A Survey of Deep Learning-Based Multi-Sensor Fusion for 3D Object Detection in Autonomous Driving

Yining Liu*
Stony Brook Institute at Anhui University, Anhui University, Hefei, Anhui, China
**Corresponding Author.*

Abstract: Multi-sensor fusion has become a fundamental framework for robust 3D object detection in autonomous driving, as it can mitigate the inherent limitations of single-modal perception under challenging real-world conditions. This paper presents a systematic review of mainstream cross-modal fusion paradigms (early, mid-level, and late fusion) tailored to 3D detection tasks, with a focus on their core mechanisms and deployment implications. We elaborate on the critical challenges of coordinate alignment and shared representation interfaces, highlighting the Bird's-Eye View (BEV) as a promising unified fusion space that balances spatial awareness and planning compatibility. A comprehensive comparison is conducted among three dominant 3D representations-point-based, voxel-based, and BEV-based-revealing their inherent trade-offs in computational cost, memory consumption, spatial structure preservation, and deployment efficiency. Furthermore, we categorize typical deployment-facing failure modes, including domain shift, calibration drift, sensor dropout, and adversarial attacks, and summarize corresponding layered mitigation strategies such as health monitoring, consistency checking, data augmentation, and graceful degradation. This work provides a structured overview of the state-of-the-art in multi-sensor fusion for 3D detection, offering practical insights for both academic research and industrial deployment.

Keywords: Autonomous Driving; 3D Object Detection; Multi-Sensor Fusion; BEV; LiDAR-Camera; Deep Learning; Transformer

1. Introduction

Autonomous driving is often described in terms of stronger models and larger datasets, yet in real systems the decisive question is whether perception stays stable when sensing conditions drift away from the benchmark distribution.[7,10] Among perception tasks, 3D object detection is central because it grounds tracking, prediction, planning, and collision avoidance in a metric description of surrounding traffic participants.[4,10] This requirement is difficult to meet on-road: illumination and weather vary widely, occlusions are common in dense traffic, and many safety-critical events lie in the long tail and are poorly represented in finite datasets.[7,10]

No single modality fails gracefully under all conditions. Cameras provide rich semantics but are vulnerable to low light and glare. LiDAR offers accurate geometry but becomes range-sparse and can degrade in adverse weather. Radar is relatively weather-robust and provides velocity cues, but its spatial resolution is limited.[7,8] Fusion should therefore be viewed not only as a way to increase average detection scores, but also as a way to reduce modality-specific failure modes by combining complementary signals.[8,10]

At the same time, "fusion" is not a single design decision. Practical systems must handle calibration drift and timestamp jitter, mismatched fields of view and sampling densities, and modality-specific noise, all under strict latency and memory budgets.[7,8,10] Meanwhile, many recent 3D detectors move toward BEV-centric unification and efficient geometric encoders. This trend is promising for on-vehicle deployment, but it also expands the design space and makes it harder to choose an appropriate solution without a clear taxonomy.[6,19-22]

With these considerations in mind, this survey reviews deep learning-based multi-sensor 3D object detection with an emphasis on camera-LiDAR fusion. Specifically, we:

Taxonomy: categorize fusion into early, mid-level, and late fusion, and discuss trade-offs in information retention, alignment sensitivity,

robustness, and compute.[8,10]
Method synthesis: connect representative 3D detection foundations to fusion frameworks, emphasizing BEV/voxel representations and efficient geometric encoders commonly used in deployable systems.[5,6,19-24]
Engineering perspective: summarize deployment constraints (e.g., calibration drift, domain shift, adversarial risk, latency, and resource budgets) and practical mitigations.[1,7,10]
Outlook: highlight engineering directions including data-centric closed-loop iteration, sensor health monitoring, graceful degradation, and system-level evaluation.[2,4,7,10]
The remainder of the paper is organized as follows. Section 2 introduces datasets and metrics. Section 3 reviews fusion paradigms and representative model families. Section 4 discusses robustness, safety, security, and efficiency. Section 5 outlines open challenges and future directions, and Section 6 concludes.

2. Background

2.1 Multi-Sensor Perception for Autonomous Driving

Perception is the front end of the autonomous driving stack. It produces the geometric and semantic estimates that downstream planning and decision-making depend on.[4,7] Because on-road conditions change sharply with illumination, weather, and traffic density, production systems rarely rely on a single sensor. Instead, vehicles combine cameras, LiDAR, radar, GNSS/IMU, and sometimes ultrasonic sensors to improve coverage and redundancy.[7,8]
The motivation for fusion is clearest when modality limitations are treated as failure modes. Cameras provide dense semantics but can degrade under low light or glare and do not measure metric depth directly. LiDAR provides accurate geometry but becomes sparse with range and may degrade in harsh weather. Radar is relatively weather-robust and provides velocity cues, but with coarser spatial resolution.[7,8] Fusion is therefore used not only to improve average scores, but also to reduce the chance that a single-modality weakness dominates system behavior in safety-critical moments.[8,10]
From an engineering standpoint, perception must also satisfy strict latency budgets, finite compute limits, and robustness requirements tied to safety.[7,10] Security has become part of the deployment-facing discussion as well, since adversarial manipulation of deep perception models can lead to safety-relevant errors.[1] These constraints explain why recent work increasingly evaluates fusion designs in terms of robustness, diagnosability, and runtime behavior-not accuracy alone.[1,7,9,10]

2.2 Common Datasets and Evaluation Metrics for 3D Object Detection

Most progress in 3D detection is reported on public benchmarks, so it is important to understand dataset assumptions when interpreting claimed improvements.[10] KITTI remains a standard reference for camera-LiDAR 3D detection. nuScenes provides a broader multi-modal setup (including radar) and more diverse scenarios. The Waymo Open Dataset offers large-scale multi-modal data with extensive annotations.[16-18] These benchmarks differ in sensor configurations, label definitions, scene distributions, and long-tail coverage, which can shift what the “best” fusion design looks like in practice.[10,17,18]
Evaluation is typically reported using AP/mAP under 3D IoU or BEV IoU thresholds.[16-18] nuScenes further reports NDS, which combines mAP with multiple error terms (e.g., translation and orientation) to better reflect driving-relevant quality.[17] In this survey, we treat benchmark metrics as necessary but incomplete: deployment decisions also depend on latency, memory footprint, and robustness under domain shift and adverse sensing conditions.[1,7,10]

3. Survey of Fusion Paradigms and Representative Foundations

3.1 Taxonomy: Early, Mid-Level, and Late Fusion

In autonomous driving perception, “fusion” can occur at different points in the detection pipeline, and that placement strongly affects both accuracy and failure behavior.[8,10] A widely used breakdown is early, mid-level, and late fusion as shown in Figure 1, which distinguishes whether modalities are combined at the input, feature, or decision stage.[8,10] This taxonomy is useful because it corresponds to engineering trade-offs: how much cross-modal information is retained, how sensitive the pipeline is to calibration and synchronization, what the runtime and memory costs look like, and how

the system behaves when one modality degrades.[7,8,10]

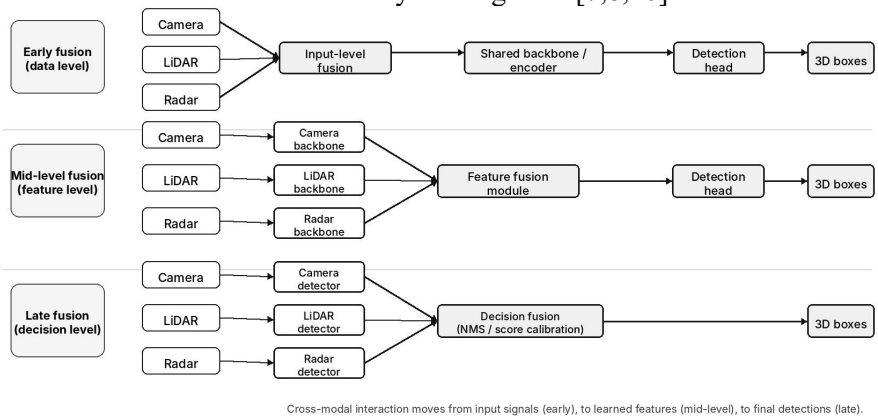

Figure 1. Fusion Paradigms in 3D Detection: Early, Mid-Level, and Late Fusion Differ by where Cross-Modal Interaction is Performed.

Early fusion (data-level). Early fusion mixes raw or lightly processed signals before the backbone performs substantial feature extraction-for example, enriching LiDAR points with image cues sampled via projection, or stacking synchronized measurements into a unified input representation.[7,8] Its main advantage is information retention: fine-grained cues can be combined before later layers compress them.[7,8] The cost is sensitivity. Early fusion is typically the most brittle under calibration drift and timestamp misalignment, and it must cope with heterogeneous sampling densities and noise in a tightly coupled preprocessing pipeline.[7,8]

Mid-level fusion (feature-level). Mid-level fusion first extracts modality-specific features and then performs cross-modal interaction at intermediate network stages.[8,9,10] It has become the dominant choice because each modality can use an appropriate backbone while the fusion module learns how to combine complementary cues.[9,10] In practice, performance often hinges on (i) whether features are aligned well enough to interact meaningfully, and (ii) whether the chosen fusion operator-concatenation, gating, or attention-fits the compute budget and expected noise patterns [8,9,10].

Late fusion (decision-level). Late fusion combines high-level outputs (e.g., 3D boxes and confidence scores) from separate modality-specific detectors, often via score re-weighting and NMS-style consensus [8,10]. From an engineering standpoint, the main benefit is modularity: each detector can remain operational if another sensor becomes unreliable.[8,10] The limitation is that cross-modal interaction happens after strong information compression, so late fusion may miss complementary evidence that would be available at feature level.[8,10]

Overall, selecting a fusion paradigm is not just an accuracy decision. It also sets expectations for calibration tolerance, diagnosability, runtime cost, and robustness, including under realistic sensor imperfections and attacks.[1,7,10]

3.2 From 2D Object Detection to 3D Detection: Representation and Backbone Foundations

Although the target task is 3D detection, many backbone and training practices in autonomous driving trace back to 2D object detection. Two-stage detectors such as Faster R-CNN established proposal-based pipelines and ROI feature aggregation, while instance-centric methods such as Mask R-CNN influenced the organization of backbones and heads.[19,20] One-stage detectors (e.g., YOLO) demonstrated that dense prediction can achieve high throughput, reinforcing the emphasis on end-to-end efficiency in latency-constrained stacks.[12-14]

Feature hierarchy is another key inheritance. FPN-style multi-scale extraction is valuable in driving because object size and visibility change sharply with distance.[11] Modern 3D pipelines adopt similar multi-scale ideas but operate on geometric representations-points, voxels, or BEV grids-to capture metric structure and long-range spatial context [5,6,10]. Representative LiDAR-based detectors (PIXOR, PointPillars, SECOND, and CenterPoint) further show why BEV/voxel representations are widely used: they often strike a practical balance between accuracy and real-time feasibility, and they provide a natural interface for multi-sensor fusion, as summarized compactly in Figure 2 and illustrated qualitatively in Figure 3 [19-22].

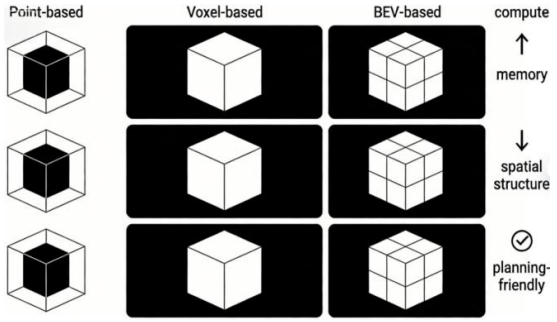

Figure 2. Compact Comparison of Point-, Voxel-, and BEV-Based Representations for 3D Detection

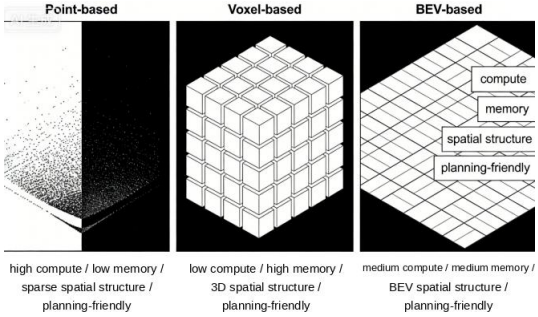

Figure 3. Qualitative Trade-offs among Point-Based, Voxel-Based, and BEV-Based 3D Encodings

3.3 Camera-LiDAR Fusion for 3D Object Detection

Camera-LiDAR fusion is the most common multi-sensor configuration for 3D detection in autonomous driving because it combines complementary strengths. Cameras provide dense semantics and appearance cues, while LiDAR provides metric geometry that directly supports localization and planning.[7,8,10] Under the taxonomy in Section 3.1, camera-LiDAR methods are typically described as early, mid-level, or late fusion. In deployment-facing terms, the main differences are (i) where alignment is performed, (ii) how tightly the modalities are coupled, and (iii) which representation space serves as the shared interface, as shown in Figure 4 and 5.[8,9,10]

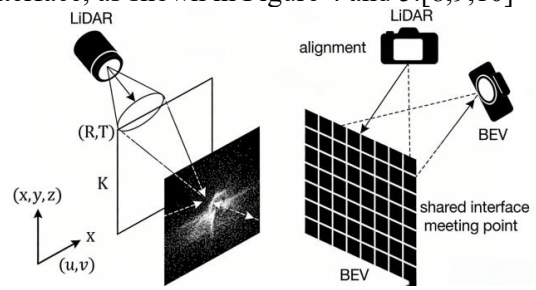

Figure 4. Camera-LiDAR Geometric Alignment: Point-to-Image Projection and BEV Interface

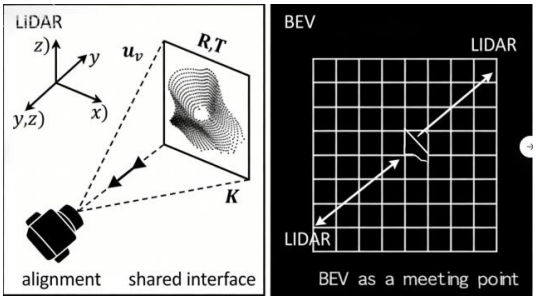

Figure 5. BEV as a Shared Interface for Camera-LiDAR Feature Fusion

3.3.1 Early fusion (data-level fusion)

Early fusion injects camera information into LiDAR (or vice versa) before substantial feature extraction. Typical implementations project LiDAR points into the image to sample pixel features, or augment point attributes with image-derived semantics.[7,8]

Why it can work. By combining signals before separate backbones compress each modality, early fusion can preserve fine-grained complementary cues.[7,8]

Where it breaks in practice. Early fusion is usually the most sensitive to calibration drift and timestamp misalignment, and projection-based association becomes ambiguous under occlusion or limited visibility.[7,8] It also tends to require tightly coupled preprocessing, which increases pipeline complexity and makes debugging and validation harder in on-vehicle deployment.[7]

3.3.2 Mid-level fusion (feature-level fusion)

Mid-level fusion extracts modality-specific features first and then performs cross-modal interaction at intermediate network layers.[8,9,10] It dominates recent systems because it keeps strong, specialized backbones per modality while enabling learned fusion modules to exploit complementarity.[9,10]

In practice, mid-level design is largely determined by three choices:

Alignment: explicit geometric projection versus implicit learned association to tolerate imperfect calibration;[7,8,9,10]

Fusion operator: concatenation, gating, or attention-style interaction with different compute and robustness profiles;[9,10]

Shared interface: fusing in image space, voxel space, or a unified BEV space.[7,8,9,10]

BEV is particularly attractive because it provides a metric, planning-friendly coordinate system and can serve as a common interface for multi-modal features.[19-22] Representative examples include voxel-based fusion (e.g., MVX-Net) and BEV-unified fusion (e.g.,

BEVFusion).[23,24]

Trade-off. Mid-level fusion can deliver strong accuracy, but multi-branch backbones and cross-modal interaction often raise memory cost and latency. This makes efficiency-aware design essential under strict latency budgets.[7,10]

3.3.3 Late fusion (decision-level fusion)

Late fusion merges modality-specific detection outputs-typically 3D boxes and confidence scores-after each modality has produced its own predictions.[8,10]

Why it is used. It is modular and can tolerate partial sensor degradation, since each modality-specific detector remains independently functional.[8,10]

Limitation. Because modalities only interact at the decision stage, late fusion may miss fine-grained complementary evidence that would be accessible at feature level.[8,10]

3.3.4 Learning-based fusion trends

Transformer-based fusion has been explored to model cross-modal relationships through attention, and it has been adopted in end-to-end driving settings to combine multi-sensor information for downstream tasks.[3,25] Although this survey focuses on 3D object detection, attention-based fusion remains conceptually relevant. It offers a flexible interaction mechanism that can be adapted to detection-oriented designs, provided that alignment errors and runtime cost are managed explicitly.[25]

3.3.5 Author's note: when to prefer which fusion strategy

From a deployment-facing viewpoint, the "best" fusion strategy depends on what is expected to be unstable in the system. If calibration and synchronization are difficult to maintain over time, designs that depend on brittle point-to-pixel correspondence are typically higher risk.[7,8,10] If compute is the binding constraint, heavily multi-stage mid-level fusion may be hard to run within strict latency budgets unless the representation and fusion operator are simplified-often by using BEV-style unification as the shared interface.[19-22]

4. Performance Optimization and Engineering Considerations

4.1 Robustness, Safety, and Security for On-Vehicle Perception

For on-vehicle perception, robustness should be treated as a primary requirement rather than an assumed byproduct of higher benchmark accuracy.[7,10] In camera-LiDAR fusion, failure sources include benign factors-weather and illumination changes, sensor aging, calibration drift, and domain shift-as well as malicious factors such as adversarial perturbations and spoofing attempts.[1,7] Because perception outputs feed directly into planning and control, errors can propagate into unsafe maneuvers. Robustness evaluation and mitigation are therefore essential in safety-critical deployment.[4,10] More broadly, robustness targets and strict latency budgets should be defined with downstream constraints in mind, since the perception-to-actuation pathway determines whether the vehicle can brake or avoid hazards safely under time pressure.[2,4]

Sensor imperfections and drift. Real vehicles experience gradual-and sometimes abrupt-changes in sensor parameters, including calibration drift and timestamp misalignment.[7,8] Practical countermeasures include disciplined calibration workflows, online health monitoring that checks cross-modal alignment quality, and fusion mechanisms that tolerate small projection errors (e.g., soft association rather than brittle hard matching).[7,8,10]

Domain shift and dataset bias. Models that perform well on a benchmark may degrade in new cities, sensor configurations, or traffic patterns.[10] Mitigation is often data-driven: broaden coverage, apply targeted augmentation, and maintain continual evaluation pipelines to detect regressions over time.[7,10] When appropriate, multi-task perception formulations can also regularize representations and improve stability across conditions.[9]

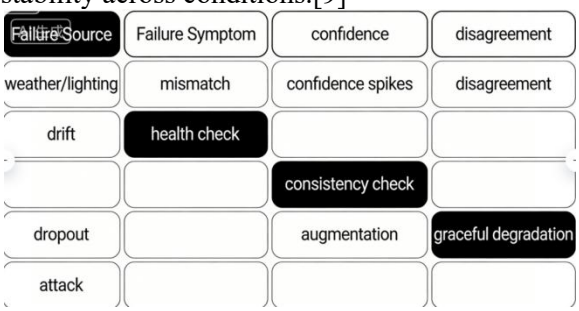

Figure 6. Runtime Monitoring and Layered Mitigation Workflow for Camera-LiDAR Fusion Failures

Security-aware defenses under resource constraints. Deep perception models can be vulnerable to adversarial manipulation, with camera-facing components often being an

exposed surface [1]. Practical defenses are typically layered: training-time robustness improvements, cross-modal consistency checks (to detect suspicious modality disagreement), and runtime monitoring that flags abnormal confidence dynamics [1,7,10]. Defenses should be assessed under realistic compute budgets, since overly expensive protection mechanisms may be infeasible on embedded platforms.[1,10]

Table 1. Deployment Checklist of Failure Sources, Symptoms, and Robustness Responses for Camera-LiDAR Fusion

Failure Source	Failure Symptoms	
weather/lighting	mismach	health check
drift		consistency check→
	confidence spikes	auggmentation
dropout	disagreement	
attack		graceful degrgdation

4.2 Efficiency and Real-Time Deployment Constraints

On an autonomous vehicle, perception runs under strict latency budgets and limited onboard compute, so efficiency optimization can be as important as peak accuracy.[7,10] Fusion adds overhead through alignment, multi-branch backbones, and cross-modal interaction modules, which increases both latency and memory footprint.[7,10] Deployable systems therefore benefit from co-design across model architecture, input resolution, and fusion granularity rather than treating fusion as a plug-in module.[6,7]

Architectural trade-offs. In 2D detection, one-stage designs (e.g., YOLO-family) illustrate a practical principle: simpler pipelines often improve throughput and reduce tail latency.[12-14] Similar ideas carry over to fusion-based 3D detection, where systems may reduce the number of fusion stages, constrain attention complexity, and adopt compact voxel/BEV encodings that are cheaper to process.[6,10]

Point-cloud bottlenecks. Efficient point-cloud detection depends heavily on discretization choices (e.g., voxelization) and sparse computation.[6] PointPillars and CenterPoint are commonly cited examples showing that BEV-style design can remain competitive while keeping inference efficient, which matters when fusion must share compute with other on-vehicle modules.[20,22]

Deployment-facing metrics. For real systems, AP/mAP is not sufficient. Engineering teams routinely track end-to-end latency, tail latency, peak memory, utilization, and stability under sensor dropouts.[7,10] These measurements often determine whether a model is shippable, since marginal accuracy gains may not justify higher energy consumption or missed latency deadlines.[7,10]

4.3 Deployment Checklist and Typical Failure Modes

To translate benchmark performance into dependable on-road behavior, it is useful to evaluate fusion systems with a small set of deployment checks.[7,10] Typical checklist items include: (i) sensitivity to small extrinsic drift and timestamp jitter; (ii) behavior under partial sensor degradation (e.g., sparse LiDAR returns or camera overexposure); (iii) cross-modal consistency monitoring (whether modality disagreement is detectable); and (iv) tail-latency profiling across the full pipeline, including preprocessing and post-processing.[7,8,10]

Common failure modes in camera-LiDAR fusion include brittle point-to-pixel correspondence under occlusion, overconfident predictions when one modality dominates the fusion module, and silent performance drops under domain shift.[7,10] Since adversarial manipulation is a realistic concern-especially via the camera modality-defensive design should combine robustness training with runtime anomaly detection and multi-sensor sanity checks that remain feasible under embedded compute budgets.[1,7,10]

5. Open Challenges and Future Directions

Camera-LiDAR fusion has advanced quickly on public benchmarks, yet dependable large-scale on-vehicle deployment remains difficult.[7,10] A recurring reason is that many risks are system-level: they arise from the interaction between sensors, models, and operating conditions, and can be underestimated by benchmark-centric development loops.[1,7] Below we summarize deployment-facing challenges and the directions that appear most actionable in practice.

Long-tail events and distribution shift. Autonomous driving is an open-world problem. Rare events and extreme conditions may be infrequent, but they can dominate safety risk.[8,15] Because current datasets cannot fully represent these cases, models may behave brittlely when deployed to new regions, weather

patterns, or sensor configurations.[10] In practice, improving long-tail behavior is largely a data and validation problem-targeted collection, scenario-based testing, and robustness-oriented evaluation-rather than something architecture changes alone can guarantee.[7,10]

Security threats under embedded budgets. Deep perception pipelines can be exposed to adversarial manipulation, and the camera modality is often an accessible attack surface.[1] The open challenge is to design defenses that are effective and measurable while remaining feasible on vehicle hardware, without introducing new failure modes such as excessive false alarms under benign noise.[1,7] This motivates lightweight runtime monitoring and multi-sensor consistency checks that can be integrated into production stacks.[1,7,10]

Data-centric development via closed-loop iteration. Many field failures come from data gaps, inconsistent labels on edge cases, or missing diversity, not from insufficient model capacity.[7,10] Closed-loop iteration-mining difficult cases from field logs, prioritizing by safety impact, labeling efficiently, and retraining with targeted augmentation-remains one of the most scalable improvement routes.[7,10] This is especially important for fusion, where corner cases can be triggered by modality interaction (e.g., partial camera occlusion combined with sparse LiDAR returns) and are hard to anticipate from single-modality analysis.[7]

Sensor health monitoring and graceful degradation. On-vehicle deployment must assume imperfect sensors: calibration can drift, synchronization can jitter, and individual modalities may temporarily degrade.[7,8,10] A robust system should therefore include diagnostics (e.g., alignment-quality indicators and abnormal point-density checks) and enforce graceful degradation policies when a modality becomes unreliable.[7,8,10] For fusion in particular, detecting modality disagreement and avoiding overconfident outputs remains a high-impact direction.[1,7]

Hardware-aware optimization and end-to-end validation. Meeting strict latency budgets requires profiling the full pipeline-preprocessing, data movement, inference, and post-processing-rather than optimizing network FLOPs in isolation.[6,7,10] Hardware-aware operator choices and simplified fusion designs can reduce tail latency and improve runtime stability without disproportionate accuracy loss, which is often what decides deployability in practice.[6,15]

6. Conclusion

This survey synthesized deep learning methods for multi-sensor fusion in autonomous driving 3D object detection, with an emphasis on camera-LiDAR fusion and the practical trade-offs among early, mid-level, and late fusion.[7,8,10] We linked representative 3D detection foundations to BEV-oriented unification trends, and discussed deployment-facing considerations spanning robustness, safety, security, and real-time efficiency.[1,6,19-24] Finally, we argued that progress toward dependable on-vehicle deployment will depend not only on model design but also on engineering practices such as closed-loop data iteration, sensor health monitoring, graceful degradation, and system-level evaluation.[2,4,7,10]

References

[1] B. Badjie, J. Cecílio, and A. Casimiro, "Adversarial Attacks and Countermeasures on Image Classification-based Deep Learning Models in Autonomous Driving Systems: A Systematic Review," *ACM Computing Surveys*, vol. 57, no. 1, pp. 20:1-20:52, 2024.

[2] X. Hua, J. Zeng, H. Li, *et al*., "A Review of Automobile Brake-by-Wire Control Technology," *Processes*, vol. 11, no. 4, p. 994, 2023.

[3] D. Coelho and M. Oliveira, "A Review of End-to-End Autonomous Driving in Urban Environments," *IEEE Access*, vol. 10, pp. 75296-75311, 2022.

[4] J. Hu, Y. Wang, S. Cheng, *et al*., "A Survey of Decision-Making and Planning Methods for Self-Driving Vehicles," *Frontiers in Neurorobotics*, vol. 19, p. 1451923, 2025.

[5] S. Y. Alaba and J. E. Ball, "Deep Learning-Based Image 3-D Object Detection for Autonomous Driving: Review," *IEEE Sensors Journal*, vol. 23, no. 4, pp. 3378-3394, 2023.

[6] H. Li, J. Y. Wang, L. W. Xu, *et al*., "Efficient and Accurate Object Detection for 3D Point Clouds in Intelligent Visual Internet of Things," *Multimedia Tools and Applications*, vol. 80, no. 24, pp. 31297-31334, 2021.

[7] Q. P. Chen, Y. F. Xie, S. F. Guo, *et al*.,

"Sensing System of Environmental Perception Technologies for Driverless Vehicle: A Review of State of the Art and Challenges," *Sensors and Actuators A: Physical*, vol. 319, Art. no. 112566, 2021.

[8] N. Kurian and K. Vadivukkarasi, "Sensor Data Fusion Methods for Driverless Vehicle System: A Review," in *Lecture Notes in Networks and Systems*, vol. 475, pp. 333-344, 2023.

[9] H. Wang, J. Y. Li, and H. R. Dong, "A Review of Vision-Based Multi-Task Perception Research Methods for Autonomous Vehicles," *Sensors*, vol. 25, no. 8, p. 2611, 2025.

[10] J. Karangwa, J. Liu, and Z. X. Zeng, "Vehicle Detection for Autonomous Driving: A Review of Algorithms and Datasets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 11568-11594, 2023.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936-944.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788.

[13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint* arXiv:2004.10934, 2020.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2980-2988.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015, pp. 91-99.

[16] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[17] H. Caesar, V. Bankiti, A. H. Lang, *et al*., "nuScenes: A Multimodal Dataset for Autonomous Driving," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] P. Sun, H. Kretzschmar, X. Dotiwalla, *et al*., "Scalability in Perception for Autonomous Driving: Waymo Open Dataset," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] P. Sun, W. Wang, Y. Liu, *et al*., "PIXOR: Real-Time 3D Object Detection from Point Clouds," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] 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. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[21] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely Embedded Convolutional Detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[22] T. Yin, X. Zhou, and P. Krähenbühl, "CenterPoint: A Center-based 3D Object Detector," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021.

[23] V. A. Si dagi, Y. Zhou, and O. Tuzel, "MVX-Net: Multimodal VoxelNet for 3D Object Detection," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2019.

[24] Y. Liu, T. Tian, Y. Zhang, *et al*., "BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2023.

[25] K. Chitta, A. Prakash, and A. Geiger, "TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

