BEV-ConvFusion: An Efficient 2D Fusion Framework for Real-Time Autonomous Perception

Kyuan Oh*, Jaeyoung Yang*, Minseong Kang*, Kiseong Lee†
Department of Artificial Intelligence, AI Humanities Research Institute†*
Chung-Ang University
Seoul, Republic of Korea
oka04108@cau.ac.kr, youngyj0928@cau.ac.kr, rkdalstjd0413@cau.ac.kr, goory@cau.ac.kr

***Abstract*—Autonomous driving requires perception systems that balance accuracy with computational efficiency for safe and reliable real-time operation. While camera–LiDAR fusion has emerged as a powerful solution, most existing methods rely on computationally expensive 3D backbones, limiting deployment on resource-constrained vehicle hardware. We propose BEV-ConvFusion, a novel 2D-domain fusion framework that overcomes this limitation. Our approach first encodes sparse LiDAR point clouds into dense multi-channel Bird's-Eye View (BEV) representations and extracts semantically rich features from RGB images using a 2D CNN backbone. At the core of our design is the Synergistic Cross-Attention Module (SynCAM), which refines features through three sequential stages: spatial gating, bidirectional semantic cross-attention, and feature refinement, enabling reciprocal enhancement between modalities before fusion. By eliminating 3D operations, BEV-ConvFusion achieves substantial computational savings while maintaining high accuracy. Extensive experiments demonstrate that our method achieves competitive detection accuracy, significantly higher frame rates, and superior robustness compared to unimodal baselines, highlighting the effectiveness of 2D-domain fusion.**

***Index Terms*—Multimodal Representation Learning, Autonomous Driving, Cross-modal Attention, LiDAR–Camera Fusion, Real-time Perception, Adverse Weather Robustness**

I. INTRODUCTION

The pursuit of fully autonomous driving (AD) requires perception systems that can interpret complex, dynamic environments with both high accuracy and strict real-time efficiency [1]. Multimodal sensor fusion—particularly the integration of LiDAR and cameras—has become a cornerstone of modern AD perception stacks. Cameras provide dense semantic cues such as texture, color, and object categories, but are notoriously sensitive to illumination changes and adverse weather. Conversely, LiDAR sensors offer robust, illumination-invariant geometric information, yet generate sparse point clouds that lack semantic richness. Effective perception thus hinges on bridging these complementary modalities [2].

A central challenge arises from their vastly different representations: dense 2D pixel arrays versus sparse, irregular 3D point sets. Recent fusion methods typically employ specialized 3D deep learning backbones, such as sparse 3D convolutions or voxel-based networks, to process LiDAR data before fusion. While powerful, these models impose prohibitive computational demands, making them impractical for embedded automotive hardware that must operate under strict real-time constraints required for safe vehicle deployment. This computational bottleneck raises the critical question of how to exploit the geometric strengths of LiDAR and the semantic richness of images while remaining compatible with real-time, resource-constrained deployment?

To address this, we propose **BEV-ConvFusion**, a fully 2D fusion framework that avoids costly 3D operations. Our key insight is to project LiDAR point clouds into a dense Bird's-Eye View (BEV) representation, encoding height, density, and intensity in a multi-channel 2D grid. This representation allows us to leverage efficient 2D convolutional networks (CNNs), which are highly optimized on modern hardware, for both modalities. To enable effective interaction between the modalities, we introduce the **Synergistic Cross-Attention Module (SynCAM)**,a novel fusion block that refines features in three stages: spatial gating, bidirectional semantic cross-attention, and feature refinement, enabling robust bidirectional interaction between image and LiDAR streams. Unlike conventional concatenation or single-step attention mechanisms, SynCAM explicitly models complementary cues, allowing LiDAR geometry to sharpen camera semantics and vice versa.

The contributions of this work are threefold:

- **A real-time multimodal fusion architecture** that operates entirely in the 2D domain, reducing computation by an order of magnitude compared to 3D backbones, while maintaining competitive accuracy.
- **A rich BEV representation for LiDAR** that encodes geometric structure in a dense, learnable form well-suited to standard 2D CNNs.
- **The Synergistic Cross-Attention Module (SynCAM),** a novel mechanism for cross-modal refinement that enables camera and LiDAR features to enhance one another prior to fusion, achieving superior robustness under challenging conditions.

Together, these innovations provide a practical path toward real-time multimodal perception for autonomous driving, bridging the gap between high-performance fusion and deployability on resource-constrained platforms.

II. RELATED WORK

A. LiDAR-Camera Fusion Strategies

Fusion strategies can be categorized by the stage of integration [3]. **Early fusion** combines raw inputs or shallow features

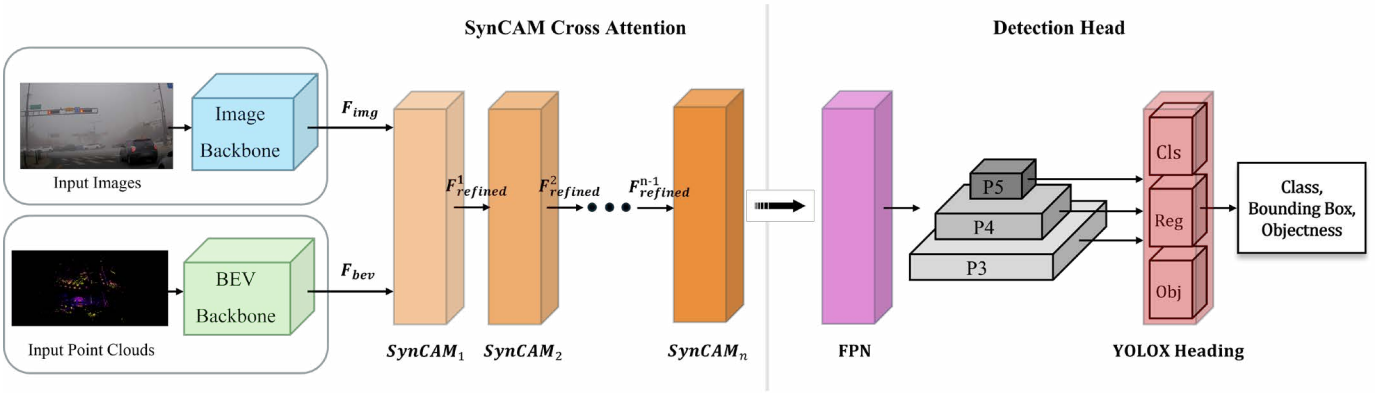

Fig. 1. **Overview of the proposed BEV-ConvFusion framework.** LiDAR point clouds are projected onto dense bird's-eye-view (BEV) feature maps via a BEV backbone, while RGB images are processed through a 2D CNN backbone. The resulting modality-specific features, F_{img} and F_{bev}, are iteratively refined using a sequence of Synergistic Cross-Attention Modules (SynCAMs), which align and exchange complementary geometric and semantic cues across BEV and image domains. The fused representation is then passed through a feature pyramid network (FPN) and a lightweight YOLOX-style detection head to jointly predict object classes, bounding boxes, and objectness scores.

(e.g., projecting LiDAR onto the image plane with depth as an additional channel), but is sensitive to calibration errors and sparse–dense alignment issues. **Late fusion** processes each modality independently and merges only the final predictions, offering robustness at the cost of cross-modal interactions. **Mid-level (deep) fusion** integrates intermediate features and has become dominant, yet many designs rely on costly 3D backbones that hinder real-time use. Our approach also adopts mid-level fusion, but avoids 3D bottlenecks through an efficient 2D pipeline.

B. LiDAR Data Representation for Deep Learning

The unstructured nature of LiDAR point clouds poses a central challenge for learning-based perception. **Voxel-based methods** [4] discretize 3D space into volumetric grids, enabling the use of 3D convolutions, but at the expense of extreme computational overhead. **Point-based methods**, such as PointNet++ [5], directly operate on raw points, capturing fine geometric detail but incurring expensive nearest-neighbor searches that limit scalability. **Pillar-based approaches**, such as PointPillars [6], project point clouds into vertical columns ("pillars") and encode them with a lightweight MLP. The resulting pseudo-image is then processed by a standard 2D CNN backbone in BEV space, thus eliminating 3D convolutions while effectively exploiting spatial context. Compared with these representations, **Bird's-Eye View (BEV) projections** [7] transform point clouds into dense, structured 2D maps that preserve essential spatial cues while enabling compatibility with highly optimized 2D CNNs. This trade-off provides a compelling balance between geometric fidelity and efficiency, though it inevitably introduces some loss of fine-grained 3D detail. Our framework adopts this BEV representation to maximize efficiency while compensating for information loss through synergistic fusion with image features.

C. Attention Mechanisms for Multimodal Learning

Attention mechanisms, first popularized in natural language processing [8], have become a cornerstone of modern deep learning. In particular, **cross-attention** allows one modality to selectively query another, thereby emphasizing features most relevant to the downstream task. This flexible formulation avoids forcing modalities into a rigidly shared feature space, which risks erasing complementary information. However, standard cross-attention modules are computationally demanding due to quadratic complexity in feature map size, making their deployment in real-time systems challenging. Our proposed **Synergistic Cross-Attention Module (SynCAM)** explicitly addresses this trade-off: it employs a structured three-step process of spatial gating, bidirectional semantic cross-attention, and feature refinement, ensuring both efficiency and rich cross-modal interaction. By aligning attention flow with the complementary strengths of LiDAR and camera inputs, SynCAM achieves effective fusion while remaining lightweight enough for real-time inference.

III. PROPOSED METHOD: BEV-CONVFUSION

Our proposed BEV-ConvFusion framework is designed as a fully 2D convolutional pipeline to maximize efficiency and real-time performance. The architecture consists of three main stages: (1) parallel feature extraction from the image and LiDAR BEV streams, (2) the Synergistic Cross-Attention Module (SynCAM) for feature fusion, and (3) a unified detection head for final object prediction.

A. Overall Architecture

As illustrated in Fig. 1, the model ingests synchronized RGB camera images and LiDAR point clouds.

- **The camera stream** employs a pre-trained 2D Vision backbone (e.g., Swin-Transformer [9] or ConvNeXt [10]) to extract multi-scale semantic features.
- **The LiDAR stream** first converts the raw 3D point cloud into a structured 2D BEV representation, which is then processed by a 2D CNN backbone to extract corresponding spatial features.

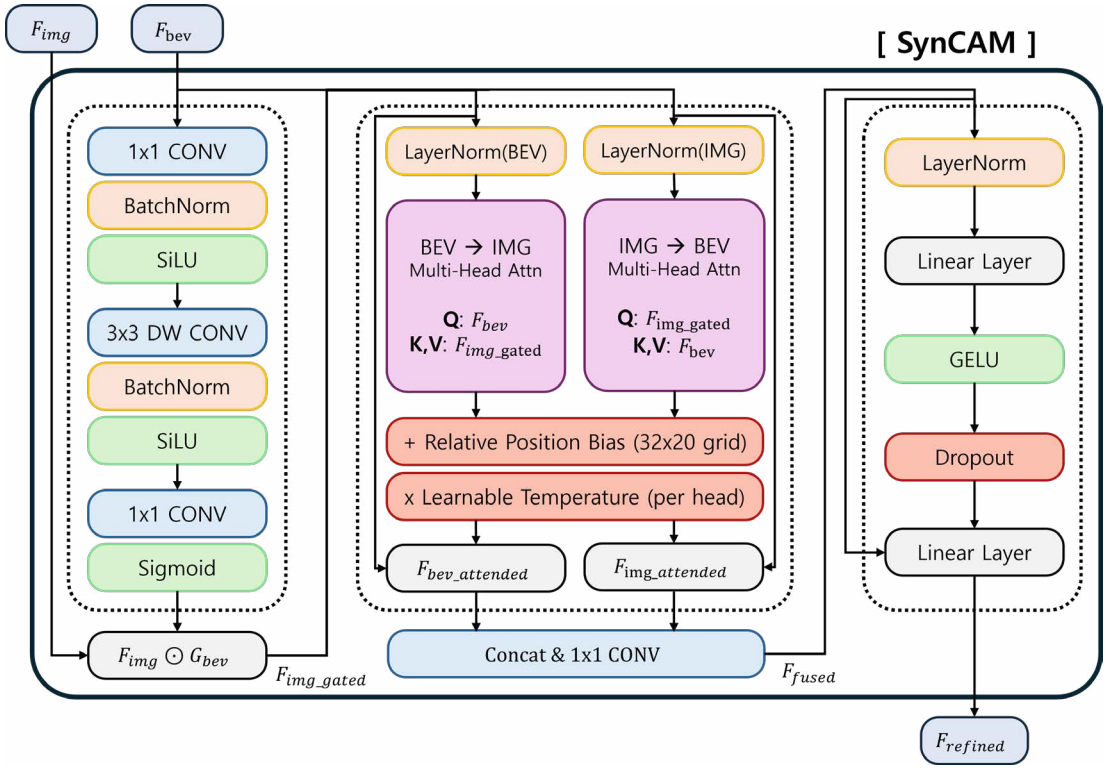

Fig. 2. **Illustration of the Synergistic Cross-Attention Module (SynCAM).** The module takes image features (F_{img}) and BEV features (F_{bev}) as inputs and refines them through three sequential stages: (1) *Spatial Gating*, where BEV-derived spatial priors suppress irrelevant image regions while emphasizing object-relevant areas; (2) *Bidirectional Semantic Cross-Attention*, where gated image features provide semantic cues to the BEV stream and, reciprocally, BEV features enhance the geometric grounding of image features, facilitated by relative position bias and learnable temperature per head; and (3) *Feature Refinement*, where attended features are concatenated and passed through a feed-forward network with residual connections and normalization to stabilize the fused representation.

The extracted feature maps are then synergistically merged by one or more stacked SynCAM blocks. After iterative refinement, the fused features are passed to a detection head for bounding box prediction.

B. LiDAR BEV Representation

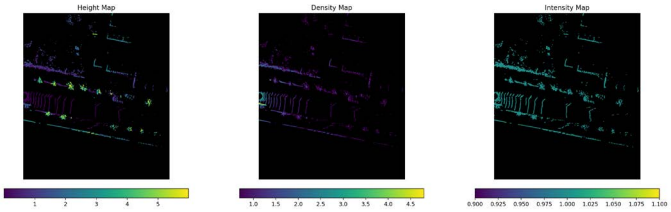

Fig. 3. **Illustration of the three-channel BEV representation generated from a single LiDAR scan.** From left to right: the Height Map, which encodes vertical structure; the Density Map, which reflects point distribution; and the Intensity Map, which captures reflectance strength. Together, these complementary channels provide a rich and discriminative 2D representation of the 3D scene.

To avoid the computational overhead of 3D convolutions, the raw LiDAR point cloud is projected into a multi-channel Bird's-Eye View (BEV) representation. Formally, given a point cloud

$$P = \{p_i\}_{i=1}^{N}, \quad p_i = (x, y, z, r) \in \mathbb{R}^4, \tag{1}$$

where each point encodes spatial coordinates and reflectance intensity, the 3D space is discretized into a 2D grid of size $H \times W$ with a fixed resolution. For each grid cell (i, j), we compute a set of statistics from the subset of points $P_{ij} \subset P$ that fall within the cell boundaries. These channels provide complementary geometric and semantic cues:

- **Height Channel:**

$$H_{ij} = \max_{p \in P_{ij}} z, \tag{2}$$

  which records the maximum elevation within each cell. This enables the model to differentiate tall structures (e.g., vehicles, pedestrians) from flat surfaces (e.g., road markings).
- **Intensity Channel:**

$$I_{ij} = \frac{1}{|P_{ij}|} \sum_{p \in P_{ij}} r, \tag{3}$$

  which represents the average reflectance intensity, useful for discriminating materials and object categories.
- **Density Channel:**

$$D_{ij} = \log\left(|P_{ij}| + 1\right), \tag{4}$$

  which encodes the logarithmic point density per cell, mitigating sparsity variations and providing cues about object solidity.

This process yields a dense, image-like tensor

$$M_{bev} \in \mathbb{R}^{H \times W \times C_{bev}}, \tag{5}$$

where C_{bev} denotes the number of encoded channels (typically three in our design).

C. Dual-Stream Feature Extraction

The two modalities are processed by independent 2D backbones to capture modality-specific multi-scale representations:

$$\{F^l_{img}\}^L_{l=1} = \text{Encoder}_{img}(\text{Image}_{RGB}) \tag{6}$$

$$\{F^l_{bev}\}^L_{l=1} = \text{Encoder}_{bev}(M_{bev}) \tag{7}$$

where $\{F^l_{img}\}$ and $\{F^l_{bev}\}$ denote feature maps extracted at different pyramid levels. For the image encoder, we employ a powerful pre-trained backbone, such as Swin Transformer [9] or ConvNeXt [10], to obtain semantic-rich contextual features. For the BEV encoder, we adopt a 2D CNN backbone (e.g., ResNet [11] or EfficientNet [12]) to effectively extract spatial structures from the BEV representation.

D. Synergistic Cross-Attention Module (SynCAM)

The core of our proposed fusion strategy is the **Synergistic Cross-Attention Module (SynCAM)**, which enables a structured bidirectional refinement between image and BEV features. Rather than relying on a single one-way attention flow, SynCAM explicitly alternates the enhancement process across modalities, allowing each to leverage complementary strengths. As illustrated in Fig. 2, the module operates in three sequential stages that progressively refine and align cross-modal representations.

1) Stage 1: Spatial Gating: We first exploit the spatial prior inherent in BEV features to guide the refinement of image features. A convolutional block transforms the BEV feature map F_{bev} into a spatial attention gate $G_{bev} \in \mathbb{R}^{H' \times W' \times 1}$, which suppresses irrelevant background regions (e.g., sky, distant buildings) in the image stream and highlights object-relevant areas:

$$G_{bev} = \sigma(\text{ConvBlock}(F_{bev})) \tag{8}$$

$$F_{img\_gated} = F_{img} \odot G_{bev} \tag{9}$$

where σ denotes the sigmoid function and $\odot$ represents element-wise multiplication. The resulting feature map, F_{img_gated}, is thus a spatially refined version of the original image features, improving downstream cross-modal alignment.

2) Stage 2: Bidirectional Semantic Cross-Attention: After spatial refinement, we perform bidirectional cross-attention between the two modalities. To improve stability and efficiency, we adopt pre-normalization and adaptive pooling before computing the attention.

Pre-Normalization. Both BEV and gated image features are first flattened and normalized:

$$\tilde{F}_{bev} = \text{LayerNorm}(\text{Flatten}(F_{bev})), \tag{10}$$

$$\tilde{F}_{img\_gated} = \text{LayerNorm}(\text{Flatten}(F_{img\_gated})). \tag{11}$$

Adaptive Pooling. To reduce redundant tokens and improve memory efficiency, the features are pooled to a fixed grid size:

$$\tilde{F}_{bev\_pool} = \text{AdaptivePool}(\tilde{F}_{bev}, (H_{pool}, W_{pool})), \tag{12}$$

$$\tilde{F}_{img\_pool} = \text{AdaptivePool}(\tilde{F}_{img\_gated}, (H_{pool}, W_{pool})), \tag{13}$$

Bidirectional Attention. We define the query, key, and value projections for both BEV and image features as:

$$\begin{aligned} Q_b &= W^b_Q \tilde{F}_{bev\_pool}, \quad K_b = W^b_K \tilde{F}_{bev\_pool}, \quad V_b = W^b_V \tilde{F}_{bev\_pool}, \\ Q_i &= W^i_Q \tilde{F}_{img\_pool}, \quad K_i = W^i_K \tilde{F}_{img\_pool}, \quad V_i = W^i_V \tilde{F}_{img\_pool}. \end{aligned} \tag{14}$$

Then the bidirectional attention updates are given by:

$$\begin{aligned} F_{bev\_attn} &= \text{softmax}\left(\frac{Q_b K_i^T}{\sqrt{d_k}}\right) V_i, \\ F_{img\_attn} &= \text{softmax}\left(\frac{Q_i K_b^T}{\sqrt{d_k}}\right) V_b, \end{aligned} \tag{15}$$

where W_Q, W_K, and W_V are learnable projection matrices, and d_k is the key dimension. This reciprocal attention flow ensures that BEV geometry sharpens image semantics, while image semantics enrich the BEV representation.

Residual Connections. The attention outputs are interpolated back to the original resolution and fused with residual connections:

$$F_{bev\_attended} = F_{bev} + \text{Interpolate}(F_{bev\_attn}), \tag{16}$$

$$F_{img\_attended} = F_{img\_gated} + \text{Interpolate}(F_{img\_attn}). \tag{17}$$

3) Stage 3: Feature Refinement: Finally, the attended BEV and image features are fused into a unified representation by first concatenating the two modalities along the channel dimension and then projecting the result through a 1×1 convolution, followed by batch normalization and ReLU activation:

$$F_{fused} = \text{ReLU}\big(\text{BN}(\text{Conv}_{1\times1}(F_{bev\_attended} \oplus F_{img\_attended}))\big), \tag{18}$$

where $\oplus$ denotes channel-wise concatenation.

To stabilize the fused representation, we apply a feed-forward network (FFN) with residual connection and layer normalization:

$$F_{refined} = F_{fused} + FFN(\text{LayerNorm}(F_{fused})). \tag{19}$$

The refined feature $F_{refined}$ is propagated to the next SynCAM block, or, after the final block, to the detection head.

Unlike conventional cross-attention modules that directly operate on the full-resolution feature maps, SynCAM incorporates spatial gating and adaptive pooling to reduce redundant interactions and focus on semantically relevant regions. While the overall computational complexity remains quadratic in theory, in practice the reduced token count and gated attention map lead to more efficient usage of computation and memory.

E. Detection Head and Loss Function

After the final SynCAM block, the fused features are aggregated by a Feature Pyramid Network (FPN) to ensure robustness across object scales. We employ a YOLOX-inspired

decoupled head [13], which separates classification and regression branches. This design improves optimization stability by preventing gradient interference between tasks.

The head outputs three predictions at each location: classification logits (L_{cls}), objectness scores (L_{obj}), and bounding box regression (L_{bbox}, L_1). The total training loss is formulated as:

$$L_{total} = L_{cls} + \alpha \cdot L_{bbox} + L_{obj} + L_1, \tag{20}$$

where:

- L_{cls}: cross-entropy loss for classification.
- L_{bbox}: An IoU-based regression loss (e.g., GIoU/CIoU) responsible for the overall box alignment.
- L_{obj}: binary cross-entropy loss for objectness.
- L_1: A smooth L1 loss applied directly to the bounding box coordinates (e.g., center points, width, height). This term complements the IoU loss, especially in early training stages, and helps stabilize the optimization.

The weighting coefficient α balances the regression loss relative to classification and objectness terms. This is crucial since bounding box regression typically has a different scale, and without reweighting, optimization can become unstable or biased toward one sub-task. In our experiments, we set $\alpha = 5$ to achieve stable convergence and improved detection accuracy.

IV. EXPERIMENTS

A. Datasets and Implementation Details

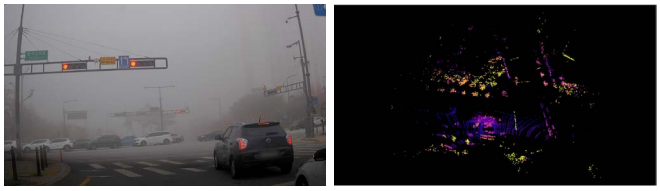

Fig. 4. **Example of a synchronized data pair from the dataset.** Left: Front-camera RGB image. Right: The corresponding LiDAR point cloud visualized as a BEV density map.

Dataset. We evaluate our approach on the *Adverse Weather Data for Autonomous Passenger Vehicles* dataset provided by the AI-Hub project[1]. The dataset contains synchronized front-camera images and 360° LiDAR point clouds collected under diverse adverse weather conditions (e.g., rain, snow, fog). Following the official split, we use 15,000 pairs for training and 3,800 for validation. Labels are annotated as 2D bounding boxes for five categories: vehicle, pedestrian, two-wheeler, traffic sign, and traffic light. Fig. 4 illustrates an example of synchronized input pairs, showing an RGB image and its corresponding BEV LiDAR density map.

Implementation Details. Our implementation is based on MMDetection 3.3.0 with PyTorch. For the image encoder, we use ConvNext-Tiny [10] pre-trained on ImageNet-1K [14], while the BEV encoder employs a 2D ResNet-18 [11]. Models are trained for 24 epochs with AdamW (lr=$1e^{-4}$, weight decay 0.05) on a single NVIDIA A100 GPU.

[1]This research used datasets from 'The Open AI Dataset Project (AI-Hub, S. Korea)'. All data information can be accessed through 'AI-Hub (www.aihub.or.kr)'.

B. Component Selection

To identify the optimal unimodal backbone for the camera stream, we benchmark several popular 2D detectors. Results are summarized in Table I. ConvNeXt-Tiny [10] achieves the highest accuracy (mAP@.50 = 73.3%), outperforming Swin-Tiny [9] and EfficientNet [12]. Interestingly, YOLOv8-m [15] shows notably lower accuracy in this domain, highlighting the importance of backbone selection for robust feature extraction under adverse conditions.

TABLE I
PERFORMANCE OF CAMERA-ONLY 2D OBJECT DETECTION MODELS.

Backbone	mAP@[.5:.95]	mAP@.50
EfficientNet [12]	50.3	69.1
ConvNeXt-Tiny [10]	**54.3**	**73.3**
YOLOv8-m [15]	33.4	51.6
Swin-Tiny [9]	52.0	71.0

From these results, **ConvNeXt-Tiny** [10] is selected as the default backbone for subsequent fusion experiments.

C. Comparison with Other Approaches

We next compare our proposed BEV-ConvFusion with both 2D and 3D methods. Since the dataset provides only 2D bounding box annotations, direct supervision for 3D detectors is not available. To ensure a fair comparison, we converted the outputs of 3D-only detectors (e.g., PV-RCNN, Voxel R-CNN) into BEV representations and trained them with 2D bounding box supervision projected onto the BEV plane. This allows both 2D and 3D models to be evaluated under the same annotation format. Results are presented in Table II.

- **2D-only baseline** (ConvNeXt-Tiny [10]) achieves strong performance (mAP@.50 = 73.3%), surpassing even the best 3D model, PV-RCNN [16] (72.3%). This highlights the maturity of modern image-based detectors.
- **3D-only models**, while providing a strong geometric baseline, highlight the challenge of relying solely on LiDAR under adverse weather conditions. For instance, even the top-performing PV-RCNN [16] at 72.3% mAP@.50 does not outperform the 2D-only baseline, suggesting that geometric data alone is insufficient without rich semantic context. This underscores the need for effective sensor fusion.
- **Fusion model (BEV-ConvFusion)** achieves the best performance (mAP@.50 = 75.3%), clearly surpassing both unimodal baselines. This confirms that our cross-attention-based fusion effectively leverages the complementary strengths of camera and LiDAR.

The results emphasize that fusion is not merely additive; it achieves synergistic gains by enhancing robustness in challenging weather.

TABLE II
PERFORMANCE COMPARISON WITH 2D AND 3D MODELS.

Modality	Method	mAP@.50
2D-Only	ConvNeXt-Tiny [10]	73.3
3D-Only	PV-RCNN [16]	72.3
	Voxel R-CNN [17]	71.6
	SECOND [18]	68.6
	PointRCNN [19]	66.4
Fusion	**BEV-ConvFusion**	**75.3**

D. Ablation Studies

Impact of SynCAM Block Depth. To analyze the contribution of SynCAM depth, we vary the number of stacked blocks from 1 to 4. Results are shown in Table III.

TABLE III
ABLATION STUDY ON THE NUMBER OF STACKED SYNCAM BLOCKS.

# SynCAM Blocks	mAP@.50
1 Block	73.6
2 Blocks	74.7
3 Blocks	**75.3**
4 Blocks	73.9

As presented in Table III, the model's performance steadily improves as the number of SynCAM blocks increases from one to three, peaking at 75.3% mAP@.50. This trend demonstrates that stacking blocks is crucial for enabling a deeper, iterative refinement where camera and LiDAR modalities can reciprocally enhance each other. However, the performance declines with the addition of a fourth block. This suggests that excessive depth leads to overfitting due to increased model complexity and potential feature saturation. Therefore, we selected a three-block configuration for our final model, as it achieves the optimal balance between feature fusion capability and generalization.

V. CONCLUSION

In this paper, we presented **BEV-ConvFusion**, a novel and efficient LiDAR–camera fusion framework designed to reconcile the trade-off between accuracy and real-time performance in autonomous driving perception. By projecting LiDAR point clouds into a 2D bird's-eye view (BEV) representation, our method circumvents the need for computationally intensive 3D backbones and enables the entire pipeline to be executed with highly optimized 2D convolutional architectures.

At the core of the framework lies the **Synergistic Cross-Attention Module (SynCAM)**, which introduces a structured bidirectional refinement mechanism. Through alternating spatial gating and bidirectional semantic cross-attention, SynCAM ensures that both modalities iteratively enhance each other, producing a robust joint representation that captures semantic richness from images and geometric stability from LiDAR.

Comprehensive experiments demonstrate that BEV-ConvFusion achieves strong detection performance, outperforming unimodal baselines and rivaling state-of-the-art 3D detectors, all while operating with substantially lower computational cost. Moreover, the framework shows increased robustness under adverse weather conditions, where the limitations of camera-only systems become apparent and LiDAR provides indispensable structural cues.

Overall, BEV-ConvFusion establishes an effective and efficient multimodal fusion paradigm, providing a practical pathway toward safer and more reliable perception systems for autonomous vehicles.

REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2012, pp. 3354–3361.

[2] M. Hnewa and H. Radha, "Object detection under challenging weather conditions: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4013–4033, 2021.

[3] F. M. L. Ribeiro, C. R. de G. C. Pacheco, and A. L. Koerich, "Deep learning for multi-modal fusion in autonomous vehicles: A review of recent approaches," *Applied Artificial Intelligence*, vol. 35, no. 15, pp. 1216–1244, 2021.

[4] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 4490–4499.

[5] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[6] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 12697–12705.

[7] Z. Liu, H. Tang, J. Amini, X. Yang, H. Hu, J. Kautz, and T. S. K. Beijbom, "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 9361–9369.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[9] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 10012–10022.

[10] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022, pp. 11976–11986.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 770–778.

[12] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114.

[13] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis. (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[15] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," version 8.0.0, GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[16] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020, pp. 10529–10538.

[17] J. Deng, S. E. Reed, J. Krause, and L. Fei-Fei, "Voxel R-CNN: Towards high performance voxel-based 3D object detection," in *Proc. AAAI Conf. on Artif. Intell.*, 2021.

[18] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[19] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 770–779.