not release pretrained language models, image generators, or scraped datasets that would require special safeguards against potential misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly cites the K-Radar dataset when referencing it for experiments. While the license (CC BY-NC-ND) is not explicitly mentioned in the paper itself, this information is publicly available on the official K-Radar release page, and the authors' usage complies with the terms of this license.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces a new ASF method with comprehensive documentation of the architecture, implementation details, and experimental results. The authors state that complete code will be made publicly available after the review process with appropriate documentation to ensure reproducibility of all experiments and findings.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: The paper focuses on sensor fusion technology for vehicle detection without conducting research with human subjects. While the K-Radar dataset contains pedestrians with faces blurred for privacy protection (as noted in the dataset documentation), this privacy measure was implemented by the dataset creators rather than being part of the research methodology presented in this paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve any research with human subjects. The study focuses exclusively on technical methods for sensor fusion in autonomous driving systems using the K-Radar dataset, where any human data (such as pedestrian images) was previously collected and anonymized through face blurring by the dataset creators.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were only used for editing purposes such as grammar checking, spelling corrections, and word choice refinements in the manuscript. This editorial assistance does not impact the core methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.