**Multimodal Fusion and Temporal Reasoning for Intelligent Robot Perception**

by

Jingyu Song

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Robotics)
in The University of Michigan
2026

Doctoral Committee:

Assistant Professor Katherine A. Skinner, Chair
Assistant Professor Bernadette Bucher
Assistant Professor Maani Ghaffari
Professor John Leonard, Massachusetts Institute of Technology

Jingyu Song
jingyuso@umich.edu
ORCID iD: 0000-0001-8458-4303

© Jingyu Song 2026

# DEDICATION

To those who have loved, supported, and shaped me.

# ACKNOWLEDGEMENTS

I would like to sincerely thank my PhD advisor, Prof. Katie Skinner, for her invaluable guidance, support, and mentorship throughout my PhD journey. I am also deeply grateful to Prof. Maani Ghaffari for his constant support and encouragement. I would like to thank the other members of my dissertation committee, Prof. Bernadette Bucher and Prof. John Leonard, for their guidance and insights, which helped shape this dissertation.

I am sincerely grateful to my friends, whom I feel so fortunate to have. My PhD journey has been more joyful and meaningful because of all of you, and I will always remember the moments we shared together.

I also want to thank my family. As I have grown older, I have come to realize how deeply my family has shaped who I am today. Looking back, it is truly amazing to see how far I have come, and I could not have done any of this without them.

Special thanks go to my wife. Thank you for always supporting me in everything I want to do, and for filling my life with warmth and joy through your smile.

Work throughout my graduate studies was supported by a number of generous funding sources. The TURTLMap work relates to Department of Navy award N00014-21-1-2149 issued by the Office of Naval Research. The LiRaFusion and CRKD work was supported by the Ford Motor Company via the Ford-UM Alliance under award N028603. The MemFusionMap work was completed while I was an intern at NVIDIA. The final stage of this dissertation was supported by the Rackham Predoctoral Fellowship Program from the Rackham Graduate School at the University of Michigan. I gratefully acknowledge these sources of support, which provided the resources, time, and collaborative environment that made this dissertation possible.

# TABLE OF CONTENTS

DEDICATION . . . . . ii

ACKNOWLEDGEMENTS . . . . . iii

LIST OF FIGURES . . . . . viii

LIST OF TABLES . . . . . xv

ABSTRACT . . . . . xx

CHAPTER

**1 Introduction . . . . . 1**

1.1 Motivation . . . . . 1
1.2 Robot Autonomy . . . . . 2
1.2.1 Localization and Mapping for Underwater Robots . . . . . 3
1.2.2 Robust Perception for Autonomous Vehicles . . . . . 3
1.3 Multimodal Fusion and Temporal Reasoning . . . . . 7
1.4 Problem Statement . . . . . 7
1.5 Contributions . . . . . 8

**2 Robust Localization and Dense Mapping of Low-texture Underwater Environments . . . . . 10**

2.1 Introduction . . . . . 10
2.2 Related Work . . . . . 12
2.2.1 Localization for Marine Robots . . . . . 12
2.2.2 Mapping in Underwater Scenes . . . . . 13
2.3 Method . . . . . 13
2.3.1 Sensor Integration . . . . . 14
2.3.2 Pose Graph Localization . . . . . 14
2.3.3 Real-time Volumetric Mapping . . . . . 17
2.4 Experiments . . . . . 18
2.4.1 Vehicle Setup . . . . . 18
2.4.2 Experimental Setup . . . . . 21
2.4.3 Evaluation Setup . . . . . 21
2.4.4 Localization Results . . . . . 22
2.4.5 Mapping Results . . . . . 25

2.4.6 Runtime . . . 26
2.5 Conclusion . . . 26
**3 Deep Adaptive LiDAR-Radar Fusion for 3D Object Detection . . . 28**
3.1 Introduction . . . 28
3.2 Related Work . . . 30
3.2.1 Radar Datasets for Autonomous Driving . . . 30
3.2.2 Multimodal 3D Object Detection . . . 30
3.2.3 Gated Network for Sensor Fusion . . . 31
3.3 Method . . . 32
3.3.1 Early Fusion . . . 34
3.3.2 Middle Fusion . . . 34
3.4 Experiments . . . 36
3.4.1 Experimental Setup . . . 36
3.4.2 Quantitative Results . . . 37
3.4.3 Qualitative Results . . . 40
3.4.4 Ablation Studies . . . 41
3.4.5 LiDAR-Camera-Radar Fusion . . . 41
3.5 Conclusion . . . 42
**4 Enhanced Camera-Radar Fusion with Cross-modality Knowledge Distillation . . . 43**
4.1 Introduction . . . 43
4.2 Related Work . . . 45
4.2.1 Multimodal 3D Object Detection . . . 45
4.2.2 Cross-modality Knowledge Distillation . . . 46
4.3 Method . . . 47
4.3.1 Model Architecture Refinement . . . 48
4.3.2 Cross-Stage Radar Distillation (CSRD) . . . 48
4.3.3 Mask-Scaling Feature Distillation (MSFD) . . . 49
4.3.4 Relation Distillation (RelD) . . . 50
4.3.5 Response Distillation (RespD) . . . 51
4.3.6 Overall Loss Function . . . 51
4.4 Experiments . . . 52
4.4.1 Experimental Setup . . . 52
4.4.2 Quantitative Results . . . 53
4.4.3 Ablation Studies . . . 54
4.4.4 Improvement Analysis . . . 58
4.4.5 Qualitative Results . . . 59
4.5 Conclusion . . . 62
**5 Temporal Reasoning for Online Vectorized HD Map Construction . . . 63**
5.1 Introduction . . . 63
5.2 Related Work . . . 65
5.2.1 Online Vectorized HD Map Construction . . . 65

5.2.2 Temporal Fusion for HD Map Construction . . . 66
5.3 Method . . . 67
5.3.1 Temporal Overlap Heatmap . . . 68
5.3.2 Memory Fusion Block . . . 70
5.3.3 Training Loss . . . 72
5.4 Experiments . . . 72
5.4.1 Datasets and Splits . . . 72
5.4.2 Evaluation Metrics . . . 73
5.4.3 Implementation Details . . . 73
5.4.4 Quantitative Results . . . 75
5.4.5 Ablation Studies . . . 77
5.4.6 Qualitative Results . . . 80
5.5 Conclusion . . . 83
**6 A Spatiotemporal Camera–Radar Backbone for Driving via Forecasting-Based Pretraining . . . 84**
6.1 Introduction . . . 84
6.2 Related Work . . . 87
6.2.1 Pretraining for Autonomous Driving . . . 87
6.2.2 Camera-Radar Fusion for Autonomous Driving . . . 88
6.2.3 Forecasting-Based World Models for Driving . . . 89
6.3 Preliminaries . . . 90
6.3.1 BEVFormer . . . 90
6.3.2 ViDAR . . . 91
6.4 Methodology . . . 92
6.4.1 Overview of CRISP . . . 92
6.4.2 Enhanced Radar Encoder . . . 93
6.4.3 Camera-Radar Spatiotemporal BEV Backbone . . . 95
6.4.4 Forecasting-Based Pretraining Objective . . . 100
6.4.5 Downstream Finetuning . . . 101
6.5 Experiments . . . 102
6.5.1 Experimental Setup . . . 102
6.5.2 Main Results . . . 103
6.5.3 Label-Efficiency of Pretraining . . . 108
6.5.4 Ablation Study . . . 109
6.5.5 Qualitative Analysis . . . 112
6.6 Discussion . . . 114
6.7 Conclusion . . . 117
**7 Conclusion . . . 118**
7.1 Conclusion . . . 118
7.2 Future Directions . . . 119
7.2.1 Unified Multimodal-Temporal Backbones for Heterogeneous Robot Sensors . . . 119
7.2.2 Semantics-Aware Temporal Memory for Dynamic Field Environments 120

7.2.3 Toward General-Purpose Robot Perception . . . . . 121

APPENDIX . . . . . 122

BIBLIOGRAPHY . . . . . 123

# LIST OF FIGURES

FIGURE

1.1 Representative real-world challenges that motivate robust robot perception. These include environmental disturbance from water motion, low-texture and low-visibility underwater scenes, adverse weather conditions such as snow, and sudden occlusions caused by nearby dynamic objects. Together, these examples show that robot perception systems must remain reliable when sensing is degraded, visual features are limited, or important scene regions are temporarily hidden. The adverse-weather image is reproduced from MIT News [23]; photo courtesy of the researchers, used under CC BY-NC-ND. The sudden-occlusion example is reproduced from Endo et al. [32], licensed under CC BY 4.0. . . . . 1

1.2 Failure cases of existing vision-based SLAM baselines in a low-texture underwater pool environment. We evaluate ORB-SLAM3 [10] and SVIn2 [144, 145] using image sequences collected in a controlled pool experiment. Here, SVIn2* denotes our reimplementation of SVIn2 on our robot platform with the sonar input disabled, so that it operates as a visual-inertial baseline. The pool environment contains large textureless surfaces, limited salient visual landmarks, and underwater illumination artifacts, which reduce the number and reliability of visual correspondences needed for tracking and initialization. (a) ORB-SLAM3 fails to maintain robust tracking because the underwater images contain insufficient repeatable features. (b) SVIn2* fails to initialize reliably and produces an unstable, diverging trajectory estimate. These examples illustrate why visual and visual-inertial SLAM pipelines do not directly transfer to low-texture underwater environments. . . . . 4

1.3 An example of a bird's-eye view (BEV) perception pipeline, with a camera-based model shown as an example. Other modalities also follow this paradigm, with different modality-specific encoders and view transformation modules. . . . . 5

1.4 A radar chart that lays out the strengths and weaknesses of common sensors (i.e., LiDAR, red-green-blue (RGB) camera, and mm-wave radar) used for autonomous vehicle (AV) perception. . . . . 5

1.5 Illustration of temporal fusion for online HD map construction. Surrounding camera views from two consecutive timesteps, $t-1$ and $t$, capture complementary observations of the local driving scene. The red boxes indicate a pedestrian crossing observed in the image space, which corresponds to the red pedestrian-crossing annotation in the ground-truth vectorized map. Temporal fusion enables information from adjacent frames to be integrated, supporting more complete and temporally consistent reconstruction of lanes, road boundaries, and pedestrian crossings. . . . . 6

2.1 Overview of TURTLMap. We propose a real-time localization and dense mapping solution for low-texture underwater environments with a low-cost underwater robot. 11

2.2 An overview of TURTLMap for real-time localization and dense mapping. We develop the system using the Robot Operating System (ROS) [125]. The localization stack is formulated as a pose graph optimization framework with DVL, IMU, and barometer measurements. The estimated pose is published as a transform message using the ROS tf tool. Voxblox [123] subscribes to the pose and the depth point cloud from the ZED mini camera to construct a dense map in real time. . . . . 14

2.3 A comparison of the stereo depth point cloud of a rock platform placed on the tank bottom with the default calibration (left) and the underwater calibration (right). The bottom surface of the tank is flat. . . . . 18

2.4 An overview of the robot platform and the customized sensor enclosure. The Blue Robotics Bar30 barometer is located in the electronics enclosure of the BlueROV2. The Waterlinked DVL A50 is placed on the payload skid with its transducers facing downward. The ZED mini stereo camera is placed inside the customized sensor enclosure with an NVIDIA Jetson Orin Nano embedded computer. We also place a LORD MicroStrain 3DM-GX5-25 IMU inside the sensor enclosure. Additional sensors (e.g., Blue Robotics Ping360 scanning sonar, Blue Robotics Ping1D sonar) are mounted on the robot, but they are not used in this work. . . . . 20

2.5 The towing tank at the University of Iowa. We show different regions for localization evaluation and mapping evaluation. The region where the MoCap system has valid tracking is relatively small. We exclude the artificial beach for map evaluation due to the lack of its reference structure. . . . . 21

2.6 Failure cases of existing vision-based SLAM baselines in a low-texture underwater pool environment. We evaluate ORB-SLAM3 [10] and SVIn2 [144, 145] using image sequences collected in a controlled pool experiment. Here, SVIn2* denotes our reimplementation of SVIn2 on our robot platform with the sonar input disabled, so that it operates as a visual-inertial baseline. The pool environment contains large textureless surfaces, limited salient visual landmarks, and underwater illumination artifacts, which reduce the number and reliability of visual correspondences needed for tracking and initialization. (a) ORB-SLAM3 fails to maintain robust tracking because the underwater images contain insufficient repeatable features. (b) SVIn2* fails to initialize reliably and produces an unstable, diverging trajectory estimate. These examples illustrate why visual and visual-inertial SLAM pipelines do not directly transfer to low-texture underwater environments. . . . . 23
2.7 A qualitative comparison of estimated trajectories for log 1 in the x-y plane shown in meters. The starting point indicates where the robot enters the MoCap's valid tracking zone. Our method maintains the most accurate trajectory with the least amount of drift over time. Best viewed with color and zoomed-in. . . . . 24
2.8 Qualitative results of the map generated from both logs. In log 1, we highlight the reconstructed bird-shaped logo and the corresponding image. The map matches the flat bottom surface well while including details of the artificial beach. In log 2, we plot a color map of distance to the bottom surface showing the capability of the proposed solution under wave disturbance. We also show a comparison of the rock dimensions captured from both logs. . . . . 26

3.1 We propose LiRaFusion to efficiently leverage the complementary information of LiDAR and radar for 3D object detection. . . . . 29
3.2 Overview of the architecture of LiRaFusion. Our main contributions, shown as bold text, mainly include a joint voxel feature encoder to extract per-voxel features from the stacked point cloud, and a gated network to learn weights for each input feature map to fuse them adaptively. . . . . 32
3.3 The network architecture of the early fusion module. LiDAR and radar points are zero-padded to a common feature space before voxelization: LiDAR points are represented as $[x, y, z, \text{intensity}, \Delta t_l, 0, 0, 0, 0]$, while radar points are represented as $[x, y, z, 0, 0, \text{rcs}, V_{x_{\text{comp}}}, V_{y_{\text{comp}}}, \Delta t_r]$. Here, $(x, y, z)$ denotes point coordinates, $\Delta t_l$ and $\Delta t_r$ denote the sweep time offsets, RCS denotes radar cross-section, and $V_{x_{\text{comp}}}$ and $V_{y_{\text{comp}}}$ denote compensated radar velocities. The joint voxel feature encoder aggregates the stacked LiDAR-radar points within each voxel cell and outputs voxel features for sparse convolution. . . . . 33

3.4 The network architecture of the middle fusion module. The LiDAR feature map $B \times C_1 \times H \times W$ and radar feature map $B \times C_2 \times H \times W$ are concatenated along the channel dimension and passed through modality-specific $3 \times 3$ convolution layers followed by sigmoid functions to generate channel-wise adaptive weights. The original LiDAR and radar feature maps are then element-wise multiplied by their corresponding weights to produce gated features, which are concatenated into the fused feature map $B \times (C_1 + C_2) \times H \times W$. Here, $B$ denotes the batch size, $C_1$ and $C_2$ denote the LiDAR and radar channel dimensions, and $H$ and $W$ denote the spatial height and width of the bird's-eye-view feature maps. . . . . 35
3.5 Example bounding box predictions and corresponding weight maps. We present two frames in which LiRaFusion correctly detects a car (highlighted with a red circle) that is missed by the baseline LO detector. We also show a zoomed-in view in which we label radar points in magenta, and LiDAR points in gray or red (if they reside in a bounding box). We show ground truth bounding boxes in blue and predictions in green. In the visualization of weight maps, the black bounding box with arrow denotes the ego-vehicle. Boxes without an arrow denote the highlighted missed car object. Best viewed in color and zoomed-in. . . . . 40

4.1 We propose CRKD to conduct a novel cross-modality knowledge distillation path from a LiDAR-Camera teacher to a Camera-Radar student. We present a radar chart to illustrate the complementary nature of these sensing configurations and the improvement that CRKD can enable. . . . . 44
4.2 An overview of the proposed cross-modality (LC-to-CR) distillation framework CRKD. We narrow down the modality gap by unifying both the teacher and the student into the BEV space with similar 3D object detector structure. We refine the model design to enable adaptive fusion and design four novel distillation losses for effective cross-modality KD. During inference, only CR input is needed. 47
4.3 Visualization of ungated and gated camera feature maps in the teacher detector. We visualize the normalized activation magnitude of the camera feature map in bird's-eye view (BEV), where brighter colors indicate stronger feature responses. The ungated feature map shows more diffuse activations, while the gated feature map produces more localized and spatially structured responses after incorporating information from the LiDAR branch. This qualitative comparison illustrates how the gated representation can provide a cleaner feature target for distillation than the original ungated camera feature. . . . . 57
4.4 Qualitative results on nuScenes. We show zoomed-in views in panel b and c for the highlighted regions in panel a, with dashed borders indicating the corresponding regions. We show the ground truth annotation in red, teacher prediction in green, student prediction in yellow, CRKD prediction in blue, and radar points in magenta. In (1a) to (1c), we show an example frame where CRKD has more accurate predictions and fewer false predictions than the student model. In (2a) to (2c) we show another example frame where CRKD even outperforms the LC teacher by detecting a missed car and rejecting several false predictions. Best viewed on screen and in color. . . . . 60

4.5 Additional qualitative results on nuScenes. We show zoomed-in views for the highlighted regions and compare the ground truth annotation (red), teacher prediction (green), student prediction (yellow), CRKD prediction (blue), and radar points (magenta). Samples 1 and 2 show that CRKD captures missed objects and rejects false positives from the student model. Samples 3 and 4 show cases where CRKD also rejects false detections produced by the teacher. Sample 5 shows a difficult frame where CRKD outperforms both teacher and student by recovering a missed object while keeping fewer false positives. . . . . 61

5.1 We propose MemFusionMap to improve the temporal reasoning capability for online vectorized HD map construction. MemFusionMap takes in multi-view images and leverages the proposed working memory fusion module and temporal overlap heatmap to output a vectorized HD map online. . . . . 64
5.2 Overall architecture of MemFusionMap. MemFusionMap takes in multi-view images and extracts a BEV feature via an image encoder and a BEV feature encoder. The encoded BEV feature is fused with working memory features and the temporal overlap heatmap to obtain a unified BEV feature. The unified feature is fed into a decoder head to generate HD map prediction. . . . . 67
5.3 Example propagation process of the temporal overlap heatmap from $t_0$ to $t_1$. . . 68
5.4 Example temporal overlap heatmaps resulting from different speeds. The comparison demonstrates that the overlap heatmap encodes vehicle moving speed. . 69
5.5 Example temporal overlap heatmap resulting from a vehicle turning. We highlight the two noticeable edges at the corner. This pattern encodes insights into the vehicle's trajectory. . . . . 70
5.6 Convergence comparison on the new nuScenes split at the extended $50\,m$ range. 77
5.7 We show qualitative results on nuScenes [8] at the extended $50\,m$ range. We show the input multi-view images on the left. In the HD maps, green denotes road boundaries, red denotes pedestrian crossings, and blue denotes lane lines. Best viewed in color. . . . . 79
5.8 Sequential qualitative results on nuScenes at the extended $100 \times 50\,m$ range. We show a consecutive sequence of five frames. The highlighted region shows a case where MemFusionMap avoids false predictions that accumulate in the baseline under severe temporal occlusion. In the HD maps, green denotes road boundaries, red denotes pedestrian crossings, and blue denotes lane lines. Best viewed in color. . . . . 81
5.9 Sequential qualitative results on nuScenes at the extended $100 \times 50\,m$ range. We show a consecutive sequence of five frames. The highlighted region shows a case where MemFusionMap consistently recovers a pedestrian crossing under long-lasting occlusion. In the HD maps, green denotes road boundaries, red denotes pedestrian crossings, and blue denotes lane lines. Best viewed in color. . . . . 82

6.1 Overview of the motivation of CRISP. We use low-cost camera-radar inputs for inference, while leveraging LiDAR as privileged supervision during pretraining to learn a transferable multimodal driving backbone. . . . . 85

6.2 Overview of CRISP. Historical multi-view images and aggregated radar observations are encoded into a shared CR BEV backbone. Radar first passes through ego-aware radar encoding, then enters the backbone through radar-enhanced temporal propagation and multimodal feature rendering. During pretraining, a future occupancy decoder is supervised by future LiDAR; during downstream finetuning, only the pretrained CR backbone is retained. . . . . 93
6.3 Radar-enhanced temporal self-attention. The calibrated radar BEV feature $F_t^{\text{rad}}$ primes the input BEV query $Q_t^{(l-1)}$ through the query-conditioned gate $g_{t,q}^{(l)}$, producing the radar-primed query $\hat{Q}_t^{(l)}$. Ego-aligned temporal deformable attention then attends to the historical BEV memory $B_{t-1}^{\text{mem}}$ and the current query branch. The temporal gate $g_{t,\text{temp}}^{(l)}$ combines current and historical outputs instead of using uniform averaging. . . . . 96
6.4 Multimodal feature rendering. Camera and radar cross-attention branches produce residual innovations, which are selectively admitted by the Modality Innovation Gate (MIG). The pre-attention BEV query is preserved through the residual path. . . . . 98
6.5 Label-efficiency of CRISP pretraining on nuScenes 3D detection. We train the same BEVFormer-CRISP detection model for 12 epochs using 1/4, 1/2, and the full nuScenes training split. The scratch baseline uses the same FCOS3D-pretrained camera encoder but does not use CRISP backbone pretraining. CRISP improves both NDS and mAP across all data scales, with clear gains in the low-label regime. . . . . 110
6.6 Qualitative comparison between CRISP and ViDAR for future world-geometry prediction in a low-light scene. We show the front camera views at the current frame (0s) and future timestamps for visualization only; future images are not provided to the model. On the right, predicted point clouds are compared with ground truth. In this example, the CO ViDAR baseline loses the forward vehicle over time and exhibits large spatial drift, while CRISP maintains a more accurate prediction of the vehicle trajectory. This illustrates the benefit of radar motion cues for forecasting dynamic objects under challenging illumination, which is important for downstream trajectory evaluation and safe planning around leading vehicles. . . . . 114
6.7 Qualitative comparison between CRISP and ViDAR for future world-geometry prediction. We show the front camera views at the current frame (0s) and future timestamps for visualization only; future images are not provided to the model. On the right, predicted point clouds are compared with ground truth, where CRISP better preserves the scene geometry over time. In this example, the CO ViDAR baseline propagates an erroneous structure from a billboard region, while CRISP maintains more reliable geometry. . . . . 115

6.8 Failure case analysis for future world-geometry prediction. We show the front camera views at the current frame (0s) and future timestamps for visualization only; future images are not provided to the model. On the right, predicted point clouds are compared with ground truth. This scene requires reasoning about a traffic light that is small and difficult to observe from the input images: the highlighted vehicle remains stopped for the red light, while the road ahead is visually open. CRISP predicts the vehicle geometry more accurately than the CO ViDAR baseline at early horizons, but its prediction begins to drift after 2s. This suggests that, when high-level traffic-signal intent is ambiguous, the forecasting model may rely more on learned motion priors from pretraining than on the underlying traffic-control state. . . . . 116

# LIST OF TABLES

TABLE


2.1 Details of the logs used in the evaluation of the proposed method. . . . . 22
2.2 Comparison of the localization accuracy of the proposed method (TURTLMap) and baselines. Abbreviations: DVL (D), IMU (I), and Barometer (B). $ATE_{pos}$ is measured in meters, and $ATE_{rot}$ is measured in degrees. Lower values indicate better performance, and the best result for each metric is **bolded**. . . . . 24
2.3 Ablation study of the proposed localization module in TURTLMap. $ATE_{pos}$ is measured in meters, and $ATE_{rot}$ is measured in degrees. Lower values indicate better performance, and the best result for each metric is **bolded**. . . . . 25
2.4 Quantitative mapping evaluation using different localization inputs. Mapping error is measured in meters. Lower values indicate better performance, and the best result for each log is **bolded**. . . . . 25

3.1 Results with TransHead [18] evaluated on nuScenes *val* set. EZFusion-LR* represents our re-implementation of [89]. All values are percentages. No model ensemble or test-time augmentation is used. . . . . 37
3.2 Results with CenterHead [206] evaluated on nuScenes *val* set. We train these networks with 7 moving classes. EZFusion-LR stands for the results in its original paper [89]. * represents our re-implementation. In this experiment group, test-time augmentation was used to keep consistency with EZFusion [89]. No model ensemble is used. All values are percentages. . . . . 38
3.3 Comparison with LR and LCR detectors on nuScenes *val* (**top**) and *test* (**bottom**) set. The reported LiRaFusion results use TransHead [18]. No test-time augmentation or model ensemble is used. Results for other models are the reported results in the papers so some entries are missing. All values are percentages; higher is better. . . . . 38
3.4 Runtime and True Positive (TP) metrics for experiments evaluated with TransHead [18]. Runtime is measured using frames per second (FPS). Higher is better for FPS; lower is better for TP error metrics. . . . . 38
3.5 Performance by object distance of experiments with TransHead [18]. We report the mAP separately in 3 ranges: $0\,m$-$20\,m$, $20\,m$-$30\,m$, and $30\,m$-$50\,m$. With increasing object distance, LiRaFusion demonstrates higher gain over FUTR3D-LO as radar complements LiDAR at far locations where LiDAR suffers from data sparsity. All values are percentages; higher is better. . . . . 39

3.6 Performance by weather conditions of experiments with TransHead [18]. We report results of two weather conditions: Sunny and Rainy. The grouping strategy is based on the official scene description entries in nuScenes [8]. There are in total 968 rainy frames out of 6019 frames. All values are percentages; higher is better. . . . . 39
3.7 Ablation study of fusion module. NDS, mAP, and AP values are percentages. . 41
3.8 Ablation study of gated network design. All values are percentages. . . . . 42
3.9 Results with LCR fusion and TransHead [18]. NDS and mAP values are percentages. . . . . 42

4.1 Comparison on the nuScenes dataset. We group methods based on modality and whether KD is used. We include existing state-of-the-art (SOTA) works that use single-frame image input for fair comparison. Methods [19, 180, 221] omitted from the *test*-split group do not report their results with single-frame image input. mAP and NDS are percentages. The proposed CRKD outperforms the baseline methods in most metrics. ◊ denotes the distillation-based methods. † denotes using test-time augmentation. The best results are **bolded**, and the second-best results are underlined. . . . . 52
4.2 Comparison of the per-class AP results of the BEVFusion-LC (teacher), BEVFusion-CR (baseline), BEVFusion-CR* (student) and CRKD models on the nuScenes *val* split. AP and mAP are percentages; higher is better. The best results are **bolded**. . . . . 54
4.3 Ablation study of the proposed modules in CRKD evaluated on the nuScenes *val* split. The baseline is the BEVFusion-CR model. mAP and NDS are percentages. The best results are **bolded**. . . . . 54
4.4 Ablation Study of single distillation modules on the nuScenes *val* split. mAP and NDS are percentages. . . . . 55
4.5 Ablation study of fusion module design and number of channels of the student model on the nuScenes *val* split. mAP and NDS are percentages. . . . . 55
4.6 Further ablation studies of Cross-Stage Radar Distillation (CSRD) on the nuScenes *val* split. mAP and NDS are percentages. . . . . 56
4.7 Further ablation studies of Mask-Scaling Feature Distillation (MSFD) on the nuScenes *val* split. mAP and NDS are percentages. . . . . 57
4.8 Further ablation study of Response Distillation (RespD) with different class-weight settings on the nuScenes *val* split. mAP and NDS are percentages. . . . 58
4.9 Performance breakdown by range on the nuScenes *val* split. We report the gain of CRKD over the student model. mAP and NDS are percentages. . . . . 59
4.10 Performance breakdown by weather and lighting on the nuScenes *val* split. We report the gain of CRKD over the student model. mAP and NDS are percentages. 59

5.1 Performance comparison with baseline methods on the new nuScenes split under two bird's-eye-view perception ranges. AP and mAP are reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric within each perception range is **bolded**. Abbreviations: pedestrian crossing (ped), lane divider (div), and road boundary (bound). R50 denotes a ResNet-50 backbone. For StreamMapNet [210], we report our reproduced results, which match the results reported in the original paper. . . . . 73
5.2 Performance comparison with baseline methods on the new Argoverse2 split under two bird's-eye-view perception ranges. AP and mAP are reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric within each perception range is bolded. Abbreviations: pedestrian crossing (ped), lane divider (div), and road boundary (bound). R50 denotes a ResNet-50 backbone. For StreamMapNet [210], we report our reproduced results, which match the results reported in the original paper. . . . . 74
5.3 Quantitative comparison of map prediction accuracy and runtime on the new nuScenes split at the $30\,m$ perception range. mAP is reported in percentage points (%), FPS is reported in frames per second, and ↑ indicates that higher is better. The best result for each metric is **bolded**. MapTracker [15] lacks support for the extended $50\,m$ range setting. SQD-MapNet [172] is not included because it is not open-source. . . . . 76
5.4 Performance comparison on the original splits of nuScenes and Argoverse2 at the $30\,m$ perception range. mAP is reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric within each dataset is **bolded**. * indicates that we follow the StreamMapNet code repository and train for 30 epochs on the original split. † indicates results reported in the SQD-MapNet [172] paper. . . . . 76
5.5 Ablation study of the proposed components in MemFusionMap evaluated on the new nuScenes split at the $50\,m$ perception range. Starting from the StreamMapNet baseline in row (a), row (b) removes the GRU-based BEV temporal fusion module to evaluate the importance of temporal fusion. Row (c) adds the proposed convolutional working memory fusion module, row (d) further introduces dilated convolution to enlarge the receptive field, and row (e) adds the temporal overlap heatmap to guide memory aggregation. mAP is reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric is **bolded**. . . . . 78
5.6 Ablation study of the working memory capacity in MemFusionMap. $T_{\mathrm{WM}}$ denotes the number of historical BEV feature frames maintained in working memory. mAP is reported in percentage points (%), FPS is reported in frames per second, and ↑ indicates that higher is better. The best result for each metric is **bolded**. 78
5.7 Ablation study of the convolution dilation choice in the working memory fusion module. The convolution dilation is denoted as $(d_H, d_W)$, where $d_H$ and $d_W$ are the dilation factors along the height and width dimensions of the BEV feature map, respectively. mAP is reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric is **bolded**. . . . . 79

5.8 Ablation study of the non-linear activation function used in the temporal overlap heatmap feature branch on the new nuScenes split at the $100 \times 50\,m$ BEV perception range. The compared activation functions are applied after the temporal overlap heatmap feature branch to produce the heatmap-based feature representation. mAP is reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric is **bolded**. . . . . 79

6.1 Point cloud forecasting on nuScenes following the ViDAR protocol [203]. We report Chamfer Distance (CD) for future timestamps from 0.5s to 3.0s. The best result is shown in **bold** and the second-best result is underlined. † indicates a language-augmented world model [229] with an LLM-based component and additional text supervision, which is different from the sensor-only pretraining setting of the other listed methods. . . . . 104
6.2 3D detection results. All metrics are reported in percentage. The best result is shown in **bold** and the second-best result is underlined. . . . . 105
6.3 Online map segmentation results. All IoU values are percentages. The best result is shown in **bold** and the second-best result is underlined. . . . . 106
6.4 Online tracking results. AMOTA is reported as a percentage, AMOTP in meters, and Recall as a ratio. The best result is shown in **bold** and the second-best result is underlined. . . . . 106
6.5 Motion forecasting results. minADE and minFDE are reported in meters. The best result is shown in **bold** and the second-best result is underlined. . . . . 108
6.6 Future occupancy prediction results. All values are percentages. The best result is shown in **bold** and the second-best result is underlined. . . . . 108
6.7 End-to-end planning results. Avg. $\mathcal{L}_2$ is reported in meters, and Avg. Collision is reported as a percentage. The best result is shown in **bold** and the second-best result is underlined. . . . . 109
6.8 Main ablation study of the proposed CRISP modules. Starting from the ViDAR baseline, each subsequent row cumulatively adds one additional component on top of the previous row: radar-enhanced TSA, multimodal feature rendering, and the enhanced radar encoder. mAP and NDS are reported in percentage points (%), and mAVE is reported in meters per second (m/s). The best result for each metric is **bolded**. . . . . 111
6.9 Ablation study of the radar encoder design. Starting from the PointPillars radar encoder, each subsequent row cumulatively adds one additional component on top of the previous row: ego-motion awareness and residual spatial calibration. All other CRISP modules are kept the same for fair comparison. mAP and NDS are reported in percentage points (%), and mAVE is reported in meters per second (m/s). The best result for each metric is **bolded**. . . . . 112
6.10 Ablation study of design choices in radar-enhanced temporal self-attention. Starting from standard TSA, each subsequent row cumulatively adds gated radar conditioning and temporal gating. All other CRISP modules are kept the same for fair comparison. mAP and NDS are reported in percentage points (%), and mAVE is reported in meters per second (m/s). The best result for each metric is **bolded**. . . . . 113

6.11 Ablation study of modality fusion design in multimodal feature rendering. All other CRISP modules are kept the same for fair comparison. mAP and NDS are reported in percentage points (%), and mAVE is reported in meters per second (m/s). The best result for each metric is **bolded**. . . . . 113

# ABSTRACT

Reliable autonomy for field robots depends on perception systems that can operate under difficult sensing conditions. In real-world environments, robot perception is often degraded by low-texture visual patterns, environmental disturbances, adverse weather, occlusions, and sensor failures. This dissertation develops multimodal fusion and temporal reasoning methods that improve the robustness, scalability, and accuracy of robot perception across challenging environments.

The first part of this thesis addresses state estimation and dense mapping for underwater robots, where wave disturbance and low-texture environments often cause vision-based localization to fail. We introduce TURTLMap, a real-time localization and dense mapping framework for low-cost underwater robots. TURTLMap fuses Doppler velocity log, inertial, and pressure measurements for robust localization, while using stereo depth to construct dense 3D maps. Real-world experiments in a water tank environment, evaluated with underwater motion capture and a reference 3D structure, demonstrate accurate robot tracking and mapping under low-texture and wave-disturbed conditions.

The second part studies adaptive multimodal fusion for autonomous vehicle perception. We introduce LiRaFusion, a LiDAR-Radar fusion network that combines joint feature encoding with adaptive feature weighting to better exploit the complementary strengths of LiDAR and radar. Experiments on large-scale 3D object detection benchmarks show that this design improves detection performance over existing fusion methods. Building on this direction, we develop CRKD, a cross-modality knowledge distillation framework that transfers knowledge from a high-performing LiDAR-Camera teacher to a scalable Camera-Radar student. This approach provides a practical pathway for using high-quality sensor data from test fleets to improve cost-effective sensing configurations for consumer vehicles, achieving state-of-the-art Camera-Radar object detection performance.

The third part explores temporal reasoning for road scene understanding. We introduce MemFusionMap, a memory-based framework for online vectorized HD map construction that improves temporal fusion by combining current bird's-eye-view (BEV) features with multiple working-memory features. MemFusionMap further maintains a temporal overlap heatmap, which provides a spatiotemporal cue for how historical observations overlap with the current

field of view and helps the model reason over memory more adaptively. Together, these designs improve map construction under challenging and complex road conditions, including occlusion and dynamic scene changes, while preserving efficient runtime and compatibility with multiple perception models.

Finally, this thesis develops CRISP, a spatiotemporal Camera-Radar pretraining framework for autonomous driving. CRISP learns transferable BEV representations by forecasting future LiDAR point clouds from historical camera and radar observations, using LiDAR as privileged supervision only during pretraining. At deployment, the model operates using Camera-Radar inputs alone. Experiments on real-world benchmarks show that CRISP improves long-horizon point cloud forecasting and transfers effectively to downstream tasks including 3D object detection, tracking, online mapping, motion forecasting, future occupancy prediction, and planning.

Together, these contributions show how multimodal sensing, cross-modality knowledge transfer, temporal memory, and predictive pretraining can make robot perception more reliable under practical sensing constraints. The resulting methods improve localization, mapping, perception, prediction, and planning across challenging underwater and autonomous driving environments.

# CHAPTER 1

# Introduction

## 1.1 Motivation

Achieving full autonomy for intelligent robots operating in complex and challenging field environments requires robust perception capabilities. However, reliable perception remains a critical challenge due to the limitations of individual sensor modalities, susceptibility to environmental disturbances, and the dynamic nature of real-world environments [205, 114, 7, 158, 159]. This thesis focuses on addressing these challenges by developing effective perception solutions that leverage multimodal fusion and temporal reasoning.

Figure 1.1 presents several real-world challenges that motivate this thesis. For instance, in real-world deployment for underwater robots, camera-based localization and mapping algorithms face challenges such as visibility constraints, water dynamics, and sparsity of visual features. On land, autonomous vehicles frequently encounter sensor failures or degraded




Figure 1.1: Representative real-world challenges that motivate robust robot perception. These include environmental disturbance from water motion, low-texture and low-visibility underwater scenes, adverse weather conditions such as snow, and sudden occlusions caused by nearby dynamic objects. Together, these examples show that robot perception systems must remain reliable when sensing is degraded, visual features are limited, or important scene regions are temporarily hidden. The adverse-weather image is reproduced from MIT News [23]; photo courtesy of the researchers, used under CC BY-NC-ND. The sudden-occlusion example is reproduced from Endo et al. [32], licensed under CC BY 4.0.

performance under adverse weather conditions, which can lead to failures or inaccuracies in tasks such as object detection and mapping for single-modality perception systems. Thus, it is crucial to conduct *multimodal sensor fusion* effectively, combining complementary sensor strengths to achieve enhanced robustness and accuracy. Another fundamental challenge facing mobile robots is degraded perceptual capabilities in complex scenarios such as limited visibility, frequent occlusions, and intricate scene dynamics. For robots operating in these environments, instantaneous sensor readings are often insufficient for reliable perception. These conditions demand the integration of historical information, necessitating effective *temporal reasoning* strategies to enable perception systems to maintain contextual continuity, resolve occlusions, and reason effectively about complex and evolving scenes.

Yet, current multimodal fusion and temporal reasoning methods still struggle to support robust autonomous robot operations in highly complex, dynamic, and unpredictable real-world operational environments. Practical challenges include addressing noisy and asynchronous sensor data, achieving adaptive fusion of complementary modalities, maintaining a high level of scalability, and improving memory capacity against memory corruption. This thesis addresses these needs by advancing state-of-the-art perception algorithms in multimodal fusion and temporal reasoning, significantly enhancing robotic perception capabilities across diverse and challenging operational domains.

## 1.2 Robot Autonomy

Autonomous robots play an important role in a wide range of tasks, from simple home assistance to complex field work, highlighting their crucial role in enhancing efficiency, safety, and productivity in many areas. Among all the modules (e.g., perception, planning, control) that support autonomous robot operations, perception enables the robot to understand its surrounding environments and assists downstream modules in making informed decisions [134, 28]. As robots are increasingly deployed in diverse environments, from urban landscapes to underwater exploration, achieving reliable autonomy presents significant obstacles in the domain of perception. This dissertation mainly focuses on marine robotics and autonomous vehicles – fields that may seem different but share fundamental challenges in creating reliable perception systems. Both domains demand robust solutions that can handle adverse conditions while contributing to important societal goals, from marine conservation through efficient underwater surveys to eliminating road accidents with robust perception capabilities. The following subsections discuss the perception challenges specific to these two domains.

### 1.2.1 Localization and Mapping for Underwater Robots

Among various environments where autonomous robots are deployed, marine environments are receiving increasing attention. This interest is driven by their profound societal impacts and the unique challenges they pose, distinct from those in terrestrial environments. In underwater environments, robotic systems have been prominent in a broad range of applications such as environmental monitoring, exploration, marine sample collection, and infrastructure inspection [144, 219, 222, 68, 2]. Robust sensing and perception are necessary to enable autonomous operation for such applications. One focus area of this thesis is a critical task in accurately estimating a marine robot's trajectory and constructing a map of its environment. The tasks of localization and mapping are typically executed simultaneously, a process widely known as simultaneous localization and mapping (SLAM) [7, 147]. SLAM involves continuously estimating the robot's position while simultaneously constructing a coherent environmental representation, enabling effective navigation and exploration. Despite the significant advancements and widespread success of SLAM methodologies in terrestrial environments [120, 10, 142, 148], the unique challenges presented by underwater scenarios – such as limited visibility, sparse visual features in low-texture environments, varying lighting conditions, and significant environmental disturbance – often render conventional methods ineffective or unreliable [7, 152, 218, 158]. We show some failure cases of popular SLAM methods in Figure 1.2. To solve these challenges, underwater SLAM methods commonly use multimodal fusion, integrating visual, acoustic, and inertial sensor data [144, 222, 68, 33]. Measurements from these modalities are typically combined within a graph optimization framework, which systematically fuses temporal measurements to enhance accuracy and robustness [10, 148, 27, 28]. By leveraging multiple sensor inputs and historical data, these approaches effectively address sensor limitations and environmental challenges, significantly improving underwater localization and mapping performance.

### 1.2.2 Robust Perception for Autonomous Vehicles

Autonomous vehicles (AVs) must reliably perceive their surroundings under diverse and challenging conditions to enable effective and safe planning and control [40, 206]. Common perception tasks such as object detection, tracking, and mapping usually focus on perceiving the surrounding environments in the real-world 3D coordinate system [114, 8, 160]. Thanks to advances in deep learning neural networks [80, 77, 47, 164] and the vast amount of open-source datasets for AVs [40, 160, 8], modern AV perception models have become increasingly data-driven and learning-based. These methods can learn expressive feature representations from large amounts of annotated data and have achieved strong performance on benchmark

perception tasks. However, robust generalization to rare events, adverse weather, sensor degradation, and out-of-distribution environments remains an open challenge, motivating the development of more robust perception systems.

Early explorations in AV perception adopted paradigms such as voxel-based methods [111, 231] and frustum-based approaches [126, 121], each providing valuable insights but encountering limitations related to computational complexity, scalability, and adaptability to different scenarios. Recently, bird's-eye view (BEV)-based representation has emerged as a predominant paradigm, providing a more intuitive and efficient representation for percep-

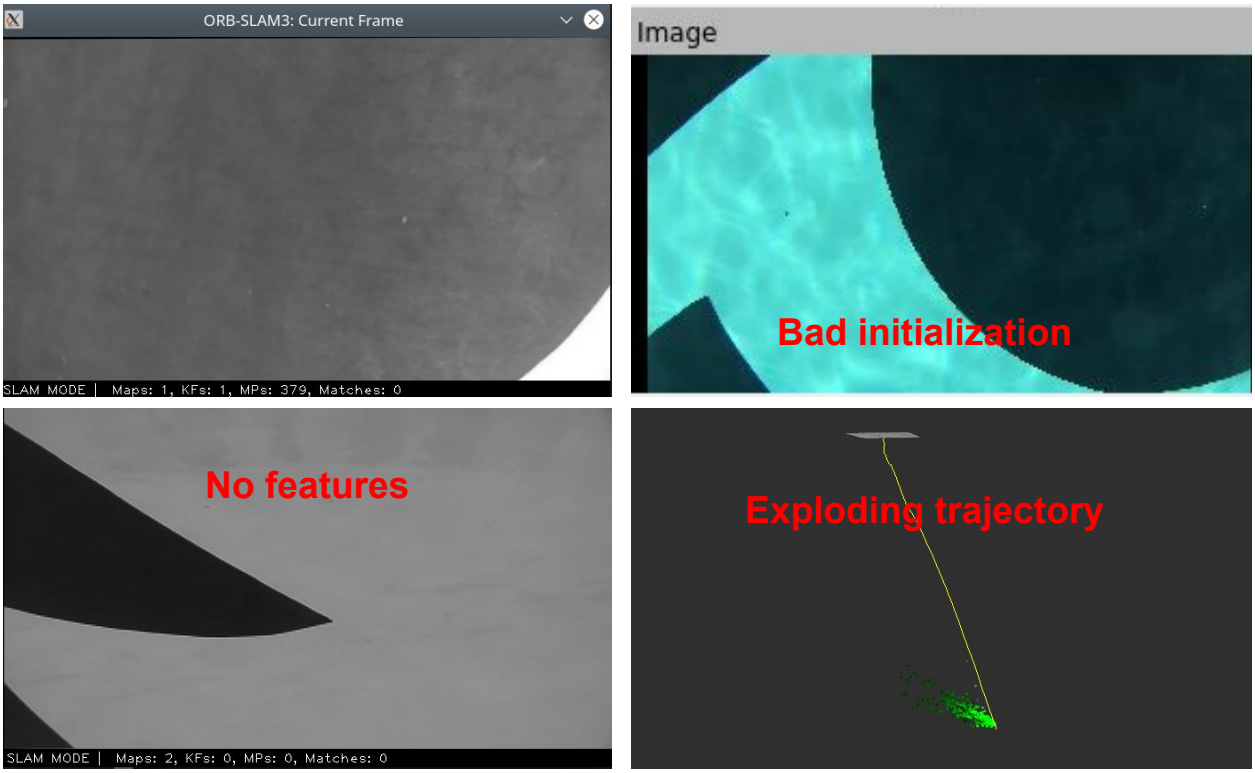



Figure 1.2: Failure cases of existing vision-based SLAM baselines in a low-texture underwater pool environment. We evaluate ORB-SLAM3 [10] and SVIn2 [144, 145] using image sequences collected in a controlled pool experiment. Here, SVIn2* denotes our reimplementation of SVIn2 on our robot platform with the sonar input disabled, so that it operates as a visual-inertial baseline. The pool environment contains large textureless surfaces, limited salient visual landmarks, and underwater illumination artifacts, which reduce the number and reliability of visual correspondences needed for tracking and initialization. (a) ORB-SLAM3 fails to maintain robust tracking because the underwater images contain insufficient repeatable features. (b) SVIn2* fails to initialize reliably and produces an unstable, diverging trajectory estimate. These examples illustrate why visual and visual-inertial SLAM pipelines do not directly transfer to low-texture underwater environments.

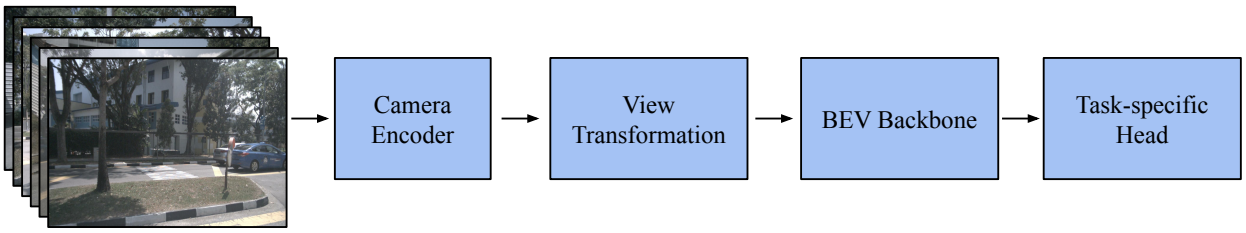



Figure 1.3: An example of a bird's-eye view (BEV) perception pipeline, with a camera-based model shown as an example. Other modalities also follow this paradigm, with different modality-specific encoders and view transformation modules.

tion tasks [54, 90, 94, 45, 107]. As shown in Figure 1.3, the general BEV-based perception paradigm typically consists of four main components: a modality-specific encoder for initial feature extraction, a view transformation module to transform features from sensor-centric to BEV space, a BEV-specific backbone to further process features, and a task-specific head to perform perception tasks such as detection or semantic segmentation.

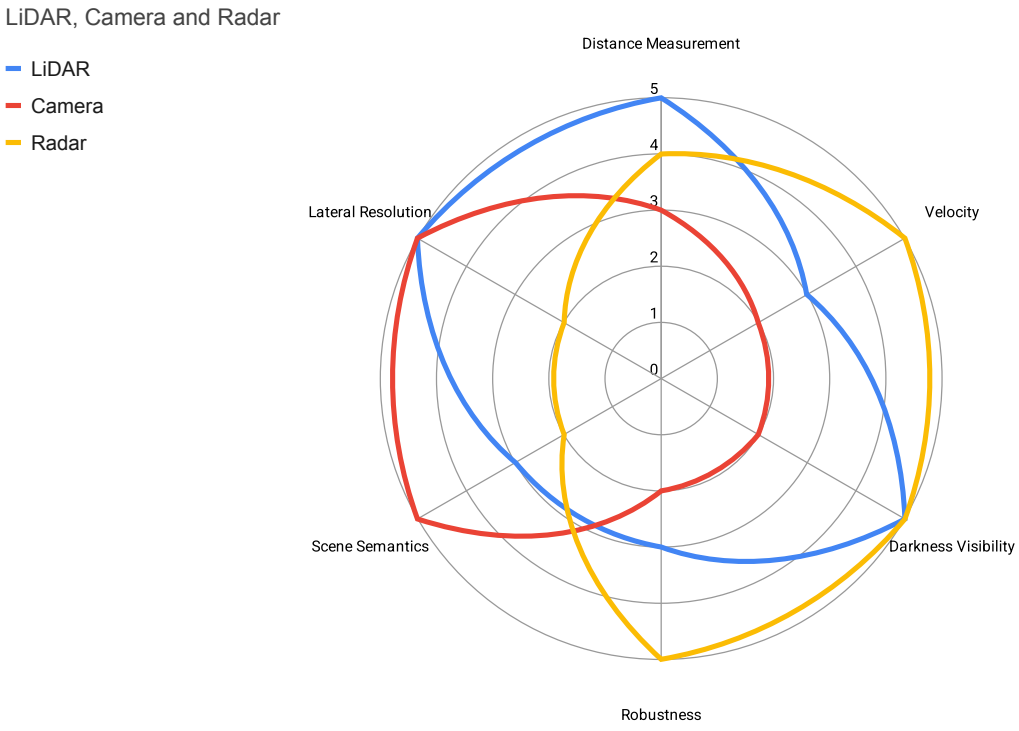



Figure 1.4: A radar chart that lays out the strengths and weaknesses of common sensors (i.e., LiDAR, red-green-blue (RGB) camera, and mm-wave radar) used for autonomous vehicle (AV) perception.

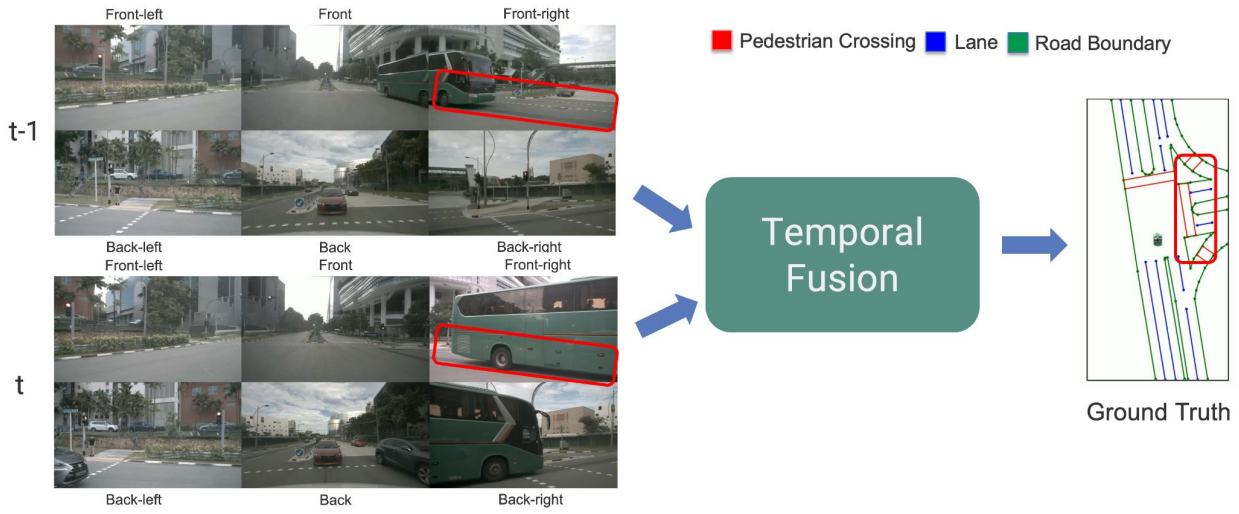



Figure 1.5: Illustration of temporal fusion for online HD map construction. Surrounding camera views from two consecutive timesteps, $t-1$ and $t$, capture complementary observations of the local driving scene. The red boxes indicate a pedestrian crossing observed in the image space, which corresponds to the red pedestrian-crossing annotation in the ground-truth vectorized map. Temporal fusion enables information from adjacent frames to be integrated, supporting more complete and temporally consistent reconstruction of lanes, road boundaries, and pedestrian crossings.

Another key advantage of BEV representation is that it serves as a unified bridge for different modalities, which simplifies the multi-sensor fusion process. Common sensors used for AV perception include RGB cameras, LiDAR, and millimeter-wave radar. Though recent camera-based and LiDAR-based BEV methods have achieved impressive performance on perception benchmarks, these single-modality methods still experience various challenges due to individual sensor limitations. Figure 1.4 shows that no single-modality solution can excel in all aspects. Therefore, there is a pressing need to conduct multimodal fusion to further improve the robustness and reliability of perception models, which can be facilitated by the unified BEV representation.

In addition, many existing perception models are limited to single-frame input, facing a significant challenge of not leveraging temporal information [54, 107, 45]. These models are prone to errors due to challenging environmental conditions, including occlusions, complex intersections, and extreme camera exposures, which commonly occur in autonomous driving scenarios [171, 43, 94, 210]. We show an example in Figure 1.5 to demonstrate the benefits of having temporal reasoning capability for perception models. Consequently, designing an effective temporal fusion strategy is of great significance for expanding the deployment of AVs.

## 1.3 Multimodal Fusion and Temporal Reasoning

Inspired by the strong potential of achieving robot autonomy and the existing gaps in robust perception research, this thesis focuses on improving the perceptual capabilities of intelligent robots through multimodal fusion and temporal reasoning. In general, multimodal fusion refers to integrating information from diverse sensors. Given each sensor modality provides unique advantages and suffers from distinct limitations, multimodal fusion has the benefits of leveraging complementary information. For instance, LiDAR excels in depth measurement and geometric accuracy but struggles with adverse weather conditions such as heavy rain or fog. Radar, conversely, performs reliably in adverse weather and provides valuable velocity information but lacks spatial resolution. Cameras offer high-resolution visual information with semantics yet degrade significantly in poor lighting or visibility conditions. Acoustic sensors, such as sonars, are particularly useful in visually limited environments, such as underwater, but are challenged by noise, lower spatial resolution, and potential interference. Effective multimodal fusion is expected to amplify their complementary strengths while mitigating individual weaknesses.

Temporal reasoning involves integrating sequential data over multiple time steps, allowing perception systems to accumulate historical observations and contextual cues. This accumulation aids in overcoming short-term sensor disruptions, maintaining scene continuity, resolving occlusions, and understanding complex motion patterns in challenging environments. For example, mobile robots operating in crowded urban scenes benefit from temporal reasoning by accurately tracking moving objects, predicting future positions, and updating maps dynamically. In underwater environments, fusing temporal information improves estimates of sensor bias for state estimation and aids the construction of global maps for navigation tasks.

Together, multimodal fusion and temporal reasoning enhance a robot's capability to interpret complex, changing environments reliably. These two concepts unite this dissertation work across marine robotics and AVs and have the potential to create a robust and effective perception model to facilitate the deployment of autonomous robots.

## 1.4 Problem Statement

This thesis addresses the following research questions relevant to perception for autonomous robots:

- How can we design a scalable low-cost solution for robust localization and real-time dense mapping for robots operating in challenging low-texture underwater environ-

ments with wave disturbance?

- What strategies can enable adaptive fusion of complementary perception modalities to magnify the strength of individual sensors?
- How can we transfer knowledge from a high-performance but non-scalable sensor configuration to a scalable multimodal sensor configuration effectively?
- How can we improve temporal reasoning capabilities of robots operating in complex and dynamic environments?
- How can we design a unified multimodal perception backbone that learns transferable spatial-temporal representations from scalable sensor configurations and supports robust perception, prediction, and planning in dynamic open-world environments?

## 1.5 Contributions

The main objective of this dissertation is to improve perception capabilities of robots by proposing effective algorithms for multimodal fusion and temporal reasoning. To achieve this objective and address the above research questions, this dissertation presents the following contributions:

1. Development of **TURTLMap**, a novel low-cost solution enabling real-time robust localization and dense mapping in wave-disturbed, low-texture underwater environments by fusing multimodal sensor measurements in the proposed framework. (Chapter 2)
2. Introduction of **LiRaFusion**, an adaptive multimodal fusion framework that fuses LiDAR and radar data to enhance perception for AVs operating in dynamic traffic environments. LiRaFusion sets a new standard for integrating LiDAR and radar data that outperforms the existing solutions. In particular, LiRaFusion delivers significant improvement in challenging scenarios such as long-range regions and rainy conditions. (Chapter 3)
3. Creation of **CRKD**, the first framework that leverages cross-modality knowledge distillation to optimize the performance of Camera-Radar perception systems. CRKD demonstrates state-of-the-art performance in Camera-Radar 3D object detection by distilling knowledge from a LiDAR-Camera teacher model. Notably, CRKD bridges the gap between high-fidelity LiDAR-Camera sensor configurations and cost-effective Camera-Radar systems. (Chapter 4)

4. Development of **MemFusionMap**, a novel temporal fusion framework that enhances BEV road scene understanding for AVs through working memory. MemFusionMap adaptively synthesizes historical features with current observations using a temporal overlap representation, improving temporal reasoning without sacrificing runtime efficiency. MemFusionMap achieves strong performance across road scene understanding tasks and can be integrated into various existing perception models. (Chapter 5)

5. Development of **CRISP**, a spatiotemporal Camera-Radar pretraining framework that learns a transferable 3D foundation backbone for AVs. CRISP uses forecasting-based world-model pretraining to predict future LiDAR point clouds from historical camera and radar observations, while requiring only Camera-Radar inputs during deployment. Through an enhanced radar encoder, radar-enhanced temporal self-attention, and multimodal feature rendering, CRISP captures joint Camera-Radar representations, temporal scene dynamics, and future-oriented 3D structure. CRISP improves downstream tasks including 3D object detection, tracking, online mapping, motion forecasting, future occupancy prediction, and end-to-end planning. (Chapter 6)

Appendix A provides further details of open-source software in support of this dissertation.

# CHAPTER 2

# Robust Localization and Dense Mapping of Low-texture Underwater Environments

## 2.1 Introduction

Robotic systems play an important role in improving our understanding of underwater environments [184, 75]. Among them, unmanned underwater vehicles (UUVs) stand out for their role in environmental monitoring and ocean exploration [75, 141]. To safely conduct these tasks, UUVs are required to have accurate and robust localization (i.e., state estimation) capabilities [141, 144, 145, 222, 158]. However, achieving such accurate localization for UUVs is non-trivial due to the lack of Global Positioning System (GPS) and Wi-Fi while the vehicle is underwater [144, 145].

The past few decades have seen great advances in localization systems, enabling robots on land to achieve real-time state estimation in complex environments [100, 208]. These methods can be categorized into odometry or simultaneous localization and mapping (SLAM) techniques [100, 66, 217, 60, 120, 10, 78]. With these methods, vision can be combined with inertial measurements to improve accuracy and robustness [37]. However, underwater environments pose unique challenges to achieving similar performance for marine robotic systems compared to terrestrial systems equipped with visual or visual-inertial sensor configurations [58, 144, 145, 112, 64]. For vision-based localization algorithms, common challenges include visibility change, light and color attenuation, absence of light, and feature sparsity [34, 124]. Despite these challenges, recent work has been successful in integrating vision into a robust localization system [68, 222]. SVin2 [144, 145] is one of the state-of-the-art underwater SLAM methods fusing camera, sonar, inertial measurement unit (IMU), and a barometer for improved accuracy and robustness. However, it still struggles in low-texture regions, which are often encountered in underwater environments. To tackle these challenges, additional modalities are usually introduced. The Doppler velocity log (DVL) is a popular acoustic sensor supporting underwater relative navigation. Existing methods have

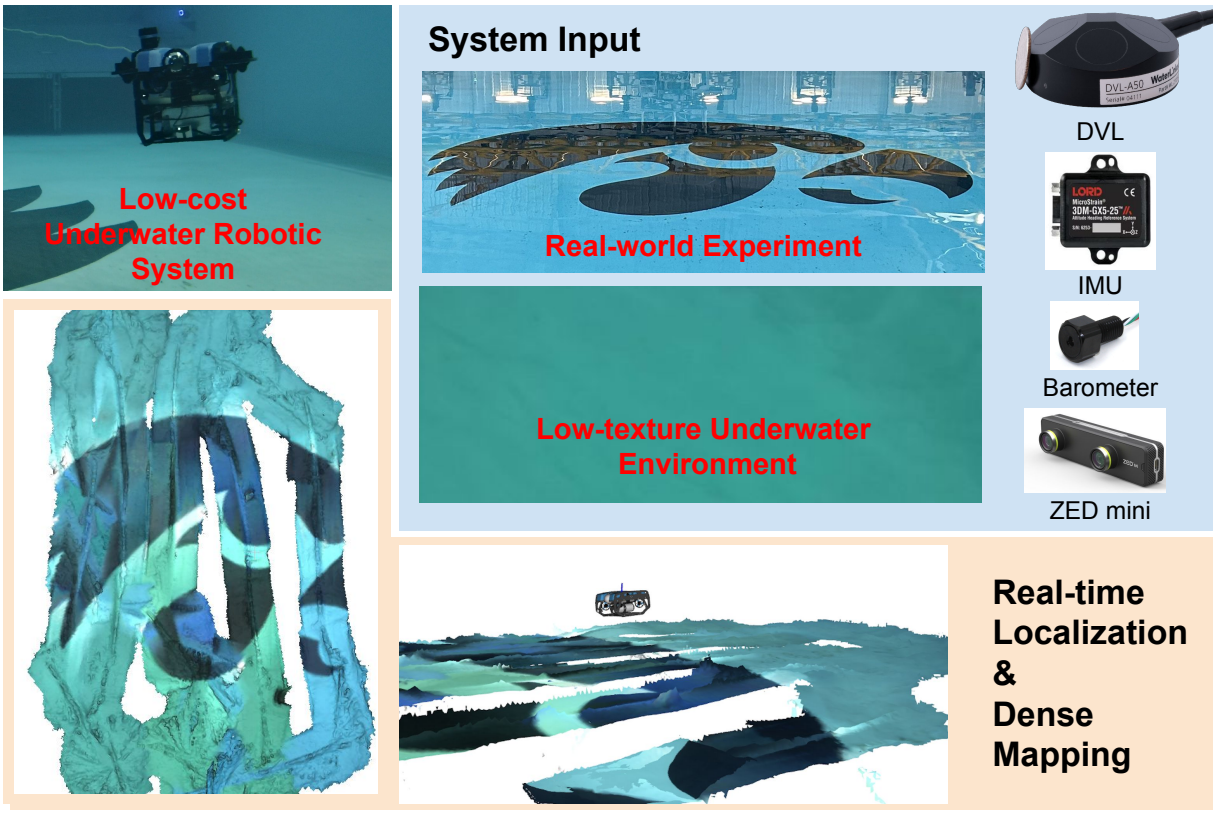



Figure 2.1: Overview of TURTLMap. We propose a real-time localization and dense mapping solution for low-texture underwater environments with a low-cost underwater robot.

seen success of fusing DVL, inertial, and pressure sensors for robust odometry with satisfying performance [158, 136]. Xu et al. [194] fuse camera and DVL in a SLAM framework with improvement in low-texture environments, but they do not address dense mapping. Researchers in [175] propose a real-time dense mapping solution built on [144, 145]. However, the method's reliance on vision-based state estimation poses open challenges in general underwater environments with low-texture scenes.

In this chapter, we propose and demonstrate **TURTLMap**, a novel framework focusing on **T**extureless **U**nderwater environments through a **R**eal-**T**ime **L**ocalization and **Map**ping method (Figure 2.1). In these environments, visual sensors are usually unreliable for navigation. To summarize, our main contributions are as follows:

- We propose a pose graph-based DVL-inertial-barometer localization package that achieves improved localization accuracy in low-texture regions.
- With the proposed localization package and a real-time volumetric mapping package, we demonstrate real-time and accurate dense mapping on an onboard embedded

computer on a low-cost UUV with a downward-facing stereo camera in a low-texture underwater environment.

- We provide extensive quantitative and qualitative evaluation of localization and mapping performance with ground truth localization provided by an underwater motion capture system and reference mapping provided by a 3D computer-aided design (CAD) model of the experimental site with and without wave conditions. Our contribution includes a detailed description of the integrated hardware, and we provide open-source software supporting future implementation and extension for the research community, which has been made available here: https://umfieldrobotics.github.io/TURTLMap.

## 2.2 Related Work

### 2.2.1 Localization for Marine Robots

Localization (i.e., state estimation) is one of the key tasks for underwater robots [144, 145, 222]. SLAM-based methods have been widely adopted to the underwater domain and they typically fuse vision, inertial sensors and/or acoustic sensors [68, 194, 144, 145]. The current state-of-the-art underwater SLAM system, SVin2, presents a multi-sensor fusion approach that leverages digital pipe profiling sonar, visual and inertial sensors, and a barometer [144, 145]. SVin2 relies on feature detection from camera images for its visual front-end [66]. Though it fuses other modalities such as sonar and IMU, the localization module is still vision-oriented, making it difficult to scale to featureless (low-texture) underwater environments [175, 63].

To reduce reliance on visual features for localization, it is common to integrate a DVL to provide seafloor-relative velocity. Researchers in [194] fuse vision with DVL odometry to achieve robust localization. Our method instead relies on DVL, IMU, and barometer sensors for enhanced localization robustness in low-texture environments with wave conditions. We additionally integrate a vision-based mapping module capable of real-time dense reconstruction. The sensor configuration in [222] is similar to ours but in a reversed upward-looking orientation for under-ice exploration. This method proposes fused odometry with camera, DVL, IMU, and a pressure sensor for a robot navigating in a low-texture under-ice environment. It differs from our method as it assumes planar features and slowly-varying altitude to aid visual tracking, which could be valid in its target scenario (i.e., under ice exploration) but does not apply to general underwater low-texture environments. Inspired by the success of recent DVL-based odometry methods [158, 136, 100], we design a pose graph-based state estimation framework fusing DVL, IMU and pressure sensors via GTSAM [27]. Since our

method does not require visual input for localization, there is no need for specific assumptions and aid for visual features, making it robust in low-texture environments with reliable localization accuracy. Another similar work [161] formulates a DVL factor and fuses it with an existing LiDAR-Camera SLAM algorithm for a surface vehicle doing maritime infrastructure inspection. Our method takes inspiration from its DVL factor design and further develops a real-time framework supporting additional sensors (i.e., barometer) for UUVs in an underwater setting.

### 2.2.2 Mapping in Underwater Scenes

There has been prominent work on underwater mapping, with works ranging from acoustic-based [158], large-scale mapping [162], to close-range visual mapping [144, 145, 62, 152, 218]. In this section, we review works relating to vision-based mapping, with an emphasis on real-time or dense mapping methods used in underwater robotics applications. Methods that enable real-time mapping in underwater environments typically utilize SLAM-based formulations [82], where the visual component plays a crucial role in both localization and mapping. In SVin2 [144, 145], a camera-based system is fused with sonar to produce a feature-based map. The produced map is sparse and relies on a feature-rich environment for denser and more accurate mapping. Wang et al. [175] leverage the localization output from SVin2 to develop a state-of-the-art method for real-time dense 3D mapping in underwater environments. However, experiments on real test sites demonstrate open challenges, showing that both localization and mapping performance suffer in environments that lack rich features, such as when the vehicle moves over a sandy region. Our proposed method overcomes this challenge by leveraging multimodal sensors for robust localization in low-texture regions, which enables dense mapping with a real-time visual mapping framework. In particular, we leverage Voxblox [123] to achieve real-time dense 3D mapping following its successful application in other robotics fields [26].

## 2.3 Method

Figure 2.2 shows an overview of the proposed system for localization and mapping in low-texture underwater environments. TURTLMap takes the measurements from all navigation sensors (i.e., DVL, IMU, and barometer) as input to the localization stack. The mapping module subscribes to the estimated pose and stereo camera depth map to conduct real-time dense mapping. Each component of the system will be discussed in more detail below.

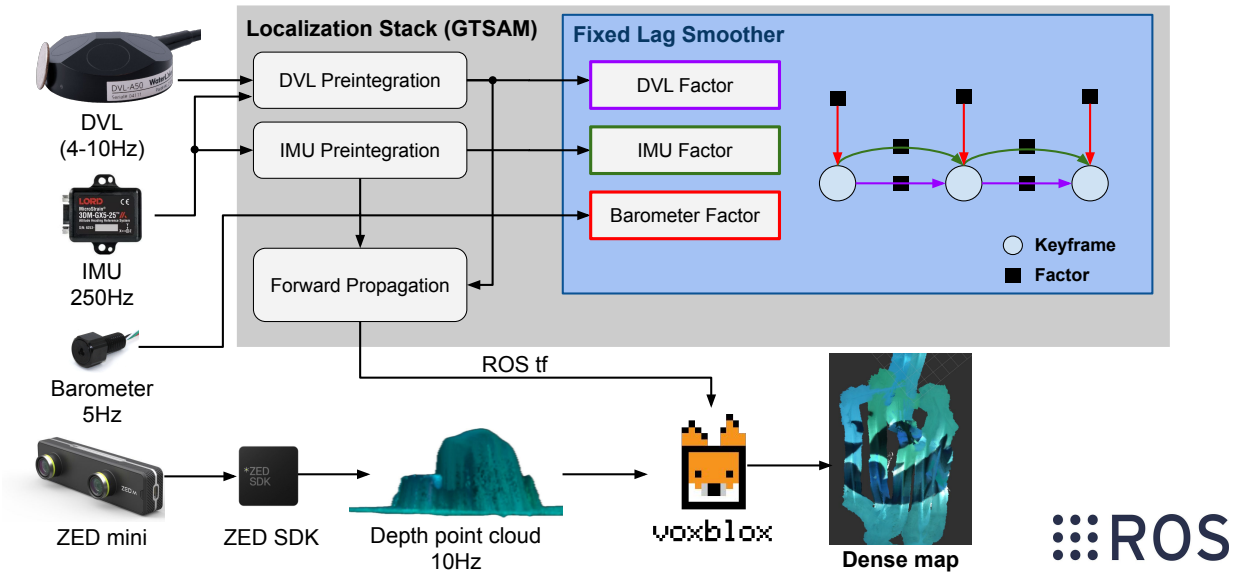



Figure 2.2: An overview of TURTLMap for real-time localization and dense mapping. We develop the system using the Robot Operating System (ROS) [125]. The localization stack is formulated as a pose graph optimization framework with DVL, IMU, and barometer measurements. The estimated pose is published as a transform message using the ROS tf tool. Voxblox [123] subscribes to the pose and the depth point cloud from the ZED mini camera to construct a dense map in real time.

### 2.3.1 Sensor Integration

Our underwater robotic system is equipped with a DVL, an IMU, and a barometer for navigation, and a stereo camera for dense mapping. The DVL provides direct measurement of the linear velocity of the vehicle, while the IMU measures the angular velocity and acceleration. The barometer measures the depth of the vehicle relative to the water surface based on the water pressure. For mapping, the stereo camera provides dense depth as a point cloud. The measurements from the aforementioned sensors are handled in ROS [125]. Detailed hardware setup will be introduced in section 2.4.1.

### 2.3.2 Pose Graph Localization

We formulate the state estimation problem as a factor graph with our integrated DVL, barometer, and IMU as navigation sensors. Specifically, we implement a keyframe-based maximum-a-posteriori DVL-inertial-barometer estimator in GTSAM [27]. We follow the state-of-the-art Kimera visual-inertial odometry (Kimera-VIO) [148] for framework design and add customized factors for the DVL and barometer, discussed in more detail below.

The state of the system at time $i$ can be described as:

$$\mathbf{x}_i = [\mathbf{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i^g, \mathbf{b}_i^a, \mathbf{b}_i^v]. \tag{2.1}$$

We assume that the IMU frame coincides with the body frame, and use the prior extrinsic calibration to handle measurements taken in different sensor frames. We define $\mathbf{R}_i \in \mathrm{SO}(3)$ as the vehicle rotation, $\mathbf{p}_i \in \mathbb{R}^3$ as the vehicle position, $\mathbf{b}_i^g, \mathbf{b}_i^a \in \mathbb{R}^3$ as the biases of IMU gyroscope and accelerometer measurements, and $\mathbf{b}_i^v \in \mathbb{R}^3$ as the linear velocity bias for the DVL. The rotation $\mathbf{R}_i$ and position $\mathbf{p}_i$ are expressed in the world coordinate frame using the north-east-down (NED) convention, while other state variables are expressed in the body frame. In our approach, we estimate the state of all keyframes:

$$\mathcal{X}_k = \{\mathbf{x}_i\}_{i \in \mathcal{K}_k}, \tag{2.2}$$

where we denote $\mathcal{K}_k$ as the set of all keyframes up to time $k$. This design maintains a sparse pose graph that ensures fast runtime. We follow Kimera-VIO [148] and VILENS [188] to use a fixed-lag smoother with a pre-defined time horizon. The fixed-lag smoother can help to bound the estimation time by marginalizing states that fall out of the smoothing horizon, which is important to ensure consistent real-time performance over long experimental duration.

We set up a ROS node to subscribe to different sensor drivers for their measurements. We assign an individual thread for IMU data since it is published at a high frequency (i.e., 250 Hz). The IMU measurements are preintegrated using the implementation of [37] in GTSAM [27]. We take inspiration from [188] to design a preintegration function for DVL measurements. We also design a barometer factor to supervise the vehicle depth.

For every keyframe, we add a preintegrated IMU factor, a preintegrated DVL factor and a barometer factor. It is important to handle the sensor synchronization appropriately. Since IMU measurements have a high frame rate, every DVL and barometer measurement has a matched IMU measurement within tolerable time difference. In our sensor setup, the frame rate of the barometer is around 5 Hz, while the DVL frequency can vary between 4 and 15 Hz depending on the altitude. We insert a new keyframe based on the time elapsed from the last keyframe using the barometer message timestamp. This strategy ensures an accurate measurement for the vehicle depth for every keyframe, which fully leverages the advantage of the barometer.

Furthermore, as the DVL message frequency can be as low as 4 Hz, the time gap between the latest DVL measurement and the current keyframe is not negligible. By assuming the IMU propagation can estimate an accurate velocity change over a short time horizon (i.e., max 0.25 second), we propose to use the IMU preintegration to obtain the latest velocity for

the current keyframe and add that as a pseudo measurement to the DVL preintegration.

#### 2.3.2.1 IMU Factor

We follow the existing works [148, 188] to use the preintegrated IMU factor to constrain the pose and velocity between two consecutive keyframes. This idea is first proposed in [37] to remove the need for recomputation of the integration between two keyframes. We use the same preintegration design implemented in GTSAM [27] to handle the high frequency signals. The parameters (e.g., noise density, noise random walk) used in IMU preintegration are set according to the data sheet of the IMU. As mentioned above, we also set up an additional IMU preintegration for estimating the velocity change since the latest DVL measurement. This additional preintegration is reset every time a new DVL measurement is received. We refer to [37] for more details about preintegrated IMU factors.

#### 2.3.2.2 Barometer Factor

The barometer provides accurate depth measurements based on the water pressure. Since the robot state is defined in the NED frame, the depth of the robot at time $i$ is defined as

$$d_i = p_i^z - p_0^z, \tag{2.3}$$

where $d_i$ is the depth compensated by the initial depth, and $p_i^z$ and $p_0^z$ are the $z$ axis coordinates of $\mathbf{p}_i$ and $\mathbf{p}_0$, respectively. For the depth measurement $z_i$, we define the residual as

$$r_{d_i,z_i} = d_i - z_i. \tag{2.4}$$

The barometer is integrated as a unary factor on $\mathbf{p}_i$, and its Jacobian used in our factor implementation is

$$\frac{\partial r_{d_i,z_i}}{\partial \delta \mathbf{p}_i} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \times \mathbf{R}_i, \tag{2.5}$$

which is a $1 \times 3$ vector derived by following the lifting strategy introduced in [37].

#### 2.3.2.3 DVL Factor

The DVL measures the linear velocity of the robot. We follow common practices [161, 188, 158] to use the constant velocity model for velocity integration. Unlike [188, 161], we decouple the IMU preintegration with the DVL preintegration and use the estimated altitude change from the IMU's onboard Extended Kalman Filter (EKF). According to our analysis and comparison with ground truth, the estimated altitude change from the onboard EKF

of our IMU is accurate in the short time gap between consecutive DVL measurements. In addition, decoupling the IMU preintegration reduces the system computation load and can leverage the fine-tuned onboard attitude reference system. We also directly leverage the reported DVL velocity measurement uncertainty from the manufacturer driver to inform the DVL factor of the measurement quality. As we observe that the uncertainty of DVL measurements can increase considerably due to multiple reasons (e.g., strong environmental disturbance such as waves, fast change of altitude in scenarios such as driving from beach to sea, out of operating range), this design has advantages of tuning the weight for DVL factors in the pose graph adaptively.

We follow [158, 188] to add a slowly varying bias term $\mathbf{b}^v$ to the velocity:

$$\tilde{\mathbf{v}} = \mathbf{v} + \mathbf{b}^v + \boldsymbol{\eta}^v, \tag{2.6}$$

where $\eta^v$ is the linear velocity Gaussian white noise term. We formulate the DVL factor as a translation residual between two consecutive keyframes. The translation measurement is defined as:

$$\Delta\tilde{\mathbf{p}}_{ij} = \sum_{k=i}^{j-1} \Delta\tilde{\mathbf{R}}_{ik}(\tilde{\mathbf{v}}_k - \mathbf{b}_i^v)\Delta t, \tag{2.7}$$

where $k$ is an intermediary measurement between keyframes $i$ and $j$. As mentioned above, we decouple the IMU preintegration by directly using the $\Delta\tilde{\mathbf{R}}_{ik}$ measurement from the IMU's onboard EKF. Our proposed DVL factor residual is formulated as

$$\mathbf{r}_{\mathcal{D}_{ij}} = \mathbf{R}_i^\intercal(\mathbf{p}_j - \mathbf{p}_i) - \Delta\tilde{\mathbf{p}}_{ij}(\mathbf{b}_i^v). \tag{2.8}$$

The Jacobians for the involved state variables $\mathbf{p}_i$ and $\mathbf{p}_j$ can be derived following [37]. We follow [161, 188] to implement the Jacobian for $\mathbf{b}_i^v$. We compute the covariance of $\mathbf{r}_{\mathcal{D}_{ij}}$ iteratively,

$$\boldsymbol{\Sigma}_{i,k+1}^{\mathcal{D}} = \mathbf{A}\boldsymbol{\Sigma}_{i,k}^{\mathcal{D}}\mathbf{A}^\top + \mathbf{B}\boldsymbol{\Sigma}_{\eta}^{\mathcal{D}}\mathbf{B}^\top, \tag{2.9}$$

where the first term $\boldsymbol{\Sigma}_{i,i}^{\mathcal{D}}$ starts from $\mathbf{0}$, and $\boldsymbol{\Sigma}_{\eta}^{\mathcal{D}}$ is set as the measurement uncertainty from the DVL driver report. We derive $\mathbf{A} = \mathbf{I}_{3\times3}$ and $\mathbf{B} = \Delta\tilde{\mathbf{R}}_{ik}\Delta t$. We refer to [161, 188] for a more detailed derivation process.

### 2.3.3 Real-time Volumetric Mapping

We leverage Voxblox [123], a real-time volumetric mapping library that can run on embedded devices, as the mapping algorithm of this work. Benefiting from its tight ROS integration, the integration of Voxblox with the proposed localization package is straightforward. As

shown in Figure 2.2, Voxblox directly subscribes to the depth measurements of the ZED mini camera. It leverages the ROS tf tool to query the estimated pose of the camera output by our robust localization system to reconstruct the scene using the mesh representation.

We use the neural mode of the depth estimation in the ZED SDK, which is the most accurate and stable depth sensing mode according to our analysis. In our experiment, we set the camera resolution to VGA and the publishing rate of the depth point cloud to 10 Hz. These settings can be adjusted according to the computation power of the embedded device. One of the important factors affecting the mapping performance is the camera calibration. We recalibrate the stereo camera for underwater usage. We use the multiple camera calibration tool in the Kalibr toolbox [116] and a customized Aprilgrid target to obtain the underwater calibration parameters. We demonstrate a comparison of the depth point cloud between default calibration and the underwater calibration in Figure 2.3. Note that the tank bottom should appear to be flat, so this validates the calibration using prior knowledge of the reference structure.

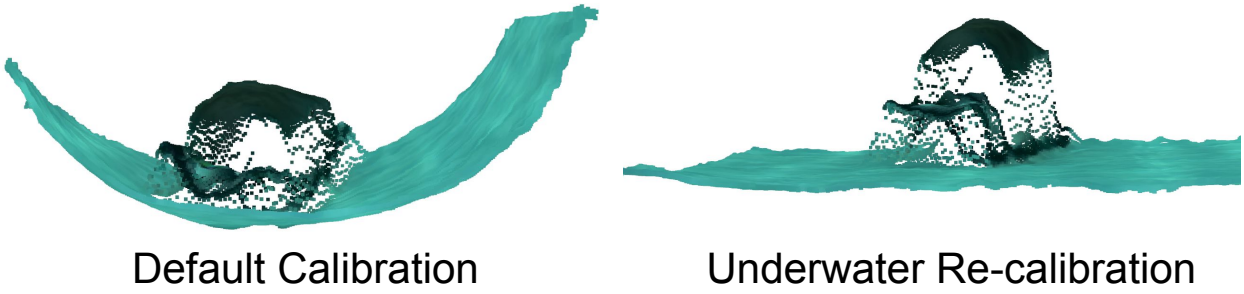



Figure 2.3: A comparison of the stereo depth point cloud of a rock platform placed on the tank bottom with the default calibration (left) and the underwater calibration (right). The bottom surface of the tank is flat.

## 2.4 Experiments

### 2.4.1 Vehicle Setup

We set up the proposed TURTLMap package based on a low-cost robotic system with our proposed sensor configuration as discussed below.

#### 2.4.1.1 Robot Platform

We build the proposed solution based on the heavy configuration of the Blue Robotics BlueROV2 UUV, which is an affordable, high-performance remotely operated vehicle (ROV) with high customization flexibility. We use the official payload skid to accommodate the added hardware items. To accommodate the components that are not waterproof, we include a customized Sexton aluminum enclosure and mount it onto the payload skid of the BlueROV2. The proposed localization and mapping package runs on the NVIDIA Jetson Orin Nano embedded computer. Figure 2.4 shows an overview of the robot platform with the customized sensor enclosure.

#### 2.4.1.2 Hardware Configuration

When choosing among various navigation and perception sensors, as well as the embedded computer, the most important considerations are the cost, size, and difficulty of integration. We choose to use the combination of DVL, IMU, and barometer to build the localization stack. We choose the Waterlinked DVL A50 model because it is a low-cost option among existing DVLs on the market and has a compact size. Another advantage of this DVL is the official integration guide and support for the BlueROV2. Though the DVL is equipped with an internal IMU, its data is inaccessible. We only use the velocity measurements from the DVL to maintain compatibility of the proposed method with various DVL models. Though the BlueROV2 comes with an IMU, its frame rate is limited by MAVLink (10 Hz by default), which is a major challenge for robust state estimation. We add an additional LORD MicroStrain 3DM-GX5-25 IMU to estimate the rotation of the vehicle. This IMU is reasonably priced and has an onboard EKF for accurate attitude measurement. The robot also comes with a Blue Robotics Bar30 barometer that has absolute measurements of robot depth. We access its measurements through communication with the BlueROV2 Raspberry Pi via MAVLink for the proposed state estimation package.

Though pose tracking via visual sensors is not reliable in underwater low-texture environments, we value its importance in dense reconstruction of these environments (aided by robust localization). We choose the ZED mini stereo camera due to its reasonable price and small form factor, which fit our budget and limited payload space. We place the ZED mini camera in a downward-facing configuration in the custom waterproof enclosure. The ZED mini camera is accompanied by the ZED SDK, providing useful functions for robotic applications (e.g., image capturing, depth sensing). In this work, we use the depth measurements from the ZED SDK for mapping. It is worth mentioning that the IMU inside the ZED mini camera is not used in this project due to its data quality and low frame rate.

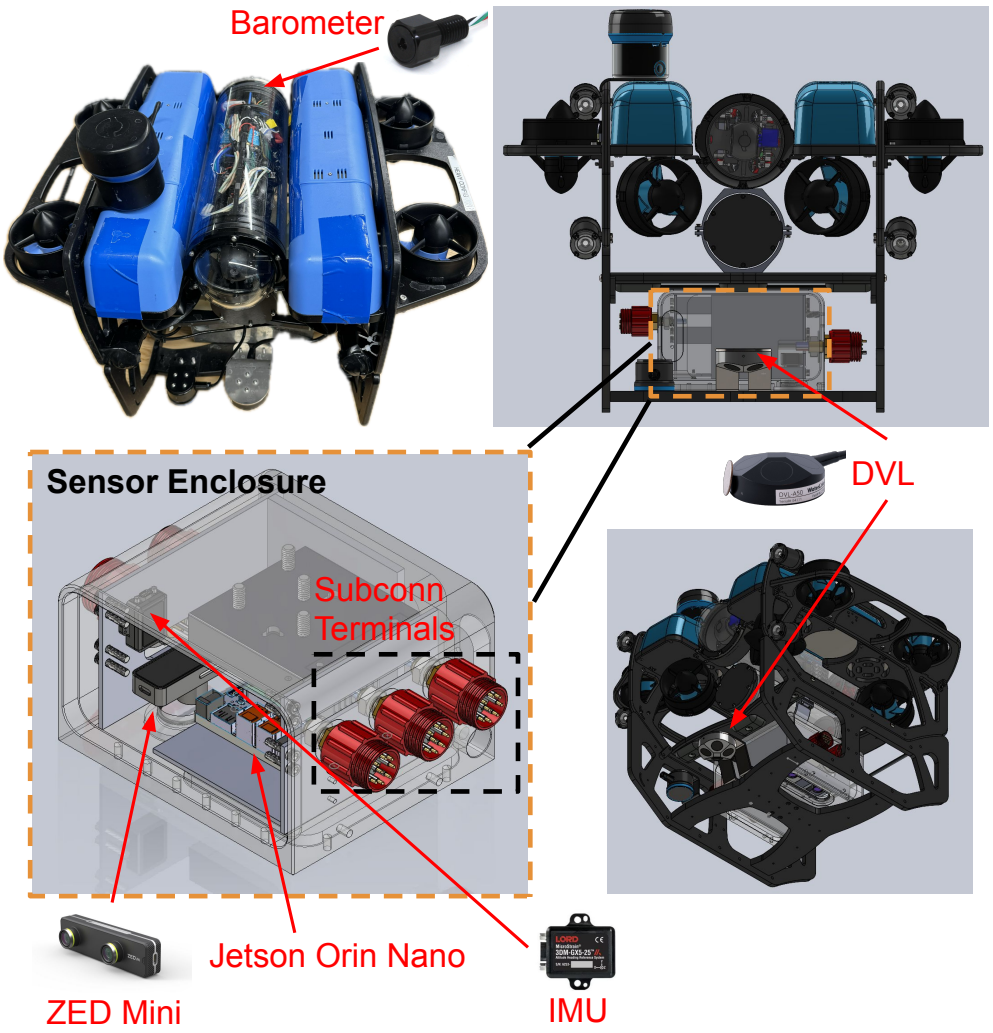



Figure 2.4: An overview of the robot platform and the customized sensor enclosure. The Blue Robotics Bar30 barometer is located in the electronics enclosure of the BlueROV2. The Waterlinked DVL A50 is placed on the payload skid with its transducers facing downward. The ZED mini stereo camera is placed inside the customized sensor enclosure with an NVIDIA Jetson Orin Nano embedded computer. We also place a LORD MicroStrain 3DM-GX5-25 IMU inside the sensor enclosure. Additional sensors (e.g., Blue Robotics Ping360 scanning sonar, Blue Robotics Ping1D sonar) are mounted on the robot, but they are not used in this work.

The ZED SDK requires an embedded computer with a GPU. We choose to use an NVIDIA Jetson Orin Nano due to the low cost. It also has a compact design that can fit in the sensor enclosure easily, and we have tested that it has stable thermal performance when placed in an enclosure.

We maintain proper calibration to ensure accurate localization and mapping. We use the IMU-camera calibration function in Kalibr toolbox [39, 38] to obtain the transformation between the IMU and the ZED mini stereo camera. We leverage the CAD model of the

robot platform to obtain the extrinsics for other sensors.

### 2.4.2 Experimental Setup

To demonstrate the capability of the proposed framework, experiments were conducted in the towing tank at the University of Iowa. We use the wave generator to create natural environmental disturbances to mimic real-world challenges for localization. The towing tank is an ideal place to evaluate the proposed system due to the equipped underwater motion capture (MoCap) system that can be used as the ground truth reference for vehicle trajectory and a reference structure of the tank for validating the mapping accuracy. We show an overview of the tank in Figure 2.5, which also indicates regions we use for evaluating the localization and mapping modules.

### 2.4.3 Evaluation Setup

We use two logs collected with different wave and driving patterns, as listed in Table 2.1. To evaluate localization performance, we use the open-source tool [220] to evaluate absolute trajectory error (ATE) between the output of our localization algorithm and the MoCap trajectory. As noted in Figure 2.5, the MoCap system does not have full coverage of the

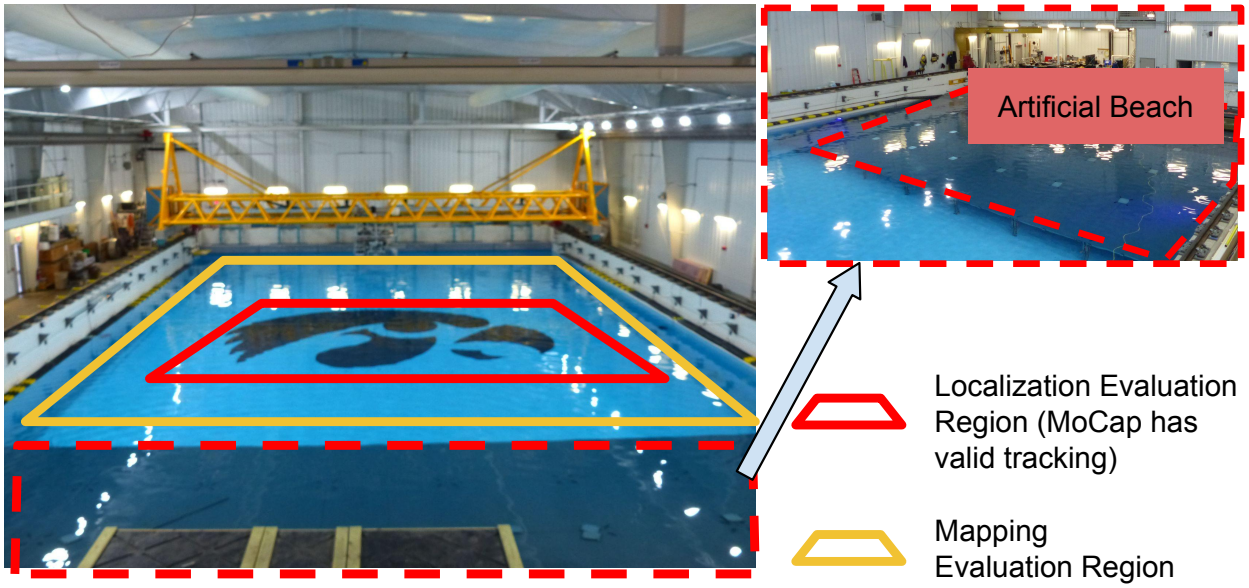



Figure 2.5: The towing tank at the University of Iowa. We show different regions for localization evaluation and mapping evaluation. The region where the MoCap system has valid tracking is relatively small. We exclude the artificial beach for map evaluation due to the lack of its reference structure.

tank, so we only conduct evaluation in the region where the MoCap system has valid tracking of the robot. As each log is initiated at the artificial beach region, which is outside of the MoCap tracking zone, we align the reference frame of our localization package to the MoCap reference frame according to the first valid MoCap pose. We run state-of-the-art vision-based SLAM methods, ORB-SLAM3 [10] and SVin2 [144, 145], as baselines to show the challenges that vision-based localization methods face in low-texture underwater environments. We further compare our proposed localization method against the onboard DVL dead reckoning results (DVL Odom.), and a baseline Unscented Kalman Filter (UKF) implemented in [158].

We use the ground truth mesh reference from the CAD model to evaluate the generated dense map in Cloud Compare [1]. Though the artificial beach is captured in our logs, due to the lack of the actual reference structure of the beach, we only evaluate the map with the bottom surface of the tank, as indicated in Figure 2.5. We also put an artificial rock platform with known 3D structure in the tank to validate the capability of the proposed solution in mapping small objects. In our experiments, we follow the default configuration of Voxblox [123] setting the voxel size to 0.05 m.

### 2.4.4 Localization Results

The underwater low-texture environments pose a significant challenge to the tested state-of-the-art SLAM methods. ORB-SLAM3 [10] has a hard time passing the initialization stage with its stereo or stereo-inertial mode. Though we try to tune the related parameters and apply Contrast Limited Adaptive Histogram Equalization (CLAHE) image preprocessing following [144, 145], due to the limited number of tracked features, the trajectory quickly diverges and the map is reset. A similar issue happens with SVin2 [144, 145], as we also notice the quickly diverging trajectory. We show examples of difficulties with these two baselines in Figure 2.6. This demonstrates that state-of-the-art visual SLAM systems are not reliable for navigation in low-texture underwater environments.

Table 2.1: Details of the logs used in the evaluation of the proposed method.

| Log | Wave | Description | Duration (s) |
|---|---|---|---|
| 1 | ✗ | Robot surveys the tank while being maintained about 1.75 m to the bottom | 352 |
| 2 | ✓ | Wave Height: 0.1 m<br>Wave Frequency: 1 Hz<br>Robot drives two squares and maintains at the water surface | 280 |

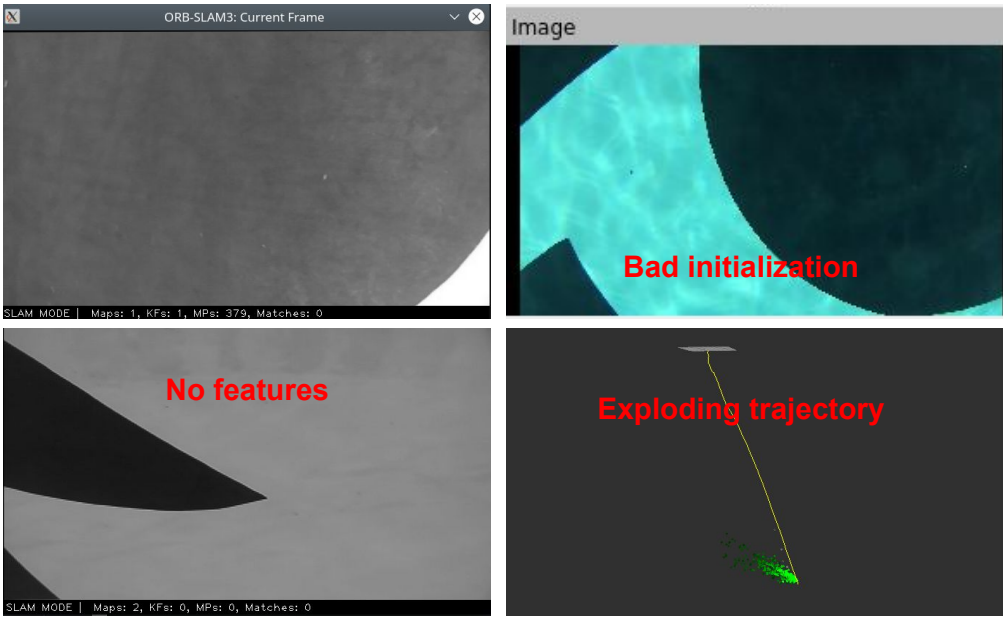



Figure 2.6: Failure cases of existing vision-based SLAM baselines in a low-texture underwater pool environment. We evaluate ORB-SLAM3 [10] and SVIn2 [144, 145] using image sequences collected in a controlled pool experiment. Here, SVIn2* denotes our reimplementation of SVIn2 on our robot platform with the sonar input disabled, so that it operates as a visual-inertial baseline. The pool environment contains large textureless surfaces, limited salient visual landmarks, and underwater illumination artifacts, which reduce the number and reliability of visual correspondences needed for tracking and initialization. (a) ORB-SLAM3 fails to maintain robust tracking because the underwater images contain insufficient repeatable features. (b) SVIn2* fails to initialize reliably and produces an unstable, diverging trajectory estimate. These examples illustrate why visual and visual-inertial SLAM pipelines do not directly transfer to low-texture underwater environments.

We report quantitative comparison of our method and other localization baselines in Table 2.2. From this table, we can see that the proposed method is able to maintain the lowest ATE in both position and rotation. In addition, we plot the trajectories of the proposed state estimation algorithm and other baselines in Figure 2.7, which shows the proposed method closely aligns to the ground truth trajectory with the least amount of drift.

In our localization system design, we propose to use IMU preintegration to generate an additional pseudo DVL velocity measurement to mitigate the time gap between the latest DVL measurement and current keyframe. We also design the DVL factor to be aware of the reported DVL velocity measurement uncertainty. We conduct an ablation study of these two design components. As shown in Table 2.3, using the IMU to predict a pseudo velocity

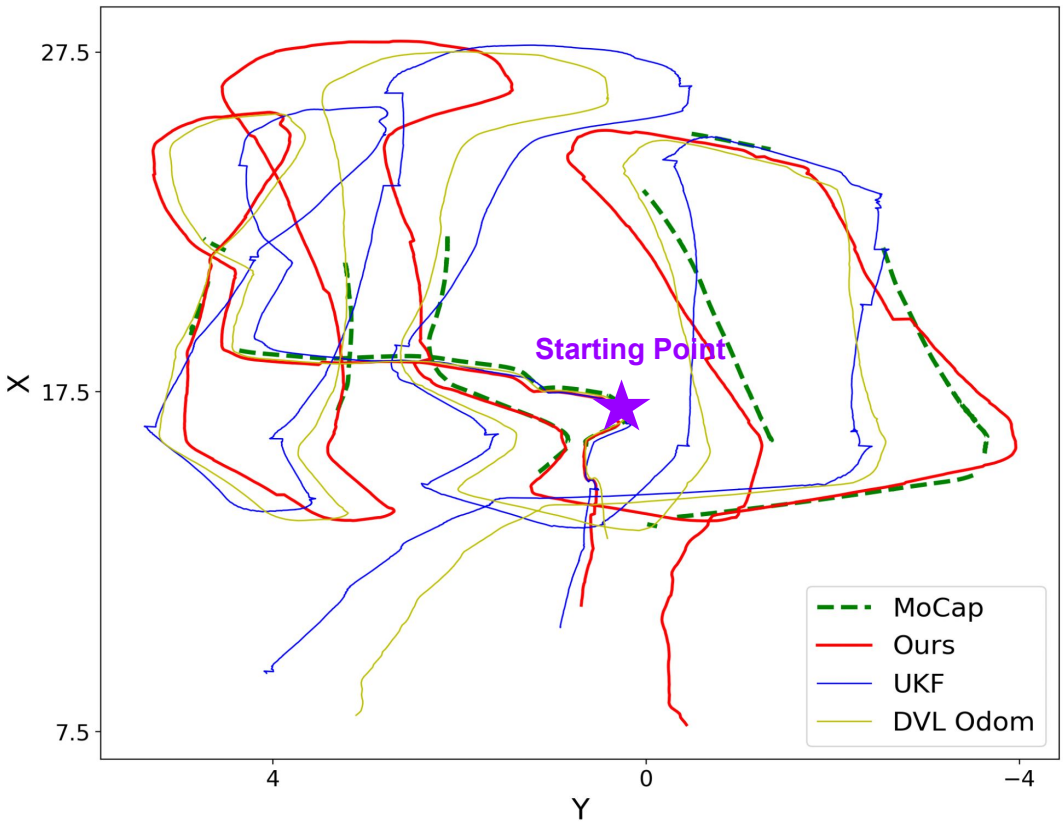



Figure 2.7: A qualitative comparison of estimated trajectories for log 1 in the x-y plane shown in meters. The starting point indicates where the robot enters the MoCap's valid tracking zone. Our method maintains the most accurate trajectory with the least amount of drift over time. Best viewed with color and zoomed-in.

Table 2.2: Comparison of the localization accuracy of the proposed method (TURTLMap) and baselines. Abbreviations: DVL (D), IMU (I), and Barometer (B). $\text{ATE}_{\text{pos}}$ is measured in meters, and $\text{ATE}_{\text{rot}}$ is measured in degrees. Lower values indicate better performance, and the best result for each metric is **bolded**.

| Method | Modality | | | Log 1 | | Log 2 | |
|---|---|---|---|---|---|---|---|
| | D | I | B | $\text{ATE}_{\text{pos}}$ | $\text{ATE}_{\text{rot}}$ | $\text{ATE}_{\text{pos}}$ | $\text{ATE}_{\text{rot}}$ |
| DVL Odom. | ✓ | ✓ | ✗ | 1.84 | 7.52 | 0.99 | 6.52 |
| UKF [158] | ✓ | ✓ | ✓ | 0.48 | 11.86 | 0.36 | 15.8 |
| TURTLMap | ✓ | ✓ | ✓ | **0.18** | **3.72** | **0.26** | **4.22** |

measurement for the latest keyframe is effective in improving the localization accuracy. When compared with using a constant uncertainty model for the DVL velocity measurement, we see improvement from the proposed method as it fuses the sensor measurement quality reported from the manufacturer driver to determine the factor weight in the pose graph.

Table 2.3: Ablation study of the proposed localization module in TURTLMap. $ATE_{pos}$ is measured in meters, and $ATE_{rot}$ is measured in degrees. Lower values indicate better performance, and the best result for each metric is **bolded**.

| Pseudo Vel. | Sensor Unc. | Log 1 | | Log 2 | |
|---|---|---|---|---|---|
| | | $ATE_{pos}$ | $ATE_{rot}$ | $ATE_{pos}$ | $ATE_{rot}$ |
| ✗ | ✓ | 0.32 | 5.44 | 0.40 | 4.72 |
| ✓ | ✗ | 0.23 | **3.63** | 0.28 | 4.42 |
| ✓ | ✓ | **0.18** | 3.72 | **0.26** | **4.22** |

Table 2.4: Quantitative mapping evaluation using different localization inputs. Mapping error is measured in meters. Lower values indicate better performance, and the best result for each log is **bolded**.

| Localization Input | Mapping Error (m) | |
|---|---|---|
| | Log 1 | Log 2 |
| DVL Odom | 1.733 | 0.806 |
| UKF [158] | 0.105 | 0.112 |
| TURTLMap (Ours) | **0.019** | **0.040** |

### 2.4.5 Mapping Results

We evaluate the dense maps generated by TURTLMap of the two logs in Table 2.1. We compare it with the reference structure of the tank, which is a simple flat surface. Table 2.4 reports the error between the generated map and the ground truth reference with different localization methods as input. We do not include ORB-SLAM3 [10] and SVin2 [144, 145] as they fail to generate a map. It is demonstrated that the proposed mapping system with our localization input is able to maintain an accurate map with low error compared to other localization baselines as our method produces more accurate state estimation. The comparison between the two logs also demonstrates that the environmental disturbance caused by waves introduces challenges to the proposed system, but the mapping error only increases to 0.04 m.

We also show a comparison of the generated map from different perspectives with images of the tank structure in Figure 2.8. The qualitative comparison highlights the capability of the proposed real-time mapping solution. In log 1, the large-scale bird-shaped logo is reconstructed in detail in the generated map and matches the picture well, demonstrating the effectiveness of the proposed solution. The map generated from log 2 also validates the system's capabilities under wave disturbance, with the rock platform being mapped precisely. We show a color map with distance to the ground truth reference of the bottom surface to highlight the mapping accuracy. We additionally show a qualitative example of the artificial

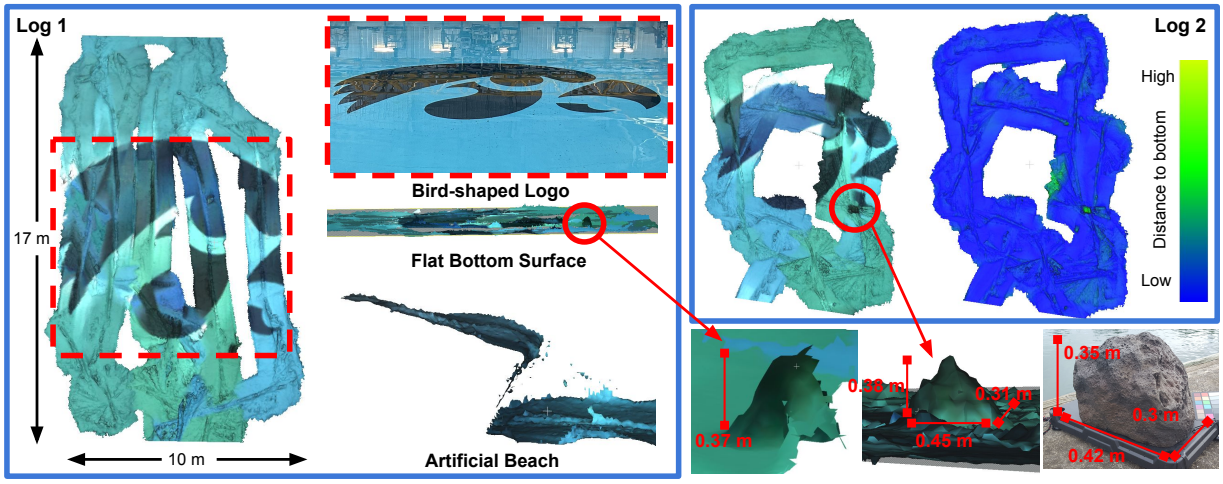



Figure 2.8: Qualitative results of the map generated from both logs. In log 1, we highlight the reconstructed bird-shaped logo and the corresponding image. The map matches the flat bottom surface well while including details of the artificial beach. In log 2, we plot a color map of distance to the bottom surface showing the capability of the proposed solution under wave disturbance. We also show a comparison of the rock dimensions captured from both logs.

beach in the generated map, demonstrating the capability of the system in maintaining an accurate map in scenarios of significant altitude change. This capability is very important for safe navigation for UUVs.

### 2.4.6 Runtime

We measure the runtime of the localization module and mapping module on the Jetson Orin Nano embedded computer. The time cost of pose graph optimization at each keyframe varies between 0.01s and 0.04s. Due to the applied fixed lag smoother, the time cost is consistent over the complete run. The time cost for each map update is consistently around 0.02s in the longest survey log. As these two modules are running on different threads, we can conclude that the proposed system is capable of running in real-time.

## 2.5 Conclusion

We have proposed TURTLMap, an effective real-time localization and dense mapping solution for a low-cost UUV. The robustness of TURTLMap is evaluated in challenging low-texture underwater scenes with and without waves. We develop a multimodal localization stack using a factor graph. We conduct real-time volumetric mapping based on stereo depth

information and estimated robot pose from the proposed localization module. We conduct extensive evaluation of the proposed system with real-world data including underwater Mo-Cap and ground truth mapping reference, which shows the effectiveness of the proposed system.

In this work, only the stereo point cloud is used for mapping. Extending this work to fuse the sparse acoustic measurements for mapping is an interesting future direction. Another future direction is to evaluate the proposed system in real field environments with a mix of low-texture and high-texture scenes. In addition, we plan to extend the proposed system to add visual odometry factors and visual loop closure in environments with sufficient visual features, enhancing performance in activities such as shipwreck surveys and coral reef monitoring [175, 144, 145, 151]. This will be a natural extension due to our factor graph formulation.

# CHAPTER 3

# Deep Adaptive LiDAR-Radar Fusion for 3D Object Detection

## 3.1 Introduction

Autonomous vehicles (AVs) are expected to accurately perceive the surrounding environment to enable effective and safe planning and control across a variety of scenarios and environmental conditions [114, 176, 187, 186, 196]. An important part of the perception task is to precisely localize the objects in the surrounding environment. A common representation for these objects is a set of 3D bounding boxes that have locations, sizes, and classes [40, 8]. Despite various combinations of sensor configurations on AVs, many object detection algorithms rely on LiDAR and cameras due to their dense returns [176, 138, 114, 212, 198, 165].

Still, LiDAR systems and cameras are sensitive to varying weather and lighting conditions, so AVs can suffer significantly from degraded perceptual capability in these scenarios. To tackle this problem, recent research has focused on leveraging radar systems, which have automotive-grade designs that ensure robust performance under various conditions [230, 199, 122, 89]. Additional benefits of radars include their low cost, long detection range, and Doppler effect information (i.e., velocity of captured targets). Therefore, it is of great significance to design a model that can effectively leverage radar for 3D object detection [176, 122, 199].

Existing detectors with radar can be categorized into single-modality methods [163] and fusion-based methods [199, 89]. Recent works [137, 91] have achieved impressive detection accuracy when fusing LiDAR and radar on the Oxford Radar RobotCar dataset [4], which has high-resolution radar data. However, this dataset uses a spinning radar, which lacks Doppler information and has increased cost [230]. Among the popular datasets for 3D object detection [24, 40, 138, 230], nuScenes [8] stands out because it is large-scale and has a complete sensor suite including radar. nuScenes represents radar data as object lists, which is a common representation that can also be interpreted as a very sparse point cloud with additional

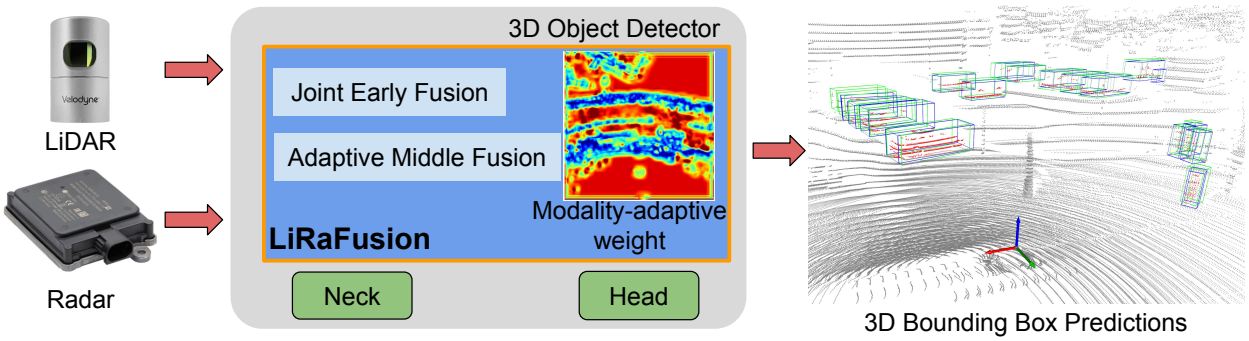



Figure 3.1: We propose LiRaFusion to efficiently leverage the complementary information of LiDAR and radar for 3D object detection.

feature attributes from radar onboard signal processing [230, 199]. The common challenges with this dataset are the sparsity and noise of the radar data. Consequently, single-modality radar detectors fail to achieve reliable performance. Some fusion-based detectors [18, 31] suffer from degraded performance when adding radar to LiDAR-Only or LiDAR-Camera fusion detectors, while some detectors [89, 199, 168] have to enforce hard constraints such as limiting detection to specific classes or limiting detection range to improve performance. Our work seeks to fill the gap in the current literature by improving the design of fusion architecture for LiDAR and radar by leveraging their shared point cloud representation.

In this chapter, we propose a novel LiDAR-Radar fusion detector, LiRaFusion (Figure 3.1). To summarize, our main contributions are as follows:

- We propose a novel joint feature extractor for effective LiDAR-Radar fusion.
- We introduce the adaptive gated network into LiDAR-Radar fusion for object detection with novel improvement considering the bird's-eye view (BEV) feature space.
- We conduct extensive evaluation on open source datasets and detectors that demonstrate improvement over existing LiDAR-Radar fusion methods. As most existing detectors follow the backbone-neck-head design [198, 206, 18], LiRaFusion can be directly integrated into existing methods by serving as the backbone to enable more modality configurations, which is validated by extending it to LiDAR-Camera-Radar fusion. Code is made available on the project website: https://github.com/Song-Jingyu/LiRaFusion.

## 3.2 Related Work

### 3.2.1 Radar Datasets for Autonomous Driving

Data is the key component for enabling development of learning-based object detectors. However, radar data was rarely available in early public autonomous driving datasets [40, 160, 24]. Recently, more datasets with radar data have become accessible to researchers. In these datasets, radar data usually has two representations. The most common representation is the point cloud, in which each point represents an object in the object list output by many off-the-shelf radars with on-board processing algorithms such as Constant False Alarm Rate (CFAR) [230, 199]. This representation is available on datasets such as nuScenes [8], aiMotive [115], and Zendar [119]. There are also datasets that directly use the raw data from radars [4, 154]. The raw data has denser information but the lack of CFAR processing leads to increased noise. The spinning radar configuration in [4] loses Doppler information (e.g., velocities of the captured targets), which is important to understand the scene. Another challenge brought by this representation is the difficulty of annotating the data [230]. In this work we leverage the nuScenes dataset [8] because of its full coverage on sensors and driving scenarios, accurate annotations, and popularity, which allows us to compare this work to many existing works. The proposed fusion architecture can be transferred to other datasets that have similar point cloud representation.

### 3.2.2 Multimodal 3D Object Detection

3D object detection is an important part of the perception system for AVs [114]. The main objective is to assign a class label and a 3D bounding box for each detected object in the scene [138]. Common sensors used for 3D object detection include LiDAR and cameras. Though several single-modality object detectors with LiDAR [206] or cameras [93] achieve impressive results on KITTI [40] or nuScenes [8] benchmarks, multimodal object detectors have recently shown promise in leveraging complementary information to improve robustness and accuracy [176, 206, 156, 192, 31, 128, 165]. LiDAR-Camera fusion is the most common configuration. However, these two sensors have shared drawbacks (e.g., sparse information at long range, lacking velocity estimation) that could be compensated by radars that are commonly deployed on AVs [230, 199]. Radar-only detectors usually fail to overcome the data sparsity to perform comparably to camera- or LiDAR-based detectors [177, 163].

One recent trend is to fuse radar with one or several other sensors. Existing fusion configurations include Camera-Radar (CR), LiDAR-Camera (LC), LiDAR-Radar (LR), and LiDAR-Camera-Radar (LCR) [121, 61, 199, 122, 5, 18, 89]. As the main focus of this work

is fusing LiDAR and radar in the shared point cloud representation, we mainly compare our methods with FUTR3D [18] and EZFusion [89] because they are the most recent state-of-the-art detectors supporting LR fusion on the nuScenes [8] dataset. In FUTR3D [18], though LR and LCR configurations are supported, they suffer from degraded performance when compared with LiDAR-Only (LO) or LC fusion. We argue the failure could come from the simple multilayer perceptron (MLP)-style feature extractor for radar and the lack of joint feature fusion before sampling features for query points. Our proposed method aims to address these limitations with the proposed adaptive fusion framework. EZFusion [89] is built on CenterPoint [206] by adding radar feature projection for the LiDAR points. In EZFusion [89], though its LR fusion shows improvement over its LO configuration (equivalent to CenterPoint [206]), its partial moving class setting, which uses only the 7 moving classes out of 10 classes in the nuScenes benchmark, has more limited application for practical deployment since the missed static classes are also vital for keeping AVs safe. Our proposed method achieves further improvement over EZFusion under the same partial classes setting and is demonstrated on the complete class setting, which has stronger potential in application since it can account for both static and moving object classes.

### 3.2.3 Gated Network for Sensor Fusion

Fusing information from different modalities requires sophisticated architecture design, and multiple prior works have addressed this challenge [35]. Among them, projection, addition, and concatenation are common practices [165, 31, 89, 122]. Though these methods demonstrate improvement in multimodal fusion, they are not learning-based and lack adaptivity. To account for this issue, researchers have turned to gated networks when different feature maps are fused. This process is also known as a mixture of experts because the backbones used for different modalities are considered different expert networks. This method was first introduced in [59], in which the expert network is defined as a domain-specific neural network to process a single sensing modality, and the gating network is a weighting neural network that selects useful features among the outputs of the expert networks. This idea has been leveraged by the perception community as more works have focused on using different modalities [176, 207, 224, 139]. For instance, in 3D object detection, 3D-CVF [207] proposes an LC fusion network using a gated network. Extensive evaluation is conducted on the KITTI [40] and nuScenes [8] datasets, which demonstrates the effectiveness of the gated network.

The success of fusing LiDAR and camera modalities through the gated network motivates us to adapt this method for the LR sensor configuration. The gated network is able to

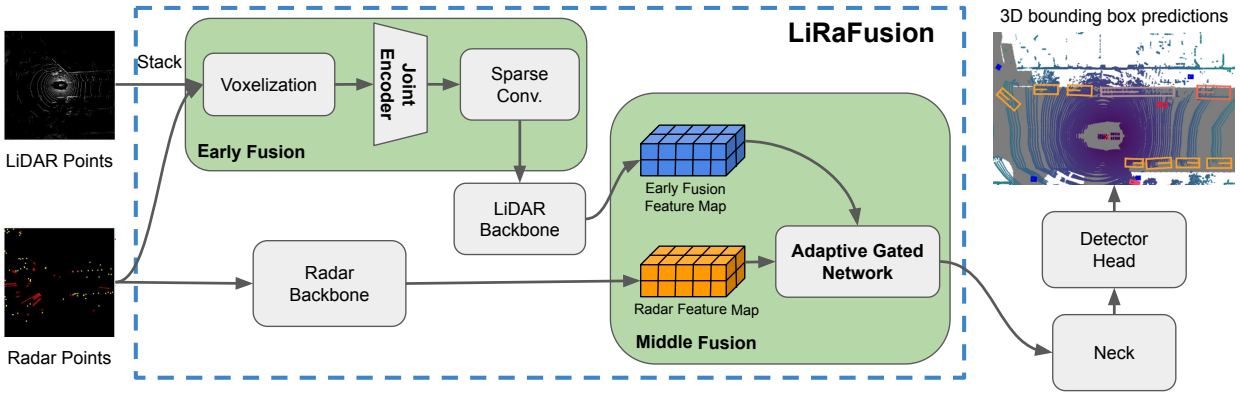



Figure 3.2: Overview of the architecture of LiRaFusion. Our main contributions, shown as bold text, mainly include a joint voxel feature encoder to extract per-voxel features from the stacked point cloud, and a gated network to learn weights for each input feature map to fuse them adaptively.

learn adaptive weights for different expert networks so the model can learn to be robust to noise from individual experts. We find this feature is of great significance in LR fusion since radar is noisier, which can degrade the performance of multimodal detectors if the information is not properly handled. In our proposed work, we extend the original gated network design [69, 207] by making it channel-wise so that each channel of the BEV map has an adaptive weight. To the best of our knowledge, LiRaFusion is the first to introduce the gated network into LR perception.

## 3.3 Method

The goal of our method, LiRaFusion, is to achieve more effective feature extraction and fusion for LiDAR and radar data in 3D object detection (Figure 3.2). The inputs to LiRaFusion are a LiDAR point cloud and a radar point cloud. One stream stacks these two point clouds as the input to the proposed early fusion module. The early fusion module processes the denser point cloud with the proposed joint feature encoder and a common sparse 3D convolution encoder. Its output is then fed into a common LiDAR backbone to obtain the feature map. In this work, we use VoxelNet following [18, 206]. The other stream uses the PointPillars [79] backbone to process the radar points, taking advantage of the pillars since the height measurements for radar points are noisy [199, 230]. The output is a radar feature map. The output feature maps from these two streams can be considered as two experts, which are further fused with the proposed gated network in the middle fusion module. The middle

fusion module learns the adaptive weights for these two feature maps and then concatenates the weighted feature maps together. The concatenated feature map is passed into the Feature Pyramid Network (FPN) [99] neck and the detector head to generate predictions. Our main contributions are the novel architectures for the early fusion and middle fusion modules. LiRaFusion is an enhanced backbone for LR feature extraction, so it can be extended to the LCR configuration as well. Technical details of each module are discussed in the following subsections.

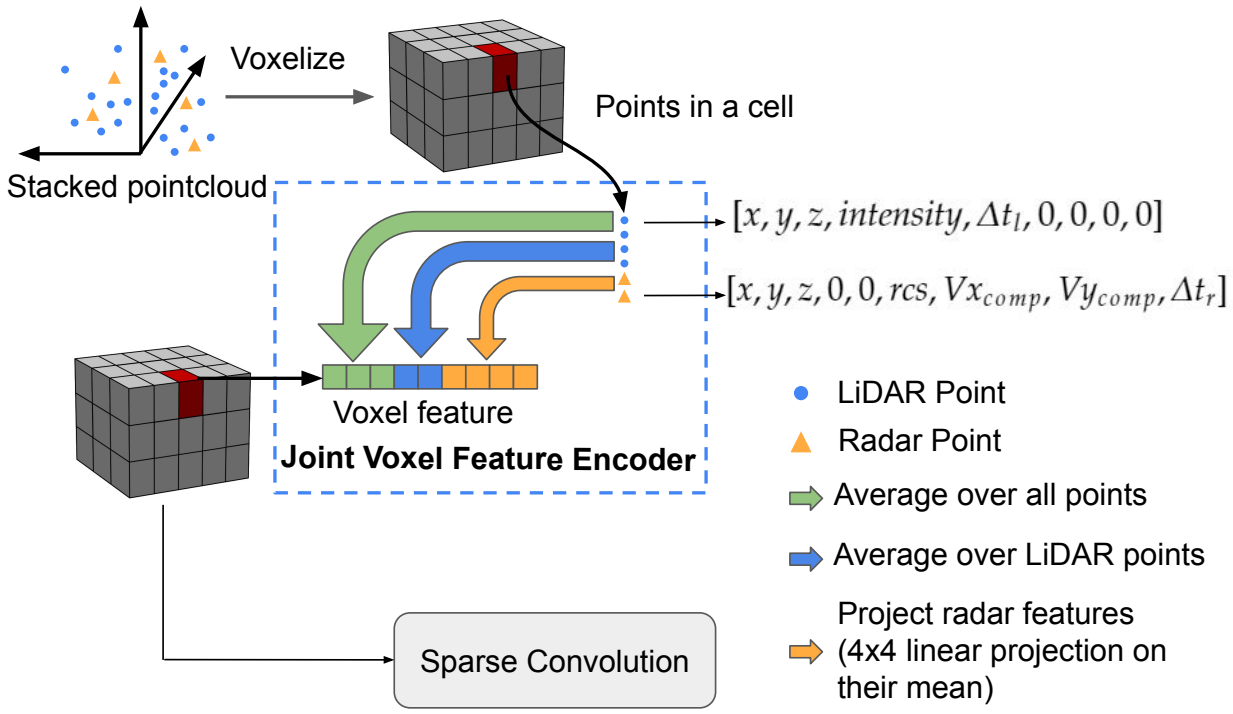



Figure 3.3: The network architecture of the early fusion module. LiDAR and radar points are zero-padded to a common feature space before voxelization: LiDAR points are represented as $[x, y, z, \text{intensity}, \Delta t_l, 0, 0, 0, 0]$, while radar points are represented as $[x, y, z, 0, 0, \text{rcs}, V_{x_{\text{comp}}}, V_{y_{\text{comp}}}, \Delta t_r]$. Here, $(x, y, z)$ denotes point coordinates, $\Delta t_l$ and $\Delta t_r$ denote the sweep time offsets, RCS denotes radar cross-section, and $V_{x_{\text{comp}}}$ and $V_{y_{\text{comp}}}$ denote compensated radar velocities. The joint voxel feature encoder aggregates the stacked LiDAR-radar points within each voxel cell and outputs voxel features for sparse convolution.

### 3.3.1 Early Fusion

We design an early fusion block to fuse the LiDAR and radar points at an early stage to extract features for each voxel cell, as shown in Figure 3.3. Unlike [199, 115, 168], which ignore features such as LiDAR intensity, radar cross-section (RCS), and velocity, we keep these features since LiDAR intensity and RCS are helpful for classifying objects, and velocity information is important for distinguishing static or dynamic objects and predicting velocity and rotation. For the LiDAR input, we keep the point intensity and the captured time offset ($\Delta t_l$) from the current frame as we accumulate multiple sweeps. For the radar points, we keep the RCS, compensated velocities ($V_{x_{\text{comp}}}, V_{y_{\text{comp}}}$), and the time offset ($\Delta t_r$). We use zero-padding to match the dimension of these points to merge them. After voxelization, we use the proposed joint voxel feature encoder to extract features for each voxel cell. We follow the simplified voxel feature encoder of VoxelNet [198] in MMDetection3D [24] to set the number of voxel feature dimensions to be the same as the input points. The first 3 feature dimensions represent the location of the centroid of this cell, which is computed by taking the mean locations of all the points in this voxel cell. The following 2 feature dimensions correspond to features from LiDAR, so we average over all LiDAR points. The last 4 feature dimensions correspond to the radar features. We pass the mean of the radar features to a $4 \times 4$ linear layer to enable the network to learn an appropriate way to handle the radar features. We only process non-empty voxel cells. For cells that do not have radar points, which is common due to sparsity, we leave the last 4 dimensions of these cells as zero. After obtaining the voxel features, we apply standard sparse convolution and further process its output with a standard LiDAR backbone, VoxelNet [198]. To simplify the terminology, we refer to the output of the early fusion stream as the LiDAR feature map in the following sections. Though radar data has already been fused when encoding the voxel feature, due to the sparsity of radar data, most information in the encoded voxel features is from LiDAR. We further fuse it with the radar feature map at a higher level with the proposed middle fusion module.

### 3.3.2 Middle Fusion

Specifically, the generated adaptive weights for the input feature maps have shape $B \times C \times H \times W$, where $B$ denotes the batch size, $C$ denotes the number of feature channels, and $H$ and $W$ denote the spatial height and width of the bird's-eye-view feature map. This differs from the previous channel-constant weighting scheme with shape $B \times 1 \times H \times W$, where a single weight is shared across all channels at each spatial location. This change enables the gated network to weight and extract LiDAR and radar features in a more flexible, channel-specific

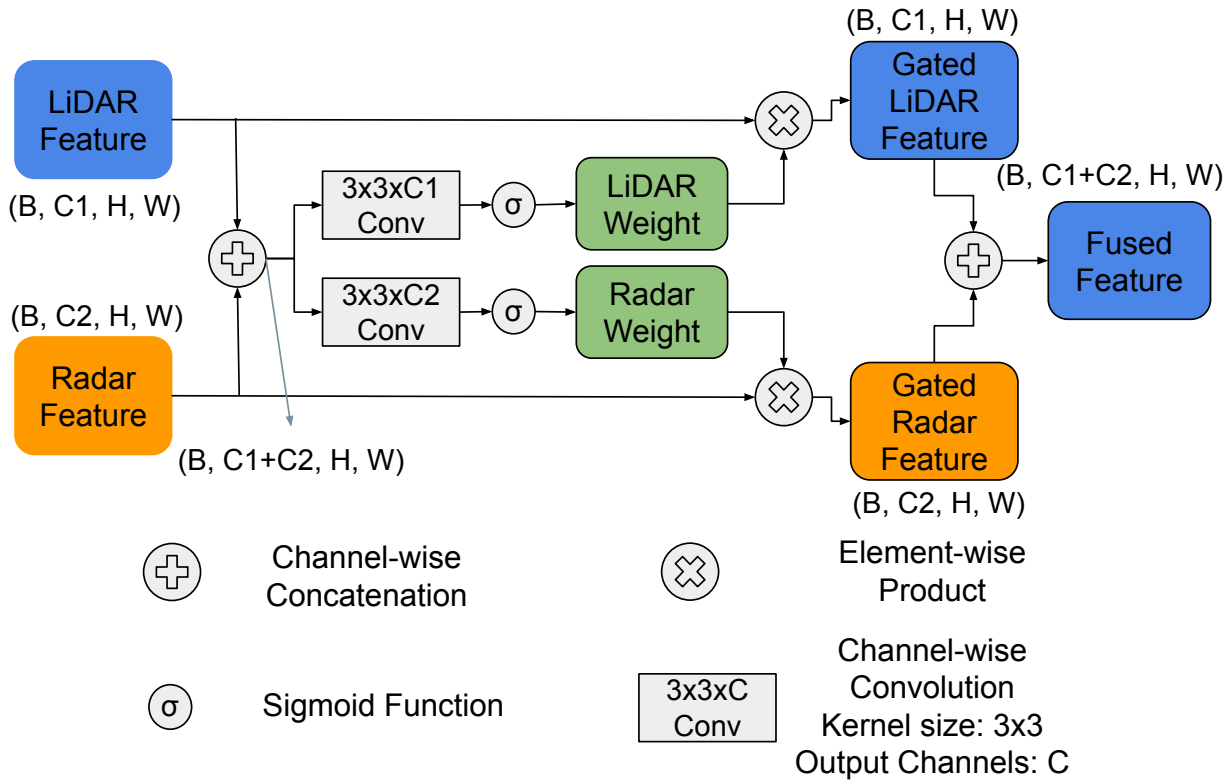



Figure 3.4: The network architecture of the middle fusion module. The LiDAR feature map $B \times C_1 \times H \times W$ and radar feature map $B \times C_2 \times H \times W$ are concatenated along the channel dimension and passed through modality-specific $3 \times 3$ convolution layers followed by sigmoid functions to generate channel-wise adaptive weights. The original LiDAR and radar feature maps are then element-wise multiplied by their corresponding weights to produce gated features, which are concatenated into the fused feature map $B \times (C_1 + C_2) \times H \times W$. Here, $B$ denotes the batch size, $C_1$ and $C_2$ denote the LiDAR and radar channel dimensions, and $H$ and $W$ denote the spatial height and width of the bird's-eye-view feature maps.

manner. Since the feature maps are represented in bird's-eye view after being flattened along the $z$ axis, the proposed channel-specific weights allow the network to better exploit spatially varying and modality-specific cues from the input feature maps.

The design of the adaptive gated network is shown in Figure 3.4. The input LiDAR feature map ($B \times C_1 \times H \times W$) and input radar feature map ($B \times C_2 \times H \times W$) are first concatenated in the channel dimension. Then the concatenated feature map is passed to a convolution block that is followed by a sigmoid function. Notably, the output feature dimension of the convolution blocks for LiDAR and radar modalities is set to match the dimension of the input feature maps ($C_1$ for LiDAR and $C_2$ for radar). The learned adaptive

weights are applied to the original input feature maps through an element-wise product operation. The obtained gated LiDAR and radar features are further concatenated together along the feature dimension as a fused feature map. The resulting fused feature map has the shape $B \times (C_1 + C_2) \times H \times W$ and serves as the output of the middle fusion block.

## 3.4 Experiments

### 3.4.1 Experimental Setup

As mentioned in section 3.2.1, we evaluate our method on the nuScenes dataset [8]. We follow the official split that has 700 scenes for training and 150 scenes for validation. It only has annotations for the keyframes (samples), but also provides non-keyframes (sweeps) without annotations. We follow common practices in [24, 18, 89] to load multiple sweeps into a current sample frame to increase the data density, while appending a time difference channel for data from each sweep as additional temporal information.

We use the mean Average Precision (mAP) and nuScenes Detection Score (NDS) [8] as the main evaluation metrics. We compare LiRaFusion with the existing LR and LCR detectors. In addition, we group predictions to break down the improvement of fusing radar. We also report runtime and additional true positive (TP) metrics from nuScenes [8]. Ablation studies are conducted to validate our model design.

As mentioned previously, many existing detectors [206, 198, 18, 24] follow the backbone-neck-head design. Therefore, LiRaFusion can be integrated by replacing their backbones with LiRaFusion and keeping the same neck and head. As the baselines we choose – FUTR3D [18] and EZFusion [89] – are initially proposed to work with different heads, we group our experiments based on the detector heads. Inspired by [11, 178], FUTR3D [18] uses a transformer-based head, which is referred to as TransHead. EZFusion [89] uses the same head as CenterPoint [206], which is referred to as CenterHead. We choose the LO and LR configurations for FUTR3D as the baselines, denoted as FUTR3D-LO and FUTR3D-LR. We implement the LR fusion strategy in EZFusion with both heads, which is named as EZFusion-LR$^*$, where $^*$ represents our re-implementation. The other group of experiments with CenterHead [206] focuses on 7 moving classes out of 10 complete classes in the nuScenes official benchmark for consistency with results reported in EZFusion [89]. We re-train the original CenterPoint [206] with 7 classes and name it as CenterPoint-7. Since FUTR3D-LO and CenterPoint are the state-of-the-art LO detectors, we include them in our comparison to demonstrate the improvement obtained with fusing radar data. We implemented LiRaFusion based on the MMDetection3D framework [24]. More implementation and training details can be found

on the project website.

### 3.4.2 Quantitative Results

Table 3.1: Results with TransHead [18] evaluated on nuScenes *val* set. EZFusion-LR* represents our re-implementation of [89]. All values are percentages. No model ensemble or test-time augmentation is used.

| Method | Sensor | NDS ↑ | AP (Average Precision) ↑ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | mean (mAP) | Car | Ped | Bicycle | Bus | Barrier | TC | Truck | Trailer | Moto | CV |
| FUTR3D-LO | LO | 65.74 | 59.39 | 84.3 | 81.4 | 49.0 | 65.4 | **62.4** | 64.2 | 53.5 | **41.8** | 66.4 | 25.5 |
| FUTR3D-LR | LR | 65.37 | 58.08 | 83.8 | 81.2 | **49.8** | 65.4 | 60.4 | 60.6 | 51.0 | 41.0 | 65.1 | 22.6 |
| EZFusion-LR* | LR | 65.77 | 59.24 | 84.6 | 81.7 | 47.3 | 69.1 | 62.0 | 65.7 | 52.2 | 39.7 | 66.9 | 23.3 |
| LiRaFusion (ours) | LR | **66.69** | **60.11** | **85.6** | **82.2** | 46.9 | **69.6** | 61.2 | **66.0** | **54.0** | 40.7 | **68.1** | **26.7** |

Abbreviations: pedestrian (Ped), traffic cone (TC), motorcycle (Moto), and construction vehicle (CV).

We perform a comparison of our method with several state-of-the-art LR fusion networks with CenterHead [206] and TransHead [18] separately on the nuScenes dataset [8]. Table 3.1 shows the results of all models trained with TransHead [18]. We can see that FUTR3D-LR performs worse than FUTR3D-LO, which proves that ineffective design of LR fusion strategy could actually harm performance. The re-implemented EZFusion with TransHead (EZFusion-LR*) fails to achieve further improvement over FUTR3D-LO. Our model, LiRaFusion, achieves the best results on the nuScenes validation set in terms of NDS and mAP. It also shows impressive improvement on certain classes such as car and pedestrian, the top two most frequent classes, which are critical for AVs to detect in the scene in order to operate effectively and safely.

Table 3.2 shows the results for models trained with CenterHead using the 7 moving classes for consistency with EZFusion [89]. Similarly to EZFusion, we re-train CenterPoint with the 7-class setting as a baseline LO detector and name it as CenterPoint-7. Though EZFusion-LR* achieves considerable improvement over CenterPoint-7, there is a small gap between EZFusion-LR* and the reported results in EZFusion [89] so we include both of them in the table. LiRaFusion is the top performer in terms of NDS and mAP. Similar to its performance with TransHead [18], LiRaFusion achieves impressive improvement in terms of AP over almost all classes.

In addition to the comparison with the most recent state-of-the-art LR detectors [18, 89] shown above, we compare LiRaFusion (with TransHead) with other LR or LCR detectors for a complete overview. Table 3.3 demonstrates that LiRaFusion has consistent improvement over existing LR and LCR detectors. It is worth mentioning that many detectors in Table 3.3 have to enforce a partial class setting, or use the extra camera modality, while LiRaFusion

Table 3.2: Results with CenterHead [206] evaluated on nuScenes *val* set. We train these networks with 7 moving classes. EZFusion-LR stands for the results in its original paper [89]. * represents our re-implementation. In this experiment group, test-time augmentation was used to keep consistency with EZFusion [89]. No model ensemble is used. All values are percentages.

| Method | Sensor | NDS ↑ | AP (Average Precision) ↑ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | mean (mAP) | Car | Ped | Bicycle | Bus | Truck | Trailer | Moto |
| CenterPoint-7 | LO | 69.41 | 61.38 | 86.0 | 79.1 | 40.3 | 68.7 | **57.7** | 37.1 | 60.7 |
| EZFusion-LR | LR | N/A | 63.21 | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| EZFusion-LR* | LR | 69.85 | 61.85 | 86.0 | 79.1 | 43.6 | 70.3 | 57.4 | 35.6 | 60.8 |
| LiRaFusion (ours) | LR | **72.16** | **65.18** | **86.8** | **79.4** | **54.8** | **70.4** | 57.5 | **38.7** | **68.7** |

Table 3.3: Comparison with LR and LCR detectors on nuScenes *val* (**top**) and *test* (**bottom**) set. The reported LiRaFusion results use TransHead [18]. No test-time augmentation or model ensemble is used. Results for other models are the reported results in the papers so some entries are missing. All values are percentages; higher is better.

| Method | Sensor | NDS | mAP | AP (car) | AP (Moto) |
|---|---|---|---|---|---|
| RadarNet [199] | LR | N/A | N/A | 84.3 | 53.7 |
| RVF-Net [122] | LR | N/A | N/A | 54.18 | N/A |
| RVF-Net [122] | LCR | N/A | N/A | 54.86 | N/A |
| DeepFusion [31] | LCR | N/A | N/A | 83.5 | N/A |
| LiRaFusion (ours) | LR | 66.7 | 60.1 | 85.6 | 68.1 |
| Sparse-PointNet [168] | LCR | N/A | 48.9 | 71 | 36 |
| Frustum PointNet [167] | LCR | N/A | 36.6 | 48 | 41 |
| LiRaFusion (ours) | LR | 66.2 | 59.5 | 84.7 | 63.3 |

Table 3.4: Runtime and True Positive (TP) metrics for experiments evaluated with TransHead [18]. Runtime is measured using frames per second (FPS). Higher is better for FPS; lower is better for TP error metrics.

| Method | Runtime | mATE | mASE | mAOE | mAVE | mAAE |
|---|---|---|---|---|---|---|
| FUTR3D-LO | 6.0 FPS | **0.342** | **0.265** | 0.321 | 0.276 | 0.193 |
| FUTR3D-LR | 5.6 FPS | 0.347 | 0.267 | 0.308 | 0.256 | 0.189 |
| EZFusion-LR* | 5.9 FPS | **0.342** | 0.266 | 0.315 | 0.267 | 0.196 |
| LiRaFusion (ours) | 5.5 FPS | 0.346 | 0.267 | **0.298** | **0.240** | **0.186** |

Abbreviations: mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), mean Average Attribute Error (mAAE).

trains on the complete class setting with the LR sensor configuration and outperforms all the LR and LCR detectors.

We additionally report the model runtime and the True Positive (TP) metrics defined in

Table 3.5: Performance by object distance of experiments with TransHead [18]. We report the mAP separately in 3 ranges: 0 $m$-20 $m$, 20 $m$-30 $m$, and 30 $m$-50 $m$. With increasing object distance, LiRaFusion demonstrates higher gain over FUTR3D-LO as radar complements LiDAR at far locations where LiDAR suffers from data sparsity. All values are percentages; higher is better.

| Method | mAP | | |
|---|---|---|---|
| | $[0\,m, 20\,m]$ | $[20\,m, 30\,m]$ | $[30\,m, 50\,m]$ |
| FUTR3D-LO | 73.86 | 55.2 | 29.93 |
| LiRaFusion (ours) | 74.14 ↑ 0.28 | 55.88 ↑ 0.68 | 31.73 ↑ 1.8 |

Table 3.6: Performance by weather conditions of experiments with TransHead [18]. We report results of two weather conditions: Sunny and Rainy. The grouping strategy is based on the official scene description entries in nuScenes [8]. There are in total 968 rainy frames out of 6019 frames. All values are percentages; higher is better.

| Method | NDS | | mAP | |
|---|---|---|---|---|
| | Sunny | Rainy | Sunny | Rainy |
| FUTR3D-LO | 65.55 | 65.39 | 59.28 | 57.32 |
| LiRaFusion (ours) | 66.42 ↑ 0.87 | 67.15 ↑ 1.76 | 59.93 ↑ 0.65 | 59.35 ↑ 2.03 |

[8] evaluated with the complete class setting in Table 3.4. The table shows LiRaFusion is comparable with other baselines in runtime, which is measured on the same desktop with an RTX A6000 GPU. We can also see LiRaFusion is comparable with the best method in terms of translation and scale error, while achieving the lowest error in estimating orientation, velocity, and attribute. We argue the significant reduction in orientation and velocity error comes from the effective fusion of the Doppler information from radar.

We also group predictions based on object distances from the ego-vehicle and weather conditions to further break down the performance boost, as this directly represents the improvement from radars. Table 3.5 shows the performance breakdown based on the distance. We find the performance gain increases with increasing distance. This finding matches the expectation that radars complement LiDAR by providing additional information at distant locations where the LiDAR returns become sparser. Table 3.6 shows that LiRaFusion achieves notable improvement over the LO baselines in rainy scenes, where LiDAR is generally believed to have reduced detection capability [5, 199]. This finding validates the importance of fusing radars and leveraging their robustness across different weather conditions.

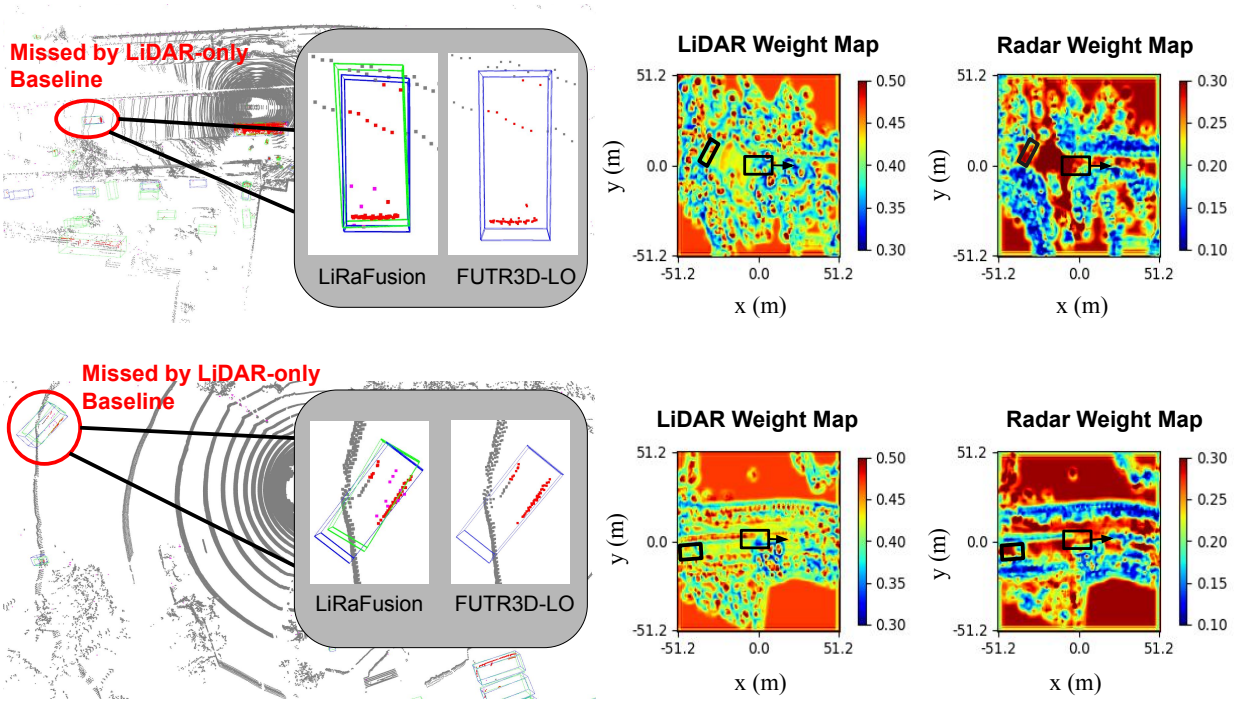



Figure 3.5: Example bounding box predictions and corresponding weight maps. We present two frames in which LiRaFusion correctly detects a car (highlighted with a red circle) that is missed by the baseline LO detector. We also show a zoomed-in view in which we label radar points in magenta, and LiDAR points in gray or red (if they reside in a bounding box). We show ground truth bounding boxes in blue and predictions in green. In the visualization of weight maps, the black bounding box with arrow denotes the ego-vehicle. Boxes without an arrow denote the highlighted missed car object. Best viewed in color and zoomed-in.

### 3.4.3 Qualitative Results

Figure 3.5 shows qualitative comparison of LiRaFusion and FUTR3D-LO by presenting the predictions along with the ground truth bounding boxes. These results show that the radar sensor contributes several measurement points (shown in magenta) for a car that was previously missed by FUTR3D-LO, which only uses LiDAR data. The radar and LiDAR points are used effectively by LiRaFusion and the prediction by LiRaFusion aligns well with the ground truth.

We also show the corresponding adaptive weights on LiDAR and radar feature maps in Figure 3.5. We see that the learned weight on the LiDAR feature map is generally higher than that in the radar feature map, which meets the expectation that LiDAR is the preferred sensor for object detection in terms of the density and geometric accuracy. We also notice that in the radar weight map, locations with larger distance to the ego-vehicle have relatively larger weight. This finding matches our intuition that incorporating radar data provides more information for long-range detection capabilities [199, 157, 230]. Basically,

the smaller radar weight at the near locations means that the detector learns to be more dependent on LiDAR at places where it has dense returns, while trusting radar returns at farther locations where LiDAR data has reduced density. Note that the radar weight within the black bounding box in Figure 3.5 is relatively large, which means that the proposed gated network is capable of learning to use the expert feature maps adaptively to let the LiDAR and the radar complement each other in a more effective way.

Table 3.7: Ablation study of fusion module. NDS, mAP, and AP values are percentages.

| Method | NDS ↑ | mAP ↑ | mAVE ↓ | AP (car) ↑ |
|---|---|---|---|---|
| FUTR3D-LO | 65.74 | 59.39 | 0.276 | 84.3 |
| LiRaFusion-early | 65.95 | 59.11 | 0.263 | 85.4 |
| LiRaFusion-middle | 66.59 | 59.45 | 0.241 | 85.3 |
| LiRaFusion | **66.69** | **60.11** | **0.240** | **85.6** |

### 3.4.4 Ablation Studies

We study the contribution of each fusion module to the overall performance. We denote LiRaFusion-early as the model with the early fusion module only and LiRaFusion-middle as the model with the middle fusion module only. We report the results of these models in addition to the baseline FUTR3D-LO. As shown in Table 3.7, both LiRaFusion-early and LiRaFusion-middle achieve improvement over the baseline in most metrics. When the early fusion and middle fusion modules are combined together, the performance of the combined model is further enhanced.

When designing the adaptive gated network used in the middle fusion module, we improve over the existing network design in [207, 69] that has a constant weight (referred to as channel-constant) for all features at one location in the bird's-eye view (BEV) feature map. Since the $z$ dimension and original feature dimension are merged together to form the BEV feature map, a specific weight for each feature dimension could help to utilize the spatial knowledge. Inspired by this, we propose a channel-specific weight map in the gated network. As shown in Table 3.8, the proposed channel-specific gated network outperforms the original network design, which validates the effectiveness of our improvement over the original gated network in [207, 69].

### 3.4.5 LiDAR-Camera-Radar Fusion

We explore the potential of LiRaFusion to fuse LiDAR, camera, and radar for 3D object detection as they are the common sensing modalities on modern AVs. Since most object de-

Table 3.8: Ablation study of gated network design. All values are percentages.

| Method | NDS ↑ | mAP ↑ |
|---|---|---|
| LiRaFusion-middle (Channel-constant) | 66.01 | 58.84 |
| LiRaFusion-middle (Channel-specific) | **66.59** | **59.45** |

tectors follow the backbone-neck-head paradigm, LiRaFusion can be applied to many LiDAR-Camera (LC) detectors by replacing the LiDAR backbone to enable LCR fusion. FUTR3D [18] supports LC configuration (FUTR3D-LC) and is one of the state-of-the-art LC detectors. By replacing its LiDAR backbone with LiRaFusion, we implemented the LCR model referred as LiRaFusion-LCR. We directly compare LiRaFusion with FUTR3D-LC to evaluate the scalability of LiRaFusion. Table 3.9 shows that LiRaFusion-LCR achieves further improvement over FUTR3D-LC. The results also demonstrate that radars can complement the LC configuration, which reinforces the importance of fusing radar data in modern object detectors. As the main focus of this project is on LR fusion, we leave more experiments on LCR fusion for future work, and hope our work can inspire more research on fusing radars with other sensors to improve perceptual capabilities of AVs.

Table 3.9: Results with LCR fusion and TransHead [18]. NDS and mAP values are percentages.

| Method | NDS ↑ | mAP ↑ | mATE↓ | mASE↓ | mAOE↓ | mAVE↓ | mAAE↓ |
|---|---|---|---|---|---|---|---|
| FUTR3D-LC | 68.0 | 64.2 | 0.350 | **0.259** | 0.304 | 0.305 | 0.193 |
| LiRaFusion-LR | 66.69 | 60.11 | 0.346 | 0.267 | **0.298** | **0.240** | 0.186 |
| LiRaFusion-LCR | **68.65** | **64.76** | **0.345** | **0.259** | 0.309 | 0.276 | **0.181** |

Abbreviations: mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), mean Average Attribute Error (mAAE).

## 3.5 Conclusion

We have proposed a novel LiDAR-Radar fusion network, LiRaFusion, to facilitate cross-modality feature extraction for 3D object detection. We design a joint voxel feature encoder to extract voxel feature encoding in an early stage. We propose an adaptive gated network to further fuse the feature maps from LiDAR and radar by learning modality-adaptive weight maps. Experimental results show that LiRaFusion achieves consistent improvement over existing LiDAR-Radar detectors on the nuScenes benchmark. Future work includes applying LiRaFusion to existing LiDAR-Camera detectors to further improve over existing LCR detectors, and also extending LiRaFusion to other scene understanding tasks.

# CHAPTER 4

# Enhanced Camera-Radar Fusion with Cross-modality Knowledge Distillation

## 4.1 Introduction

Perception is an important module for achieving safe and effective autonomous driving [12, 52, 186]. 3D object detection is an essential task in perception as it is of great significance for subsequent tasks [169, 159, 187, 131]. Among various perceptual sensors used by autonomous driving researchers, LiDAR, camera, and radar are the most common choices to enable autonomy on the road [169]. Sensor fusion techniques are usually used to improve the detector's performance and robustness. LiDAR-Camera (LC) fusion has been widely demonstrated as the top-performing sensor configuration for 3D object detection [8, 24, 107, 3, 204, 51]. However, the high cost of LiDAR constrains the wide application of this configuration. Though Camera-Only (CO) detectors have demonstrated impressive performance in recent bird's-eye view (BEV) frameworks [93, 90, 54, 53], the camera's vulnerability to lighting conditions and lack of accurate depth measurements motivates researchers to turn to other sensors such as radar. Radar is robust to varying weather and lighting conditions and features automotive-grade design and low cost. Radars are already highly accessible on most cars equipped with driver assistance features. Compared with LiDAR, radar measurements are sparse and noisy, which makes designing a Camera-Radar (CR) detector challenging. Recent CR detectors have leveraged the advancement brought by BEV-based CO detectors [93, 54, 53, 90] to achieve further improvement in accuracy and robustness to weather and lighting changes [81, 230].

Despite the advancement in architecture design, there is still a distinct performance gap when comparing LiDAR-Only (LO) and LC detectors against CO and CR detectors. Recent research has focused on applying the knowledge distillation (KD) technique to alleviate this gap [49, 227, 76, 180, 50]. Generally, KD features a teacher-student framework that aims to propagate the informative knowledge from a well-performing teacher model to facilitate the

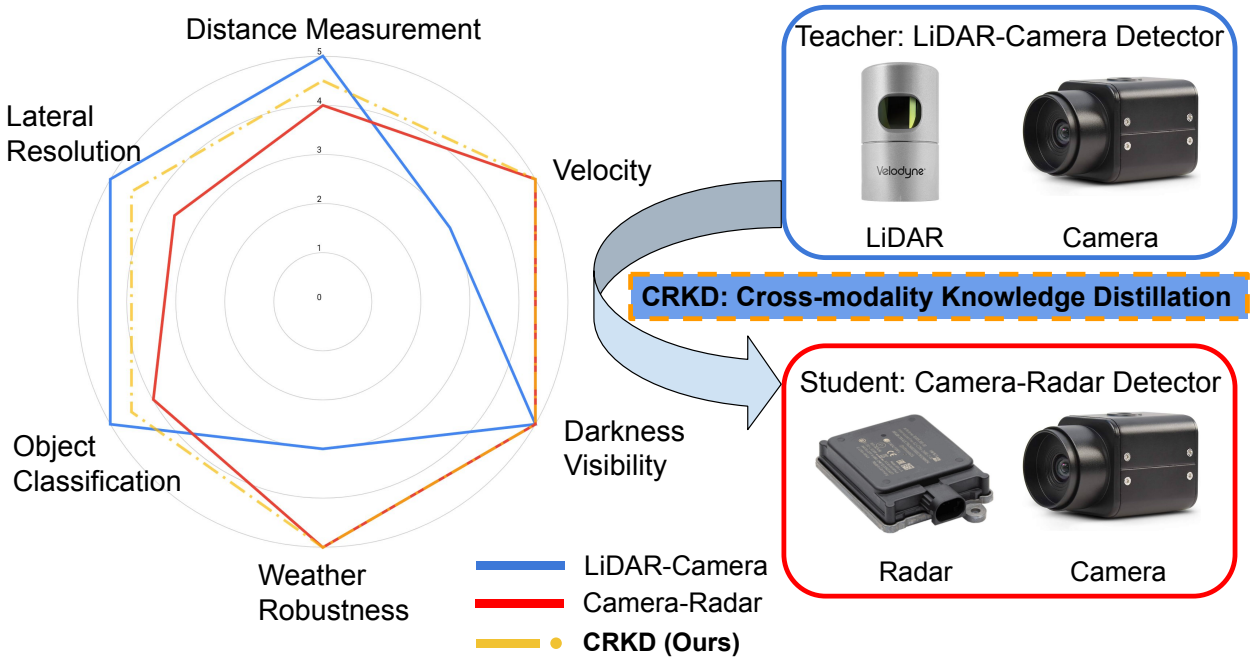



Figure 4.1: We propose CRKD to conduct a novel cross-modality knowledge distillation path from a LiDAR-Camera teacher to a Camera-Radar student. We present a radar chart to illustrate the complementary nature of these sensing configurations and the improvement that CRKD can enable.

learning process of the student model. This usually leads to improved performance compared to simply training the student model on the same task. The KD technique has been employed in either intra-modal [215, 182, 211, 84] or cross-modal [19, 21, 50, 55, 227, 76] configurations for 3D object detection. Though many cross-modal methods use a single-modality detector as the teacher model to leverage the privileged LiDAR data that is widely available in open-source datasets, they mainly focus on distilling knowledge to a LiDAR-based or camera-based student detector. We argue for the importance of designing a distillation path from an LC teacher detector to a CR student detector, which could benefit from the existing superior design of LC detectors and the shared point cloud representation between measurements of LiDAR and radar [199].

Inspired by the above observations, we propose **CRKD**: an enhanced **C**amera-**R**adar 3D object detector with cross-modality **K**nowledge **D**istillation (Figure 4.1) that distills knowledge from an LC teacher detector to a CR student detector. To our best knowledge, CRKD is the first KD framework that supports a fusion-to-fusion distillation path. As the LiDAR sensor is used only during training, we emphasize the value of CRKD as it could

facilitate the practical application of perceptual autonomy with a low-cost and robust CR sensor configuration.

To summarize, our main contributions are as follows:

- We propose a novel cross-modality KD framework to enable LC-to-CR distillation in the BEV feature space. With the transferred knowledge from an LC teacher detector, the CR student detector can outperform existing baselines without additional cost during inference.
- We design four KD modules to address the notable discrepancies between different sensors to realize effective cross-modality KD. As we operate KD in the BEV space, the proposed loss designs can be applied to other KD configurations. Our improvement also includes adding a gated network to the baseline model for adaptive fusion.
- We conduct extensive evaluation on nuScenes [8] to demonstrate the effectiveness of CRKD. CRKD can improve the mean Average Precision (mAP) and nuScenes Detection Score (NDS) of student detectors by 3.5% and 3.2%, respectively. As our method focuses on a novel KD path with a large modality gap, we provide thorough study and analysis to support our design choices.

## 4.2 Related Work

### 4.2.1 Multimodal 3D Object Detection

The multimodal 3D object detectors generally outperform single-modality detectors in accuracy and robustness as the perceptual sensors (e.g., LiDAR, camera, and radar) can complement each other [169]. Among the common sensor combinations, LC is the best-performing modality configuration on most existing datasets [8, 40, 160, 24]. In general, LiDAR and camera are fused in different ways. One trend is to augment LiDAR points with features from cameras [165, 166, 156, 56, 193], which is usually referred to as early fusion. Other solutions apply deep feature fusion in a shared representation space [207, 87, 88]. One trending choice is to leverage the BEV space to deliver impressive improvement [107, 96]. There are also methods that fuse information at a later stage. In [3, 18, 197], features are independently extracted and aggregated via proposals or queries in the detection head, while some methods [128, 129] combine the output candidates from single-modality detectors.

Nevertheless, the LC configuration is less accessible to consumer cars due to the high cost of LiDAR. Thus, the potential for CR detectors stands out due to the robustness brought by radars and the potential of large-scale deployment. One of the key challenges

facing CR detectors is how to handle the discrepancy in sensor views and data returns. CenterFusion [121] applies feature-level fusion by associating radar points with image features via a frustum-based method. Following this, more feature-level fusion methods are proposed in [31, 72, 228, 190]. Recently, as many camera-based methods [93, 90] have started to leverage the unified BEV space by transforming camera features from perspective view (PV), many CR detectors also explore fusion of camera and radar features in the BEV space [74, 70]. Though LiDAR data is not available in real inference and deployment, its wide availability in open-source datasets [8] has motivated researchers to leverage LiDAR data to guide the feature transformation process [81]. Motivated by the aforementioned works, CRKD also unifies features in the BEV space and leverages LiDAR data in a KD-based framework. To the best of our knowledge, CRKD is the first framework that improves CR detectors with cross-modality KD from a top-performing LC teacher detector.

### 4.2.2 Cross-modality Knowledge Distillation

The idea of KD is initially proposed in [49] for model compression in an image classification task. It is then extended to the field of object detection for model compression and performance improvement [92]. Specifically, in the field of 3D object detection, a group of KD methods requires that the teacher and student models use the same modality such as LO-to-LO (L2L) [215, 182, 179, 202, 65] and CO-to-CO (C2C) [211, 216, 84]. In contrast, cross-modality KD focuses on KD with different modality configurations. Typical paths include LiDAR-to-Camera (L2C) [42, 106, 21, 19, 50, 87, 55] and Camera-to-LiDAR (C2L) [227, 149]. Recently, new cross-modality KD paths including a fusion-based modality have been explored. In UniDistill [227], a universal framework that supports multiple KD paths is proposed. By unifying the features from different modalities to the shared BEV space, it supports L2C, C2L, LC-to-LO (LC2L) and LC-to-Camera (LC2C). DistillBEV [180] also supports L2C and LC2C by leveraging the shared BEV space. X3KD [76] proposes a cross-modal and cross-task KD framework for L2C KD. It is evaluated on an LO-to-CR (L2CR) KD path as a supplementary task. However, it lacks specific consideration for the large domain discrepancies with radars and further experiments and analysis with the L2CR KD path. Among the existing prior works, we observe the lack of KD methods that can support a fusion-to-fusion distillation path and handle domain differences with radars. We argue the importance of conducting distillation from an LC teacher to a CR student to leverage the shared BEV feature space of camera-based detectors and the shared point cloud representation of LiDAR and radar measurements. To the best of our knowledge, we are the first to investigate a KD framework with a fusion-to-fusion path. We demonstrate that our

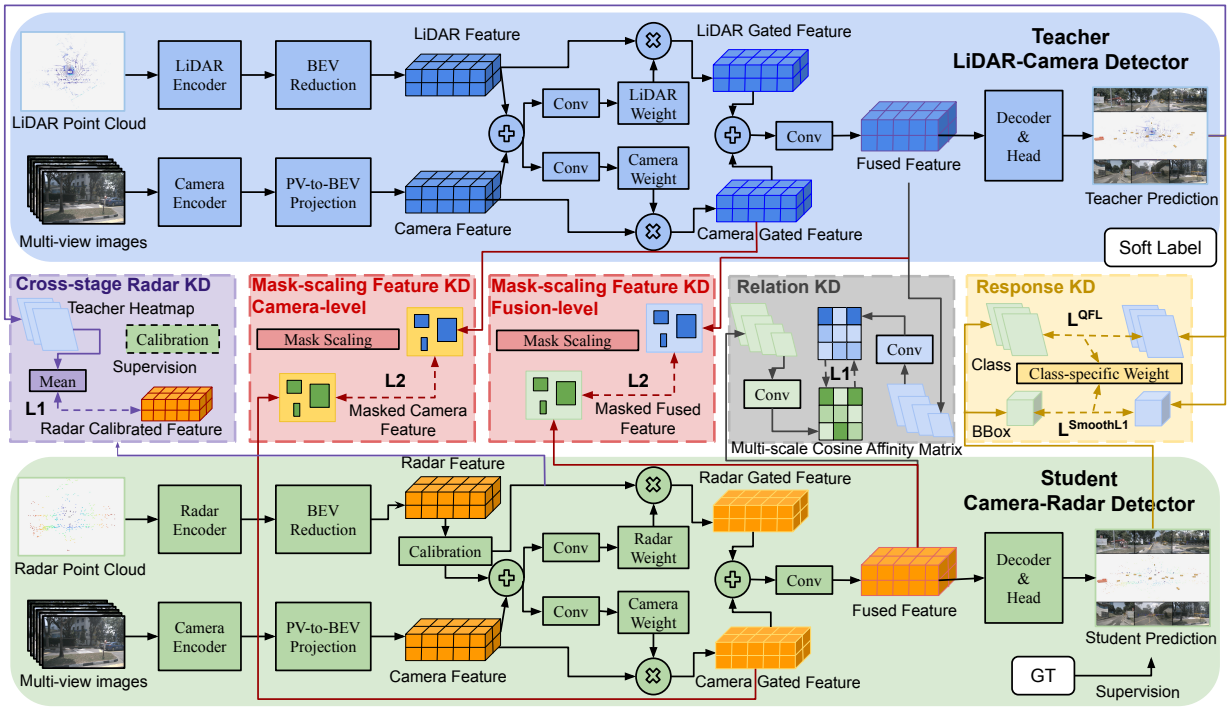



Figure 4.2: An overview of the proposed cross-modality (LC-to-CR) distillation framework CRKD. We narrow down the modality gap by unifying both the teacher and the student into the BEV space with similar 3D object detector structure. We refine the model design to enable adaptive fusion and design four novel distillation losses for effective cross-modality KD. During inference, only CR input is needed.

novel framework improves the detection performance of CR detectors comprehensively.

## 4.3 Method

We show an overview of CRKD in Figure 4.2. We set up the teacher and student models with a similar BEV-based encoder-decoder head architecture. Taking advantage of the shared BEV feature space, we build CRKD based on the highly optimized BEVFusion [107] codebase. We use BEVFusion-LC as the teacher model and BEVFusion-CR as the baseline student model. The detector head in both models is set as CenterHead [206] for response KD.

To account for the challenging cross-modality fusion-to-fusion KD, we design several KD modules. We propose cross-stage radar distillation with a learning-based calibration module to enable the radar encoder to learn a more accurate scene-level object distribution. A mask-scaling feature KD is designed for feature imitation on foreground regions while accounting

for inaccurate view transformation to BEV features for objects that are far from the sensor and dynamic. We apply a relation KD to maintain relation consistency in scene-level geometry. In addition, we improve the response KD design with class-specific loss weights to better leverage the CR model's ability to capture dynamic objects. Details of the proposed KD modules will be discussed in the following sections.

### 4.3.1 Model Architecture Refinement

We add a gated network [59, 207, 159] to BEVFusion [107] to enable the model to learn to generate attention weights on the single-modality feature maps to fuse the complementary modalities adaptively. Specifically, the gated network learns the gated features as follows:

$$\tilde{F}_{M_1} = F_{M_1} \times \sigma(\text{Conv}_{M_1}(\text{Concat}(F_{M_1}, F_{M_2}))), \tag{4.1}$$

$$\tilde{F}_{M_2} = F_{M_2} \times \sigma(\text{Conv}_{M_2}(\text{Concat}(F_{M_1}, F_{M_2}))), \tag{4.2}$$

where $\tilde{F}_{M_1}$ and $\tilde{F}_{M_2}$ are the gated features for modality $M_1$ and $M_2$, $F_{M_1}$ and $F_{M_2}$ are the input feature maps to the gated network from the backbone and view transform module of modality $M_1$ and $M_2$, respectively, $\sigma$ denotes the sigmoid function, and $\text{Conv}_{M_1}$ and $\text{Conv}_{M_2}$ are two separate convolution layers for $M_1$ and $M_2$ that could learn channel-wise attention weights for the input features. The output of the gated network is further fused by a convolutional fusion module in BEVFusion [107]. We apply the adaptive gated network to our teacher and student models to learn the relative importance between input modalities. This modification improves the detection performance of the teacher and student models, and also makes feature-based distillation more effective since the gated feature maps encode informative scene geometry from both input modalities.

In the LC teacher model, we denote the camera feature map as $F_c^T \in \mathbb{R}^{C_c^T \times H \times W}$ and the LiDAR feature map as $F_l^T \in \mathbb{R}^{C_l^T \times H \times W}$. Similarly, the camera and radar features in the student model are denoted as $F_c^S \in \mathbb{R}^{C_c^S \times H \times W}$ and $F_r^S \in \mathbb{R}^{C_r^S \times H \times W}$, respectively. We denote the fused feature map in the teacher model and the student model as $F_f^T$ and $F_f^S$. We keep the spatial dimension $H \times W$ consistent for all feature maps, and also set $C_c^T = C_c^S$ and $C_l^T = C_r^S$ for feature mimicking across the feature dimension.

### 4.3.2 Cross-Stage Radar Distillation (CSRD)

Though the measurements of radar and LiDAR are both represented as point clouds, the physical meanings behind them are slightly different. Compared with LiDAR, radar points are much sparser and can be interpreted as a list of object-level points with velocity measure-

ments [169, 199], while LiDAR is denser and captures geometry-level information. Observing this gap, we argue the common method of direct feature imitation may not work well in this scenario. Instead, as radar measurements are sparse and represent scene-level object distribution, we propose a novel Cross-Stage Radar Distillation (CSRD) method. Specifically, we design a distillation path between the radar feature map and the scene-level objectness heatmap predicted by the LC teacher model, which is denoted as $Y^T \in \mathbb{R}^{K\times H\times W}$, where $K$ is the number of classes. Since radar is generally believed to be noisy in range and azimuth angle measurements, we design a calibration module to learn to compensate for the noise. Specifically, we pass $F_r^S$ to three blocks of convolutional layer, batch normalization, and ReLU activation with kernel size $3 \times 3$. We add another $1 \times 1$ convolution layer to project the calibrated feature map to $\hat{F}_r^S \in \mathbb{R}^{H\times W}$. The CSRD loss $\mathcal{L}_{csrd}$ is formed as follows:

$$\mathcal{L}_{csrd} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left\| \hat{Y}_{i,j}^T - (\hat{F}_r^S)_{i,j} \right\|_1, \tag{4.3}$$

where $\hat{Y}^T \in \mathbb{R}^{H\times W}$ is obtained by taking the mean along the $K$ dimension of $\sigma(Y^T)$.

### 4.3.3 Mask-Scaling Feature Distillation (MSFD)

We propose feature distillation for aligning the camera feature maps and the fused feature maps. It has been acknowledged in many works [227, 21, 19, 221] that direct feature imitation between teacher and student models may not work effectively in 3D object detection tasks due to the notable imbalance between the foreground and background. Therefore, a common fix to this issue is to generate a mask $M \in \mathbb{R}^{H\times W}$ to only distill information from the foreground region. Meanwhile, more works have demonstrated that the boundary region of the foreground can also contribute to effective KD [19, 221]. We follow this finding and propose Mask-Scaling Feature Distillation (MSFD) that is aware of object range and movement. For the student CR model, the detection performance is mainly dependent on the depth prediction for images and the geometric accuracies of radar points. Since the range and object movement can cause extra challenges for view transformation to BEV space, we scale up the area of the foreground region to account for the potential misalignment. We increase the width and length of the mask by $\alpha$ and $\beta$ if the objects are in the range groups $[r_1, r_2]$ and $[r_2, \infty]$. We also increase the width and length by $\alpha$ and $\beta$ if the velocities along that axis are within $[v_1, v_2]$ and $[v_2, \infty]$. In practice, we clip the increase of object size within a pre-defined range to balance between different sizes of objects. We form the MSFD loss as

follows:

$$\mathcal{L}_{msfd} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} M_{i,j} \left\| F_{i,j}^{T} - F_{i,j}^{S} \right\|_2, \tag{4.4}$$

where $F^T$ and $F^S$ represent the feature maps in the teacher and student model, respectively. We compute MSFD loss for the gated camera feature maps ($\tilde{F}_c^T$ and $\tilde{F}_c^S$) and the fused feature maps ($F_f^T$ and $F_f^S$).

### 4.3.4 Relation Distillation (RelD)

While the aforementioned CSRD and MSFD can handle feature-level distillation effectively, we follow MonoDistill [21] to highlight the importance of maintaining similar geometric relations in the scene level between the teacher and student models. We compute the affinity matrix describing cosine similarity of the fused feature map. We propose the RelD loss as follows:

$$C_{i,j} = \frac{F_i^{\top} F_j}{\|F_i\|_2 \cdot \|F_j\|_2}, \tag{4.5}$$

where $C_{i,j}$ denotes the cosine similarity value at $(i, j)$ in the affinity matrix, and $F_i$ and $F_j$ represent the $i^{th}$ and $j^{th}$ feature map separately. The scene-level information gap between the student and teacher models can be computed as the $\mathcal{L}_1$ norm between their respective affinity matrices. We refer to the RelD loss as $\mathcal{L}_{reld}$, as shown below:

$$\mathcal{L}_{reld} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left\| C_{i,j}^{T} - C_{i,j}^{S} \right\|_1, \tag{4.6}$$

where $H$ and $W$ represent the BEV spatial size, and $C^T$ and $C^S$ denote the affinity matrix of the teacher and student model, respectively. In CRKD, we compute $\mathcal{L}_{reld}$ between the fused feature maps of the teacher and student models since they are the input to the decoder and detector heads. The refined feature maps with distilled relation information could improve the detection performance. Moreover, in order to distill the scene-level relation information of different scales, we apply a downsampling operation and a convolutional block. Then we use these multi-level feature maps to calculate the multi-scale RelD losses and take the average value as the final loss term.

### 4.3.5 Response Distillation (RespD)

Response Distillation has been proven effective in image classification [49] and 3D object detection [50, 227, 180]. The predictions inferred by the teacher are served as the soft labels for the student. The soft labels and the hard labels are combined to supervise the learning of the student model. We refer to the RespD design in CMKD [50] and improve it to be aware of modality strength. Since radar has the unique advantage of direct velocity measurements due to the Doppler effect [230, 199], we set larger weights for dynamic classes in RespD to allow higher priority for dynamic objects to leverage the student CR model's strength. The loss for dynamic response distillation is denoted as $\mathcal{L}_{resp}$, consisting of the classification loss $\mathcal{L}_{cls}$ and the regression loss $\mathcal{L}_{reg}$. $\mathcal{L}_{cls}$ is for object categories and is computed using the Quality Focal Loss (QFL) [86]. $\mathcal{L}_{reg}$ is for 3D bounding box regression and can be obtained by calculating the $SmoothL1$ loss. We compute these two losses as:

$$\mathcal{L}_{cls} = \sum_{i=1}^{K} QFL(P_{C_i}^T, P_{C_i}^S) \times w_i, \tag{4.7}$$

$$\mathcal{L}_{reg} = \sum_{i=1}^{K} \text{SmoothL1}(P_{B_i}^T, P_{B_i}^S) \times w_i, \tag{4.8}$$

where $P_{C_i}^T$ and $P_{C_i}^S$ denote the classification predictions of the $i^{th}$ task generated by the teacher and student model, and $P_{B_i}^T$ and $P_{B_i}^S$ denote the regression predictions in the teacher and student models. $K$ is the number of tasks in the CenterHead [206], and $w_i$ represents the class-specific weights.

### 4.3.6 Overall Loss Function

We combine the proposed KD loss and standard 3D object detection loss $\mathcal{L}_{det}$. The overall loss function we apply in the training stage is:

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_{csrd} + \lambda_2 \cdot \mathcal{L}_{msfd} + \lambda_3 \cdot \mathcal{L}_{reld} + \lambda_4 \cdot \mathcal{L}_{resp} + \lambda_5 \cdot \mathcal{L}_{det}, \tag{4.9}$$

where $\lambda_1$ through $\lambda_5$ are hyperparameters we set for weighting different loss components.

Table 4.1: Comparison on the nuScenes dataset. We group methods based on modality and whether KD is used. We include existing state-of-the-art (SOTA) works that use single-frame image input for fair comparison. Methods [19, 180, 221] omitted from the *test*-split group do not report their results with single-frame image input. mAP and NDS are percentages. The proposed CRKD outperforms the baseline methods in most metrics. ♢ denotes the distillation-based methods. † denotes using test-time augmentation. The best results are **bolded**, and the second-best results are underlined.

| Set | Method | Modality | Backbone | Resolution | mAP↑ | NDS↑ | mATE↓ | mASE↓ | mAOE↓ | mAVE↓ | mAAE↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *val* | BEVFormer-S [93] | C | R101 | 900 × 1600 | 37.5 | 44.8 | 0.725 | 0.272 | **0.391** | 0.802 | 0.200 |
| | BEVDet [54] | C | R50 | 256 × 704 | 29.8 | 37.9 | 0.725 | 0.279 | 0.589 | 0.860 | 0.245 |
| | RCM-Fusion [70] | C+R | R101 | 900 × 1600 | 44.3 | 52.9 | - | - | - | - | - |
| | CenterFusion [121] | C+R | DLA34 | 450 × 800 | 33.2 | 45.3 | 0.649 | **0.263** | 0.535 | 0.540 | **0.142** |
| | CRAFT [73] | C+R | DLA34 | 448 × 800 | 41.1 | 51.7 | 0.494 | 0.276 | 0.454 | 0.486 | 0.176 |
| | RCBEV [228] | C+R | SwinT | 256 × 704 | 37.7 | 48.2 | 0.534 | 0.271 | 0.558 | 0.493 | 0.209 |
| | BEVFusion [107] | C+R | SwinT | 256 × 704 | 43.2 | 54.1 | 0.489 | 0.269 | 0.512 | **0.313** | 0.171 |
| | UVTR (L2C) [87] | C♢ | R101 | 900 × 1600 | 37.2 | 45.0 | 0.735 | 0.269 | 0.397 | 0.761 | 0.193 |
| | BEVDistill (BEVFormer-S) [19] | C♢ | R50 | 640 × 1600 | 38.6 | 45.7 | 0.693 | 0.264 | 0.399 | 0.802 | 0.199 |
| | UniDistill (LC2C) [227] | C♢ | R50 | 256 × 704 | 26.5 | 37.8 | - | - | - | - | - |
| | BEVSimDet [221] | C♢ | SwinT | 256 × 704 | 40.4 | 45.3 | 0.526 | 0.275 | 0.607 | 0.805 | 0.273 |
| | X3KD (LC2C) [76] | C♢ | R50 | 256 × 704 | 39.0 | 50.5 | 0.615 | 0.269 | 0.471 | 0.345 | 0.203 |
| | DistillBEV (BEVDet) [180] | C♢ | R50 | 256 × 704 | 34.0 | 41.6 | 0.704 | 0.266 | 0.556 | 0.815 | 0.201 |
| | X3KD (L2CR) [76] | C+R♢ | R50 | 256 × 704 | 42.3 | 53.8 | - | - | - | - | - |
| | **CRKD** | C+R♢ | R50 | 256 × 704 | 43.2 | 54.9 | 0.450 | 0.267 | 0.442 | 0.339 | 0.176 |
| | **CRKD** | C+R♢ | SwinT | 256 × 704 | **46.7** | **57.3** | **0.446** | **0.263** | 0.408 | 0.331 | 0.162 |
| *test* | BEVFormer-S [93] | C | R101 | 900 × 1600 | 40.9 | 46.2 | 0.650 | 0.261 | 0.439 | 0.925 | 0.147 |
| | BEVDet† [54] | C | SwinT | 640 × 1600 | 42.4 | 48.2 | 0.528 | **0.236** | 0.395 | 0.979 | 0.152 |
| | RCM-Fusion [70] | C+R | R101 | 900 × 1600 | **49.3** | 58.0 | 0.485 | 0.255 | 0.386 | 0.421 | 0.115 |
| | CenterFusion† [121] | C+R | DLA34 | 450 × 800 | 32.6 | 44.9 | 0.631 | 0.261 | 0.516 | 0.614 | 0.115 |
| | CRAFT† [73] | C+R | DLA34 | 448 × 800 | 41.1 | 52.3 | 0.467 | 0.268 | 0.456 | 0.519 | 0.114 |
| | RCBEV [228] | C+R | SwinT | 256 × 704 | 40.6 | 48.6 | 0.484 | 0.257 | 0.587 | 0.702 | 0.140 |
| | UVTR (L2C) [87] | C♢ | V2-99 | 900 × 1600 | 45.2 | 52.2 | 0.612 | 0.256 | **0.385** | 0.664 | 0.125 |
| | X3KD (LC2C) [76] | C♢ | R101 | 640 × 1600 | 45.6 | 56.1 | 0.506 | 0.253 | 0.414 | **0.366** | 0.131 |
| | UniDistill (LC2C) [227] | C♢ | R50 | 256 × 704 | 29.6 | 39.3 | 0.637 | 0.257 | 0.492 | 1.084 | 0.167 |
| | X3KD (L2CR) [76] | C+R♢ | R50 | 256 × 704 | 44.1 | 55.3 | - | - | - | - | - |
| | **CRKD** | C+R♢ | SwinT | 256 × 704 | 48.7 | **58.7** | **0.404** | 0.253 | 0.425 | 0.376 | **0.111** |

# 4.4 Experiments

## 4.4.1 Experimental Setup

We evaluate our method on the nuScenes dataset [8] as the three modalities (i.e., LiDAR, camera and radar) are all available. We follow the official split that has 700 scenes for training and 150 scenes for validation. We set $[-54m, -54m, -5m] \times [54m, 54m, 3m]$ as the region to conduct object detection. We use the mean Average Precision (mAP) and nuScenes Detection Score (NDS) [8] as the main evaluation metrics. We also report the True Positive (TP) metrics [8] for comprehensive evaluation.

Our implementation is based on the MMDetection3D codebase [24]. All of our experiments are conducted using 4 NVIDIA A100 GPUs. We set the default BEVFusion-LC with CenterHead [206] as the teacher model. As mentioned in section 4.3.1, we add the adaptive

gated network to the baseline BEVFusion-CR model and denote it as BEVFusion-CR*. We set BEVFusion-CR* as the student model. We set the camera backbone as SwinT [105] and image resolution as $256 \times 704$ for both the teacher and student models. We also test CRKD with a ResNet R50 backbone [47] for comprehensive evaluation. The BEV spatial size is set to $180 \times 180$. We use PointPillars [79] as the backbone of the radar branch in BEVFusion-CR*. We set the same CenterHead [206] as the detector head for the student model.

For data processing, we follow the BEVFusion [107] pipeline. The teacher and student share the same camera processing pipeline with six surround-view images resized to $256 \times 704$. During training, we apply image scale jitter in the range of $[0.38, 0.55]$, random rotation in $[-5.4°, 5.4°]$, and standard image normalization. For the teacher model, the LiDAR branch loads the keyframe point cloud with 9 previous sweeps, and we apply random scaling in $[-0.9, 1.1]$ and translation augmentation within 0.5m. For the student model, the radar branch loads the keyframe with 6 previous sweeps and uses the same geometric augmentation for consistency. We follow BEVFusion to select the radar point attributes and apply the class-balanced grouping and sampling (CBGS) strategy during pretraining. We do not apply test-time augmentation to any model.

Before distillation, we separately pretrain BEVFusion-LC* and BEVFusion-CR* for 20 epochs using AdamW [109] with an initial learning rate of 2e-4, cosine annealing scheduling, and batch size 16. The object sampling strategy [198] is applied during the first 15 epochs of this pretraining stage. During CRKD training, we freeze the teacher, initialize the student from the pretrained BEVFusion-CR* checkpoint, disable object sampling, and train for 20 epochs with batch size 8 and learning rate 1e-4.

For the overall loss, we set $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ to 100, 10, 0.25, 1, and 1, respectively. In MSFD, we set the range thresholds $r_1 = 20$m and $r_2 = 30$m, the mask-scaling factors $\alpha = 0.25$ and $\beta = 0.5$, and the velocity thresholds $v_1 = 0.3$m/s and $v_2 = 0.8$m/s. We clip the mask expansion within $[0.5\text{m}, 4\text{m}]$ to avoid over-expanding objects of different sizes. For CSRD, we use an $\mathcal{L}_1$ loss instead of $\mathcal{L}_2$ because the radar branch is more affected by sparse measurements and outliers; empirically, this choice improves mAP by 0.4%.

### 4.4.2 Quantitative Results

We give an overall comparison of CRKD with existing CO and CR detectors with single-frame image input on nuScenes [8]. We follow common practices [19, 227, 76, 180] to show a complete comparison on both the *val* and *test* splits. As shown in Table 4.1, CRKD is the top-performing model in most metrics on the *val* set of nuScenes. We also present the

performance of CRKD on the *test* split. The results show that CRKD has the best or second best performance on most metrics without using any test-time optimization techniques (e.g., test-time augmentation, larger image resolution). Overall, CRKD is the most consistent in achieving high performance across all of the baselines.

We also show a complete comparison of per-class Average Precision (AP) in Table 4.2 to break down the improvement brought by CRKD. The results show that CRKD achieves consistent improvement in AP of all the classes. There is an interesting finding that larger gains of CRKD come from dynamic classes, which indicates that CRKD successfully helps the student model leverage its strength on dynamic object detection more effectively due to the availability of direct velocity measurements from radar.

Table 4.2: Comparison of the per-class AP results of the BEVFusion-LC (teacher), BEVFusion-CR (baseline), BEVFusion-CR* (student) and CRKD models on the nuScenes *val* split. AP and mAP are percentages; higher is better. The best results are **bolded**.

| Model | Modality | Car | Truck | Bus | Trailer | CV | Ped | Motor | Bicycle | TC | Barrier | mAP↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Teacher | L+C | 88.4 | 62.4 | 73.8 | 40.6 | 29.2 | 78.7 | 75.3 | 65.8 | 74.9 | 72.3 | 66.1 |
| Baseline | C+R | 72.1 | 37.8 | 48.9 | 18.3 | 12.6 | 48.4 | 42.0 | 33.8 | 58.8 | 59.6 | 43.2 |
| Student | C+R | 72.2 | 41.3 | 51.0 | 19.2 | 15.2 | 49.0 | 46.2 | 35.5 | 59.1 | **60.1** | 44.9 |
| **CRKD** | C+R | **74.8**(+2.7) | **44.1**(+6.3) | **53.6**(+4.7) | **20.6**(+2.3) | **16.9**(+4.3) | **50.6**(+2.2) | **46.8**(+4.8) | **38.2**(+4.4) | **61.5**(+2.7) | **60.1**(+0.5) | **46.7**(+3.5) |

Table 4.3: Ablation study of the proposed modules in CRKD evaluated on the nuScenes *val* split. The baseline is the BEVFusion-CR model. mAP and NDS are percentages. The best results are **bolded**.

| Model | Gated | RespD | CSRD | MSFD | RelD | mAP↑ | NDS↑ | mATE↓ | mASE↓ | mAOE↓ | mAVE↓ | mAAE↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | | | | | | 43.2 | 54.1 | 0.489 | 0.269 | 0.512 | 0.313 | 0.171 |
| CRKD | ✓ | | | | | 44.9 | 55.9 | 0.464 | 0.267 | 0.458 | **0.304** | 0.165 |
| | ✓ | ✓ | | | | 45.7 | 56.7 | 0.448 | 0.262 | 0.409 | 0.330 | 0.166 |
| | ✓ | ✓ | ✓ | | | 46.0 | 57.0 | 0.445 | 0.261 | 0.407 | 0.326 | 0.163 |
| | ✓ | ✓ | ✓ | ✓ | | 46.2 | 57.2 | **0.439** | **0.260** | **0.394** | 0.332 | 0.166 |
| | ✓ | ✓ | ✓ | ✓ | ✓ | **46.7** | **57.3** | 0.446 | 0.263 | 0.408 | 0.331 | **0.162** |

### 4.4.3 Ablation Studies

To further break down the improvement brought by each module we design, extensive experiments are conducted to discuss and validate our design choices. We first present the main ablation study in Table 4.3. It can be observed that all the proposed modules have contributed to the superior performance of CRKD. Among the four proposed KD losses, we see the most improvement comes from RespD, which indicates the significance of RespD in cross-modality KD. The other three losses contribute more to the improvement of mAP. This

Table 4.4: Ablation Study of single distillation modules on the nuScenes *val* split. mAP and NDS are percentages.

| Module | LiDAR | Heatmap | mAP↑ | NDS↑ |
|---|---|---|---|---|
| CSRD | ✓ | | 44.9 | 56.3 |
| | | ✓ | 46.0 | 57.0 |

(a) Ablation study of Cross-stage Radar Distillation (CSRD).

| Module | Mask | Mask-scaling | mAP↑ | NDS↑ |
|---|---|---|---|---|
| MSFD | ✓ | | 46.0 | 56.7 |
| | | ✓ | 46.2 | 56.9 |

(b) Ablation study of Mask-scaling Feature Distillation (MSFD).

| Module | Vanilla | Adapt | mAP↑ | NDS↑ |
|---|---|---|---|---|
| RelD | ✓ | | 45.9 | 56.9 |
| | | ✓ | 46.2 | 57.0 |

(c) Ablation study of Relation Distillation (RelD).

| Module | Vanilla | Dynamic | mAP↑ | NDS↑ |
|---|---|---|---|---|
| RespD | ✓ | | 45.3 | 56.7 |
| | | ✓ | 45.7 | 56.7 |

(d) Ablation study of Response Distillation (RespD).

Table 4.5: Ablation study of fusion module design and number of channels of the student model on the nuScenes *val* split. mAP and NDS are percentages.

| *Fuser* | *In Channels* | *Out Channels* | *mAP*↑ | *NDS*↑ |
|---|---|---|---|---|
| Conv | 80 + 256 | 256 | 43.2 | 54.1 |
| Gated | 64 + 64 | 64 | 44.2 | 54.3 |
| | 128 + 128 | 256 | 44.4 | 54.7 |
| | 80 + 256 | 256 | 44.9 | 55.9 |

finding validates our design objectives in improving object localization as CSRD and MSFD supervise the feature maps, and RelD tries to align the scene-level geometric relation.

We next demonstrate the experiments we conduct to validate the design choices of each module. As mentioned before, response distillation (RespD) brings the most improvement among all the proposed KD modules. Our empirical finding indicates that for other KD modules, the best-performing design alone may not be the best choice that works with RespD and other modules. We conjecture that this inconsistency in performance gain comes from the considerably large domain discrepancies between LC and CR. We would like to highlight the importance of RespD in the cross-modality KD work and use the experiments of combining different modules with RespD to guide the overall design of CRKD. Therefore, for the following ablation study in Table 4.4, unless otherwise mentioned, we show results of experiments with using RespD together. All the ablation study instances are trained using the same setting as the full model.

#### 4.4.3.1 Effect of CSRD

We conduct an ablation study to validate the proposed CSRD module. As mentioned in section 4.3.2, the radar points represent object-level information. Therefore, the common practice to distill information at the same stage of the network may not work well for radar feature maps. We propose CSRD to add cross-stage supervision on the object-level information. We demonstrate the model with CSRD outperforms that using LiDAR feature maps as the distillation source in Table 4.4a. The significant improvement from CSRD validates that the higher-level objectness heatmap provides more suitable guidance to distill radar features.

We further analyze several design choices in CSRD. As shown in Table 4.6, adding the radar calibration module improves the student detector, which supports our design choice to compensate for radar noise before aligning it with the teacher signal. We also compare using the teacher-predicted objectness heatmap and the ground-truth heatmap as the distillation source. The teacher heatmap performs better, indicating that the softened and scene-aware teacher response is a more suitable supervision target than hard labels. In addition, taking the mean over class channels works better than max pooling when constructing the heatmap target for radar distillation.

Table 4.6: Further ablation studies of Cross-Stage Radar Distillation (CSRD) on the nuScenes *val* split. mAP and NDS are percentages.

| Module | w/o calib | w/ calib | mAP↑ | NDS↑ |
|---|---|---|---|---|
| CSRD | ✓ | | 45.9 | 56.9 |
| | | ✓ | 46.0 | 57.0 |

(a) Calibration module.

| Module | GT | Teacher Heatmap | mAP↑ | NDS↑ |
|---|---|---|---|---|
| CSRD | ✓ | | 45.9 | 56.8 |
| | | ✓ | 46.0 | 57.0 |

(b) Distillation source.

| Module | max | mean | mAP↑ | NDS↑ |
|---|---|---|---|---|
| CSRD | ✓ | | 45.6 | 56.8 |
| | | ✓ | 46.0 | 57.0 |

(c) Heatmap pooling rule.

#### 4.4.3.2 Effect of MSFD

We conduct a comparison between the proposed mask-scaling strategy accounting for object range and velocity and the common foreground mask of ground truth bounding boxes. In Table 4.4b, the improvement achieved with the proposed mask-scaling strategy verifies that our mask-scaling strategy in MSFD helps to achieve more effective KD.

We also study several design choices for feature distillation. As shown in Table 4.7a, using the gated camera feature for distillation is better than using the ungated feature, which supports our motivation that the gated representation contains more informative scene geometry. Figure 4.3 visualizes this effect. We further compare distilling the camera feature, the fused feature, and both together. The results in Table 4.7b show that jointly distilling the camera

and fused features is the most effective choice. Finally, Table 4.7c shows that the proposed mask-scaling strategy outperforms dense supervision and a Gaussian-style foreground mask.

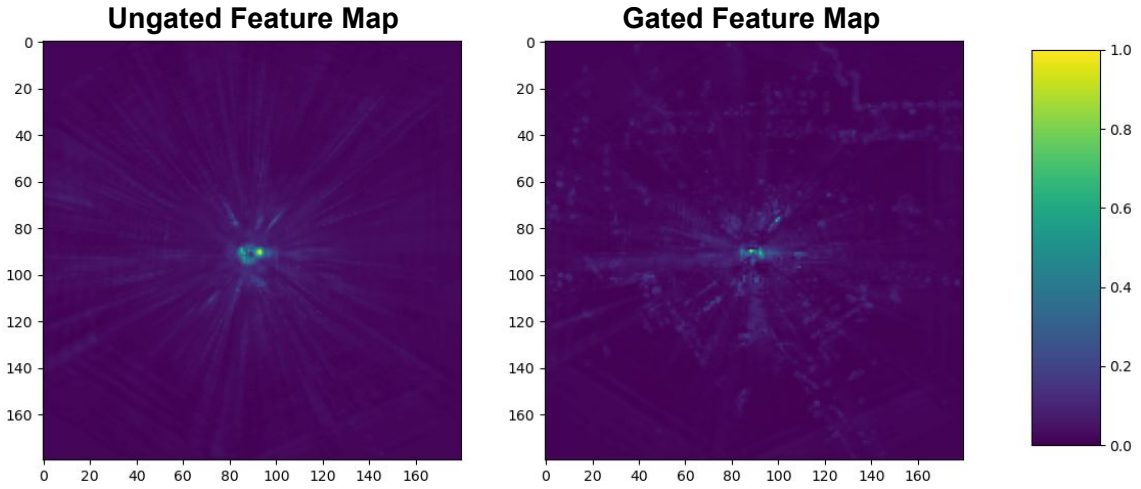



Figure 4.3: Visualization of ungated and gated camera feature maps in the teacher detector. We visualize the normalized activation magnitude of the camera feature map in bird's-eye view (BEV), where brighter colors indicate stronger feature responses. The ungated feature map shows more diffuse activations, while the gated feature map produces more localized and spatially structured responses after incorporating information from the LiDAR branch. This qualitative comparison illustrates how the gated representation can provide a cleaner feature target for distillation than the original ungated camera feature.

Table 4.7: Further ablation studies of Mask-Scaling Feature Distillation (MSFD) on the nuScenes *val* split. mAP and NDS are percentages.

| Module | Ungated | Gated | mAP↑ | NDS↑ |
|---|---|---|---|---|
| MSFD | ✓ | | 45.5 | 56.8 |
| | | ✓ | 45.7 | 56.9 |

(a) Gated feature source.

| Module | Cam | Fused | Cam&Fused | mAP↑ | NDS↑ |
|---|---|---|---|---|---|
| MSFD | ✓ | | | 45.7 | 56.9 |
| | | ✓ | | 45.8 | 56.7 |
| | | | ✓ | 46.2 | 56.9 |

(b) Distillation location.

| Module | Dense | Gaussian | Ours | mAP↑ | NDS↑ |
|---|---|---|---|---|---|
| MSFD | ✓ | | | 45.7 | 56.8 |
| | | ✓ | | 45.5 | 56.7 |
| | | | ✓ | 46.0 | 57.0 |

(c) Mask generation strategy.

#### 4.4.3.3 Effect of RelD

We study the effect of applying a convolutional block after the downsampling operation for the feature maps used in RelD. We name the instance with the convolution layer as Adapt and the baseline instance as Vanilla in Table 4.4c. The results verify our design choice.

#### 4.4.3.4 Effect of RespD

Though RespD has been a widely used loss term in several KD works [50, 227, 180], we are the first to weight the RespD loss differently based on different classes. In practice, we set $w_i$ as 2 for the dynamic classes and 1 for static classes. This design allows the training supervision to prioritize dynamic classes, which radars are more capable of detecting. In the vanilla setting we set $w_i$ as 1 for all classes. As shown in Table 4.4d, applying the class-specific weights helps to improve the overall performance of the student detector. We further compare this dynamic weighting with a static-priority setting in Table 4.8. The results show that emphasizing dynamic classes is more effective than emphasizing static classes, suggesting that RespD works best when it amplifies the strength of the camera-radar student rather than compensating only for its weaknesses.

Table 4.8: Further ablation study of Response Distillation (RespD) with different class-weight settings on the nuScenes *val* split. mAP and NDS are percentages.

| Module | Vanilla | Static | Dynamic | mAP↑ | NDS↑ |
|---|---|---|---|---|---|
| RespD | ✓ | | | 45.3 | 56.7 |
| | | ✓ | | 45.4 | 56.6 |
| | | | ✓ | 45.7 | 56.7 |

#### 4.4.3.5 Effect of Model Architecture Refinement

We study the effect of adding the adaptive gated network to the original fusion module in BEVFusion [107] and tuning the number of channel dimensions. As shown in Table 4.5, the addition of the gated network brings notable improvement over the default fusion module in the baseline BEVFusion-CR model. We also notice that matching the input and the output channel dimension of the fusion module to the teacher model (Camera: 80, LiDAR: 256) brings additional improvement. The following KD operation also benefits from the same channel dimension setting between the teacher and student models since no additional channel-wise projection is needed for feature-level KD.

### 4.4.4 Improvement Analysis

Since CRKD performs a novel LC-to-CR distillation path, we further break down the performance gains by object range and scene condition. As shown in Table 4.9, CRKD improves the student model across all range groups. The largest gain appears in the short-range

regime, where the teacher's LiDAR branch provides especially reliable geometric supervision. At the same time, the long-range mAP gain remains positive, suggesting that CRKD does not simply copy the teacher's behavior but can also strengthen the student detector in scenarios where radar measurements are advantageous.

Table 4.9: Performance breakdown by range on the nuScenes *val* split. We report the gain of CRKD over the student model. mAP and NDS are percentages.

| Method | Modality | NDS↑ | | | mAP↑ | | |
|---|---|---|---|---|---|---|---|
| | | [0m, 20m] | [20m, 30m] | [30m, 50m] | [0m, 20m] | [20m, 30m] | [30m, 50m] |
| Teacher | L+C | 76.71 | 68.63 | 50.57 | 77.11 | 62.37 | 38.25 |
| Student | C+R | 63.03 | 52.87 | 38.86 | 58.54 | 38.64 | 19.50 |
| **CRKD** | C+R | **65.53(+2.50)** | **53.52(+0.65)** | **39.21(+0.35)** | **61.59(+3.05)** | **39.04(+0.40)** | **20.53(+1.03)** |

We also group scenes according to weather and lighting conditions in Table 4.10. CRKD improves the student model in all four categories, with the most noticeable gain in rainy scenes. This result supports our motivation that cross-modality KD can help the student better exploit radar's robustness under adverse conditions while still benefiting from the teacher's stronger geometric prior.

Table 4.10: Performance breakdown by weather and lighting on the nuScenes *val* split. We report the gain of CRKD over the student model. mAP and NDS are percentages.

| Method | Modality | NDS↑ | | | | mAP↑ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Sunny | Rainy | Day | Night | Sunny | Rainy | Day | Night |
| Teacher | L+C | 70.22 | 71.01 | 70.54 | 44.92 | 66.02 | 65.48 | 66.25 | 41.12 |
| Student | C+R | 55.60 | 57.56 | 56.37 | 33.40 | 44.73 | 47.27 | 45.78 | 23.94 |
| **CRKD** | C+R | **56.56(+0.96)** | **59.97(+2.41)** | **57.59(+1.22)** | **34.22(+0.82)** | **45.95(+1.22)** | **49.59(+2.32)** | **47.16(+1.38)** | **24.14(+0.20)** |

### 4.4.5 Qualitative Results

We show the visualization of the 3D object detection results to highlight the effectiveness of CRKD in Figure 4.4. With CRKD, the detector is able to predict fewer false positive predictions and localize the objects better. We also show an impressive comparison between the teacher LC model and CRKD to demonstrate that CRKD can even outperform the teacher model with the help of radar measurements. This qualitative example validates the effectiveness of the CRKD framework and the value of radars for modern perception for autonomous driving.

We include more qualitative examples in Figure 4.5. The additional cases show three recurring behaviors. First, CRKD consistently corrects missed detections from the student model by transferring stronger geometric cues from the teacher. Second, radar measurements

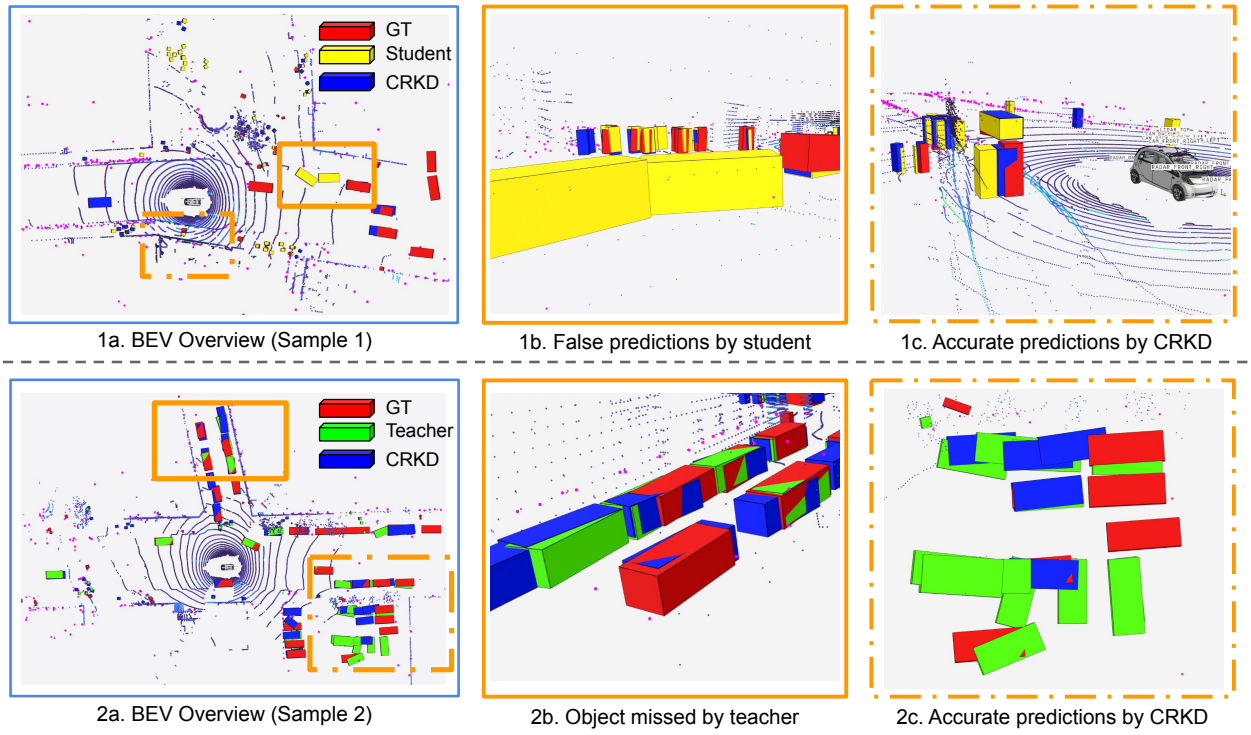



Figure 4.4: Qualitative results on nuScenes. We show zoomed-in views in panel b and c for the highlighted regions in panel a, with dashed borders indicating the corresponding regions. We show the ground truth annotation in red, teacher prediction in green, student prediction in yellow, CRKD prediction in blue, and radar points in magenta. In (1a) to (1c), we show an example frame where CRKD has more accurate predictions and fewer false predictions than the student model. In (2a) to (2c) we show another example frame where CRKD even outperforms the LC teacher by detecting a missed car and rejecting several false predictions. Best viewed on screen and in color.

can help CRKD reject false positives from both the student and, in some cases, the teacher detector. Third, CRKD is capable of outperforming both models in challenging scenes by combining the teacher's supervision with the complementary long-range and motion-sensitive cues provided by radar.

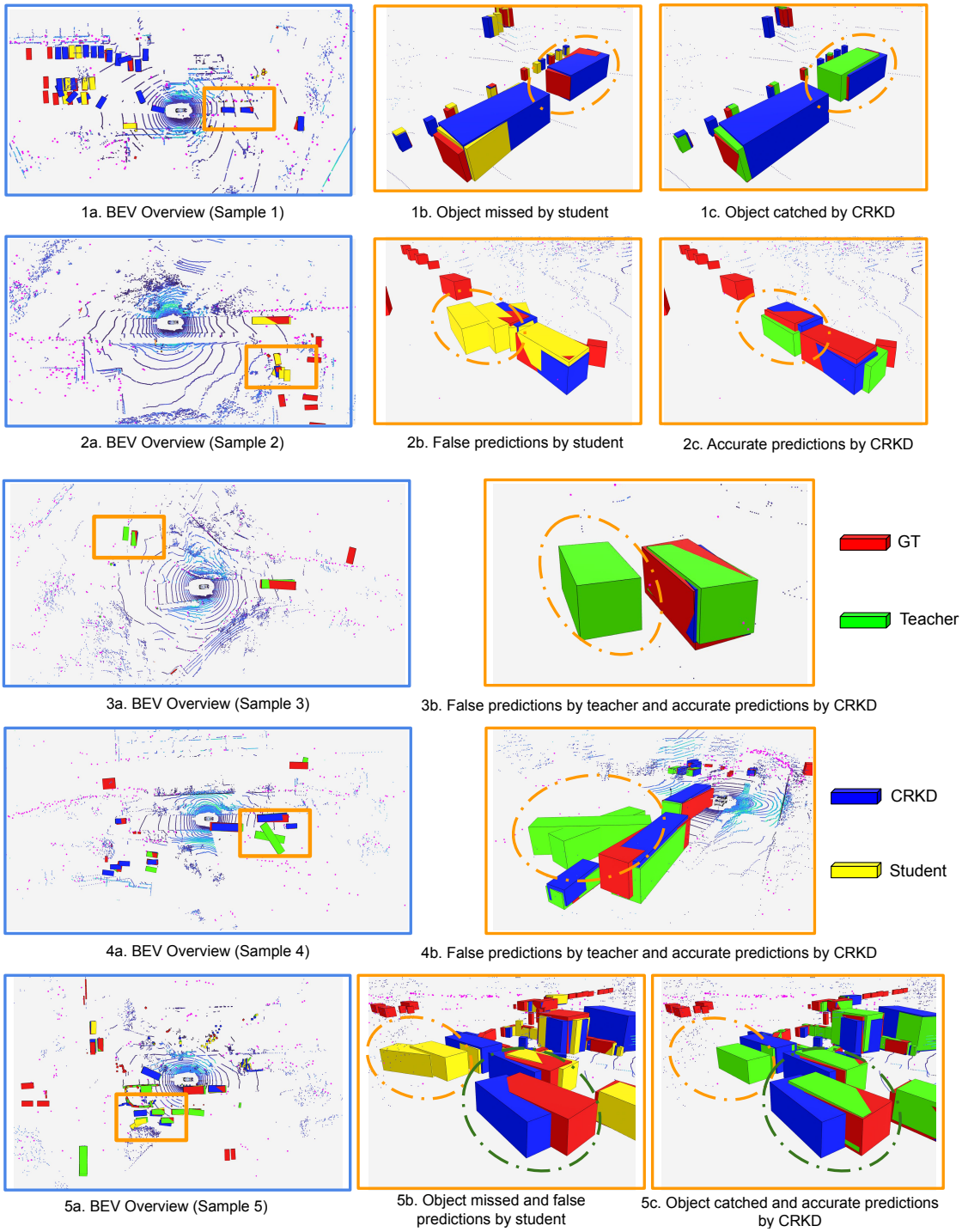



Figure 4.5: Additional qualitative results on nuScenes. We show zoomed-in views for the highlighted regions and compare the ground truth annotation (red), teacher prediction (green), student prediction (yellow), CRKD prediction (blue), and radar points (magenta). Samples 1 and 2 show that CRKD captures missed objects and rejects false positives from the student model. Samples 3 and 4 show cases where CRKD also rejects false detections produced by the teacher. Sample 5 shows a difficult frame where CRKD outperforms both teacher and student by recovering a missed object while keeping fewer false positives.

## 4.5 Conclusion

In this chapter, we have proposed CRKD, a novel KD framework that supports a cross-modality fusion-to-fusion KD path for 3D object detection. We leverage the BEV space to design a novel LC-to-CR KD framework. We design four distillation losses to address the significant domain gap and facilitate the distillation process in this cross-modality setting. We also introduce the adaptive gated network to learn the relative importance between two expert feature maps. Extensive experiments show the effectiveness of CRKD in improving the detection performance of CR detectors. We hope CRKD will inspire future research to leverage our proposed KD framework to further explore the potential of CR detectors to improve the reliability of this widely accessible sensor suite. In future work, we plan to extend the proposed CRKD framework to other perception tasks such as occupancy mapping.

# CHAPTER 5

# Temporal Reasoning for Online Vectorized HD Map Construction

## 5.1 Introduction

With recent advances in leveraging bird's-eye view (BEV) representations, perception capabilities of autonomous vehicles (AVs) have been largely enhanced [93, 135, 107, 159, 223]. Among all the BEV perception tasks, constructing high-definition (HD) maps in an online manner has received increasing attention from the AV research community [104, 85, 97, 210]. Specifically, online HD map construction aims to use onboard vehicle sensor data to reconstruct road elements such as lanes, pedestrian crossings, and road boundaries on the fly. The reconstructed HD maps are essential for downstream planning tasks such as prediction and planning [41]. This task is of great significance in advancing large-scale deployment of AVs as it avoids the need for a sophisticated and labor-intensive offline HD map generation process [104].

Existing works have demonstrated the effectiveness of formulating online HD map construction in an end-to-end framework [104, 108, 140, 210]. Though significant advances have been made in this field, many methods fall short of leveraging temporal information, which has been shown to be very beneficial, especially during instances of occlusion [210, 172, 15]. StreamMapNet [210] is the first end-to-end vectorized HD map construction work leveraging temporal cues. By maintaining a recurrent BEV memory feature map and propagating map queries, StreamMapNet conducts temporal fusion effectively and significantly outperforms previous non-temporal baselines [104, 97], demonstrating the importance of incorporating temporal cues for online HD map construction. The success of StreamMapNet [210] has inspired follow-up works to further improve it by incorporating auxiliary learning tasks or by integrating ideas from the tracking literature [172, 15].

Still, we highlight the challenges of directly accumulating all the temporal features into a single memory feature map, as proposed in [210]. The network can struggle to reason about

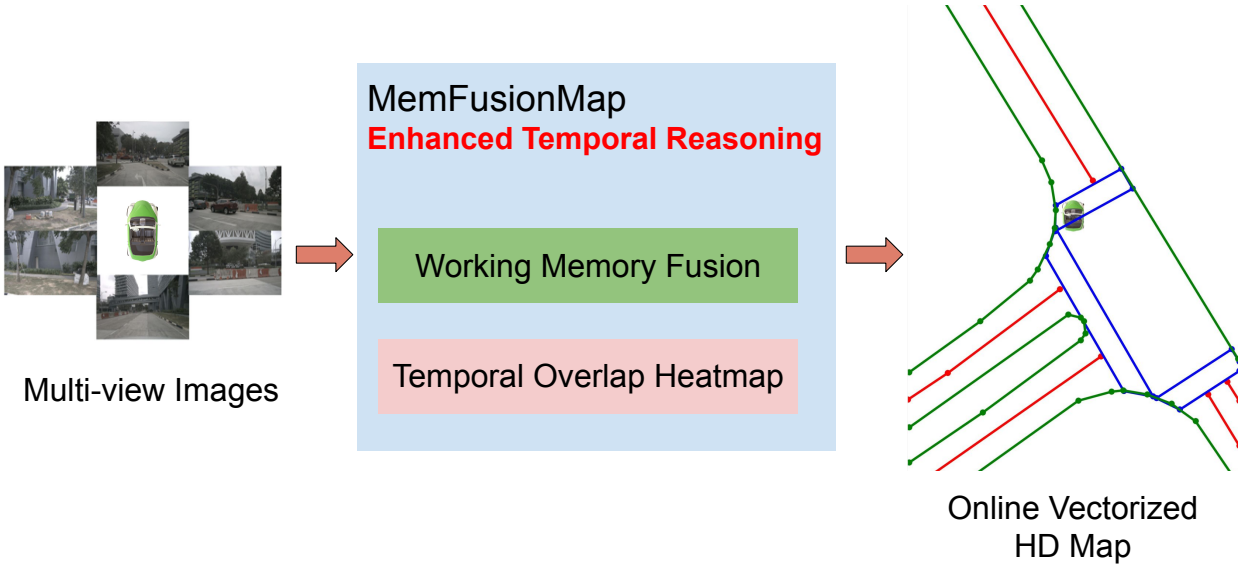



Figure 5.1: We propose MemFusionMap to improve the temporal reasoning capability for online vectorized HD map construction. MemFusionMap takes in multi-view images and leverages the proposed working memory fusion module and temporal overlap heatmap to output a vectorized HD map online.

the entire history due to limited memory capacity in complex road environments since the network only has access to the latest memory feature and current BEV feature. Furthermore, this design can struggle with occlusion. A typical example is when a sudden occlusion occurs due to a moving truck near the ego-vehicle. This can trigger a bad update to the memory feature due to challenging 2D-to-BEV projection, and thus affect all future predictions. A similar issue is also noted in the video object segmentation field [20, 226]. We share the same insight that maintaining working memory features from a subset of historical frames is helpful, which is also pointed out by [15]. Lastly, we argue the importance of maintaining explicit temporal overlap information, as the model can leverage this input to reason about temporal information more effectively.

Given these insights, we propose MemFusionMap (Figure 5.1), a novel memory fusion framework designed for online vectorized HD map construction. We design a working memory buffer to maintain working memory features in a fixed-lag manner. The working memory features are propagated recursively to align with the current field of view of the ego-vehicle. The fixed-lag design ensures bounded memory usage to fit real-world deployment needs. In addition, we propose a novel use of a temporal overlap heatmap, represented as a BEV image with a single channel. The heatmap score indicates the number of times each grid cell falls within the field-of-view. We recurrently propagate the temporal overlap heatmap. To facilitate the model's temporal reasoning capabilities, the temporal overlap heatmap is fused

with the working memory features by a convolutional temporal fusion block, whose output is fed into a standard detector head to generate road element predictions.

To summarize, our main contributions are as follows:

- We propose a simple yet effective model to fuse working memory features in BEV space for online vectorized HD map construction. MemFusionMap focuses on improving the network's temporal reasoning capability while also maintaining a versatile design for scalability and compatibility.
- We propose a novel design of maintaining a temporal overlap heatmap, providing a strong cue for the model to reason across a history of frames and also implicitly encoding valuable insights of the vehicle's trajectory.
- We conduct extensive evaluation on nuScenes [8] and Argoverse2 [185] to demonstrate the effectiveness of MemFusionMap. The proposed method significantly outperforms the state-of-the-art method [210], achieving a maximum improvement of 5.4% in mAP.

## 5.2 Related Work

### 5.2.1 Online Vectorized HD Map Construction

Constructing HD maps is one of the key tasks for AV perception. While traditional simultaneous localization and mapping (SLAM)-based methods [214, 153] suffer from high cost during offline construction and maintenance, online HD map construction solutions have obtained increasing attention as they can effectively simplify the map construction process and handle map changes. HDMapNet [85] marks a significant step toward vectorized HD map representations, which are preferred by downstream tasks over the rasterized map representations generated from previous segmentation methods [93, 135, 225] or lane detection methods [36, 16, 170]. VectorMapNet [104] proposes the first end-to-end vectorized HD map construction framework where a DETR-based map decoder [11] is used to generate map element predictions with an autoregressive transformer. Its performance is further improved by MapTR [97] and MapTRV2 [98] by treating the map construction task as a point set prediction problem and adding auxiliary tasks and decoder design improvements. Other works focus on different map element representations: BeMapNet [140] and PivotNet [29] explore using Bézier curves and pivot-based points, respectively. Additionally, other methods investigate network design improvement and training pipeline enhancements [103, 108, 195, 14]. More recently, StreamMapNet [210] and its follow-up works [15, 172] demonstrate notable improvement by leveraging temporal fusion for this task.

### 5.2.2 Temporal Fusion for HD Map Construction

Fusing temporal information has proven effective in various works in domains like 3D object detection and scene completion [43, 187, 171, 133, 93]. HDMapNet [85] is the first to demonstrate the benefits of temporal fusion in online HD map construction. It directly applies max pooling to compress temporal information into a BEV feature map, which is fed into the decoder. StreamMapNet [210] develops the first temporal fusion pipeline in an end-to-end framework for this task. In addition to propagating and reusing queries in the decoder, it proposes a BEV temporal fusion paradigm leveraging the streaming strategy following [171, 43]. StreamMapNet [210] compresses the history of information into a latent memory feature and propagates the latent memory feature recurrently. It applies a Gated Recurrent Unit (GRU) [22] to fuse the latent memory feature with the current BEV feature from the BEV feature encoder.

The success of StreamMapNet has motivated subsequent works such as SQD-MapNet [172] and MapTracker [15]. Inspired by DN-DETR [83], SQD-MapNet [172] designs a stream query denoising approach to further improve over StreamMapNet. However, SQD-MapNet is only evaluated on the original splits of nuScenes [8] and Argoverse2 [185] that are prone to overfitting, contradicting the main motivation of online HD map construction. MapTracker [15] explores leveraging additional ground truth of tracked road elements to enhance the query propagation paradigm. MemFusionMap shares the same intuition as MapTracker [15] that the redundancy of maintaining a subset of past frames can improve the model's robustness. However, we design MemFusionMap with a strong emphasis on scalability and versatility for practical application. Notably, while MapTracker fuses the last 20 frames of memory latents selected according to the distance strides, our proposed method, MemFusionMap, designs a simple yet effective memory fusion module, which only uses 4 past frames to avoid computational overhead and to achieve better training efficiency and runtime. MemFusionMap also covers the perception range of $100 \times 50\,m$, which has more practical value for real-world AV deployment, while MapTracker [15] omits this extended perception range. Furthermore, SQD-MapNet [172] and MapTracker [15] require implementing additional task and loss modules, which can cause extra overhead when integrating into existing multi-task AV perception pipelines. MapTracker also requires additional post-processing to generate the tracks of map elements, which can be time-consuming and potentially incompatible with large-scale databases that use an autolabeling process that cannot guarantee quality-assured labels for all frames. In contrast, MemFusionMap maintains a compact design with better versatility that can be directly applied to existing pipelines by switching the BEV temporal fusion module. Lastly, MemFusionMap also proposes a novel method for maintaining and integrating a temporal overlap heatmap, which is demonstrated to significantly improve the

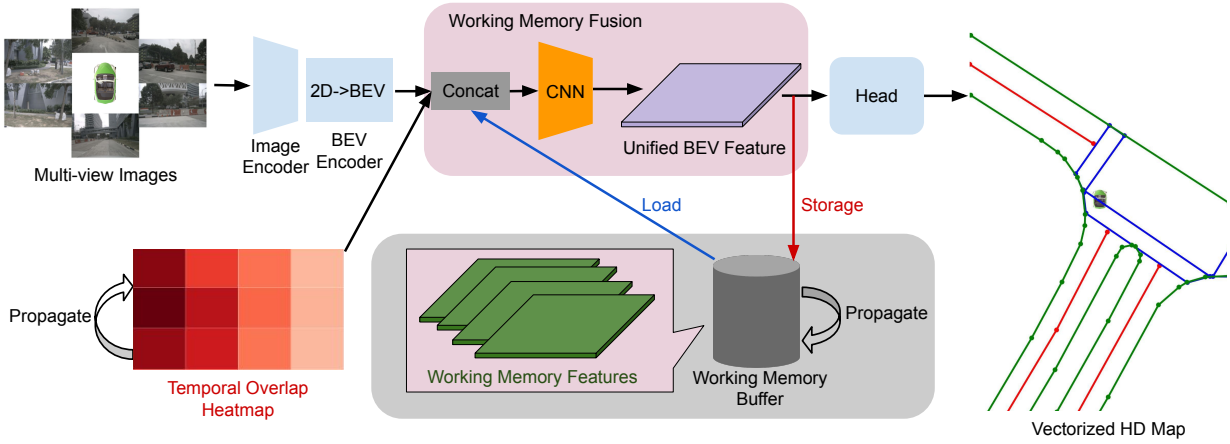



Figure 5.2: Overall architecture of MemFusionMap. MemFusionMap takes in multi-view images and extracts a BEV feature via an image encoder and a BEV feature encoder. The encoded BEV feature is fused with working memory features and the temporal overlap heatmap to obtain a unified BEV feature. The unified feature is fed into a decoder head to generate HD map prediction.

model's temporal reasoning capability.

## 5.3 Method

The overall framework of MemFusionMap is shown in Figure 5.2. The pipeline of MemFusionMap is inspired by StreamMapNet [210]. However, MemFusionMap proposes a novel structure for the temporal BEV fusion module. MemFusionMap first uses a shared ResNet-50 [47] image encoder to extract image features. A BEV feature encoder further projects the image features from 2D to BEV space. The projected BEV feature is denoted as $\mathcal{F}_{\text{BEV}} \in \mathbb{R}^{C \times H \times W}$, where $C$ is the BEV feature dimension, and $H$ and $W$ represent the spatial dimensions of the BEV feature. The BEV feature is fed into the memory fusion block, which will be introduced in the following sections. The output of the memory fusion block is a unified BEV feature map containing temporal cues, denoted as $\tilde{\mathcal{F}}_{\text{BEV}} \in \mathbb{R}^{C \times H \times W}$. Following [210], we use a DETR-based transformer decoder [11] with a multi-point attention mechanism for extended perception range. We also maintain a memory buffer to properly propagate and reuse history map queries.

The following sub-sections discuss the key contributions of the proposed architecture in more detail. The core contribution of MemFusionMap is its temporal BEV fusion design, which consists of a temporal overlap heatmap (section 5.3.1) and the working memory fusion module (section 5.3.2).

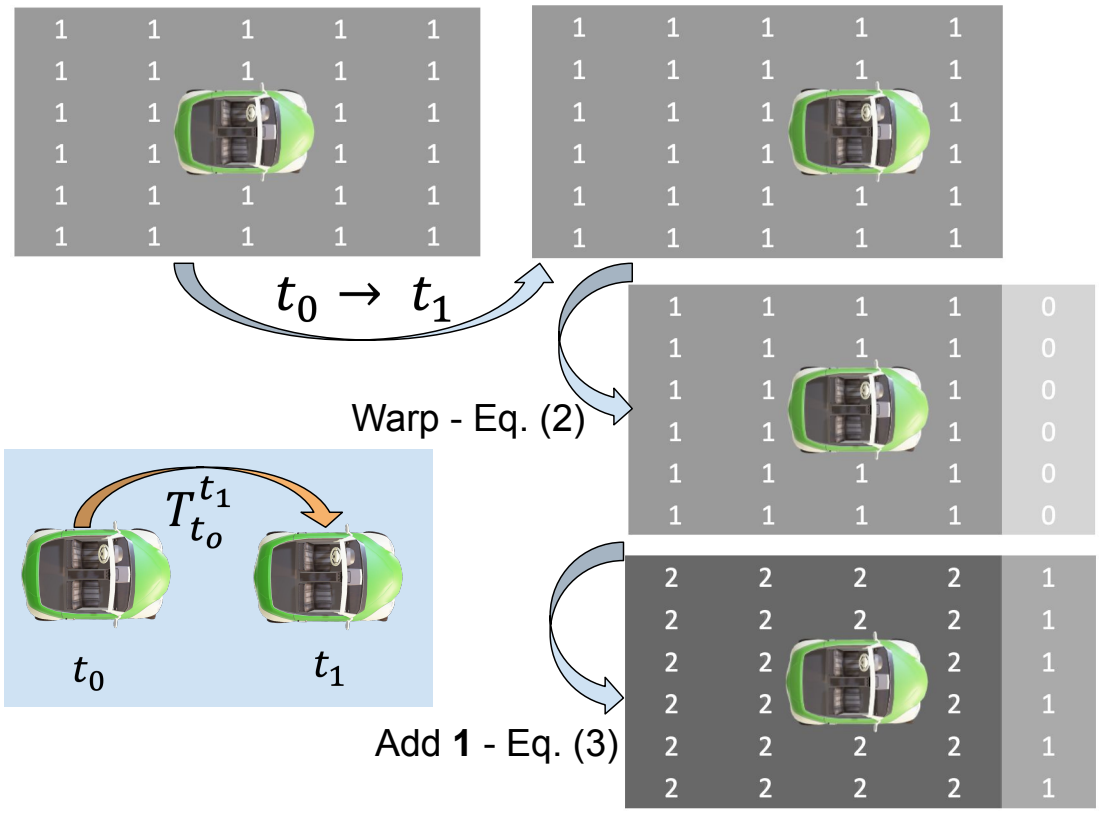



Figure 5.3: Example propagation process of the temporal overlap heatmap from $t_0$ to $t_1$.

### 5.3.1 Temporal Overlap Heatmap

We propose a novel paradigm to maintain a temporal overlap heatmap to explicitly inform the model of the temporal overlap information. Figure 5.3 illustrates the process of maintaining the heatmap. Starting from the first frame of each clip, we initialize the temporal overlap heatmap $\mathbf{H}_{t_0}$ as follows:

$$\mathbf{H}_{t_0} = \mathbf{1}^{1\times H\times W}. \tag{5.1}$$

Specifically, $\mathbf{H}_{t_0}$ is a single-channel heatmap with the same spatial size as the BEV feature map $\mathcal{F}_{\text{BEV}}$, where $H$ and $W$ denote the height and width of the BEV feature grid, respectively. Each pixel of $\mathbf{H}$ indicates the temporal overlap score of the corresponding cell in the BEV space. $\mathbf{H}_{t_0}$ is initialized with all ones, indicating that the whole BEV region has not been observed by previous frames at the beginning of the clip. Then, as the second frame ($t_1$) arrives, we first propagate the temporal overlap heatmap to the new vehicle position as follows:

$$\hat{\mathbf{H}}_{t_1} = \text{Warp}(\mathbf{H}_{t_0}, T_{t_0}^{t_1}), \tag{5.2}$$

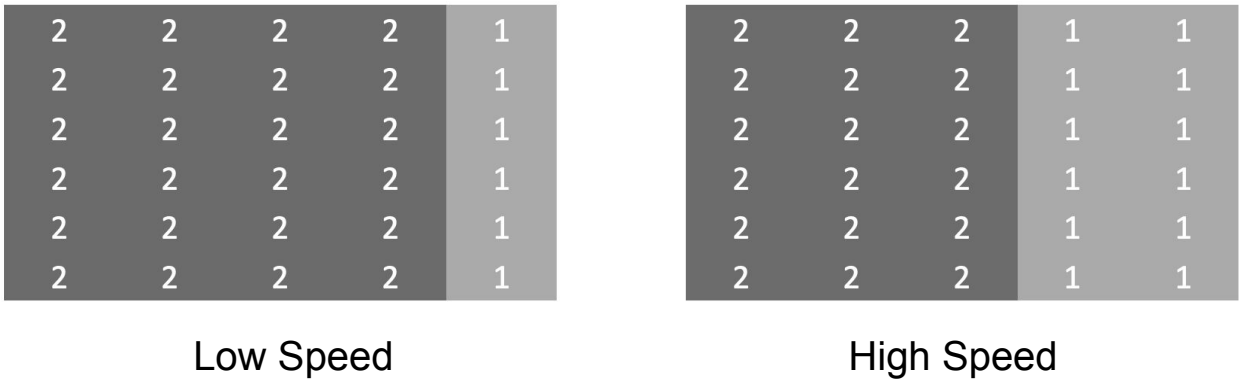



Figure 5.4: Example temporal overlap heatmaps resulting from different speeds. The comparison demonstrates that the overlap heatmap encodes vehicle moving speed.

where $\hat{\mathbf{H}}_{t_1}$ is the propagated and warped overlap heatmap from $t_0$ to $t_1$ according to the egomotion $T_{t_0}^{t_1}$. The warping is implemented using the grid sample function in PyTorch with zero padding mode for the out-of-bound grids. We then obtain the final heatmap for $t_1$ as:

$$\mathbf{H}_{t_1} = \hat{\mathbf{H}}_{t_1} + \mathbf{1}^{1\times H\times W}. \tag{5.3}$$

Following this strategy, we keep updating the temporal overlap heatmap recurrently. At each timestamp, the overlap heatmap is fed into the memory fusion block, which will be discussed in section 5.3.2.

One main benefit of maintaining this temporal overlap heatmap is that it provides informative insights into the regions with heavy overlaps and those without. With this information, the model can more adaptively reason across the current BEV feature and the historical features. Specifically, in the overlap heatmap $\mathbf{H}_t$, a smaller value indicates that a region is less temporally overlapped, meaning more information from the current BEV feature should be trusted. Conversely, a larger value suggests a region with greater overlap, implying that information from the memory latents could be more valuable. Furthermore, the overlap heatmap naturally encodes the vehicle trajectory, providing the model with additional knowledge about vehicle motion to enable more adaptive temporal reasoning.

We demonstrate this benefit with two visual examples. As shown in Figure 5.4, we present two example overlap heatmaps resulting from different speeds. The heatmap on the right exhibits a larger area with a value of **1**, indicating that a larger area is newly seen, which is a result of higher-speed egomotion. We present another example of a vehicle turning in Figure 5.5. The propagated overlap heatmap reveals two noticeable edges at the corner.

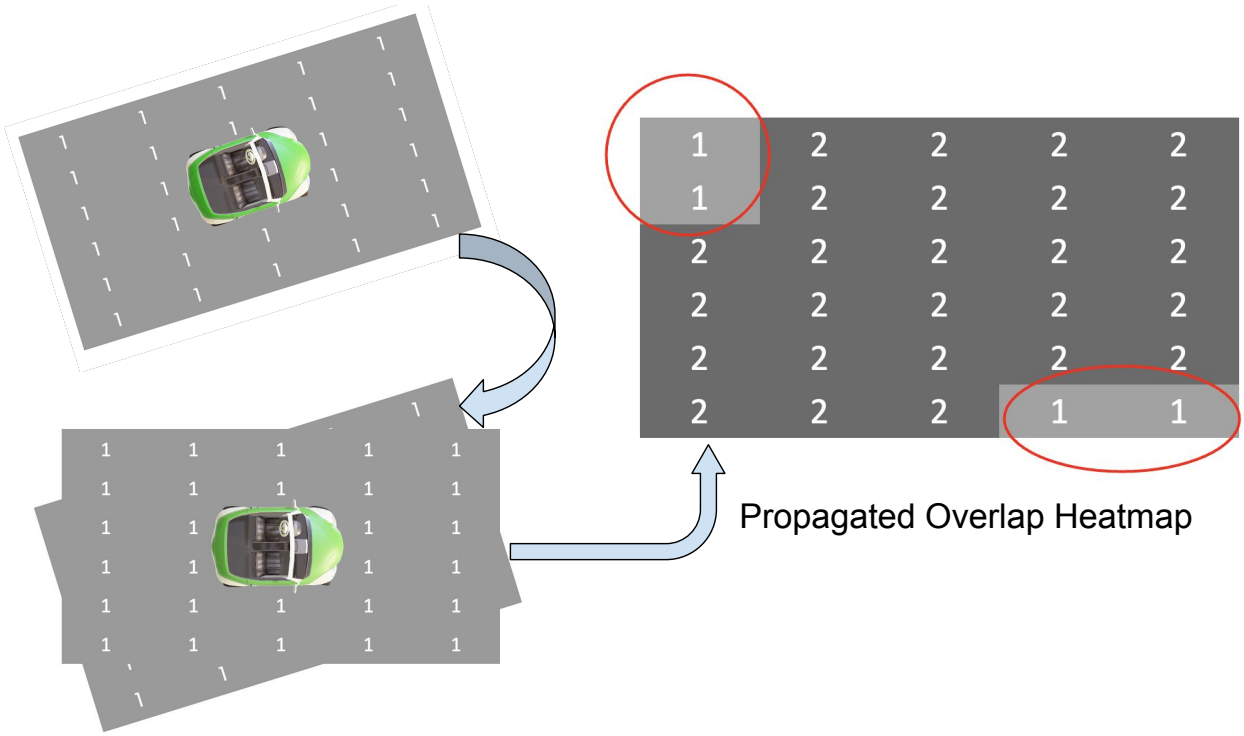



Figure 5.5: Example temporal overlap heatmap resulting from a vehicle turning. We highlight the two noticeable edges at the corner. This pattern encodes insights into the vehicle's trajectory.

These low-level feature patterns can be easily detected by a convolutional neural network, providing critical insights into the vehicle's trajectory.

### 5.3.2 Memory Fusion Block

We design a memory fusion block to synthesize memory features with the temporal overlap heatmap. As discussed in section 5.1, we highlight the importance of increasing the memory capacity for better temporal reasoning capability and robustness against occlusion. We aggregate a fixed number of BEV features from historical frames to fuse with the current BEV feature. We follow [20] to use the terminology used in cognitive science [30, 25, 191] and refer to the subset of historical frames as working memory features. As shown in Figure 5.2, we develop a working memory buffer to properly handle storage and loading of working memory features. We also design a memory fusion block to temporally reason across working memory features, aided by the temporal overlap heatmap.

#### 5.3.2.1 Working Memory Buffer

We design a working memory buffer to manage the memory features and update them recurrently. In the streaming training schedule, at time $t$, before memory fusion, the working memory feature $\mathcal{F}^t_{\text{WM}}$ is defined as follows:

$$\mathcal{F}^t_{\text{WM}} = \{\tilde{\mathcal{F}}^{t-T_{\text{WM}}}_{\text{BEV}}, \tilde{\mathcal{F}}^{t-T_{\text{WM}}+1}_{\text{BEV}}, \ldots, \tilde{\mathcal{F}}^{t-2}_{\text{BEV}}, \tilde{\mathcal{F}}^{t-1}_{\text{BEV}}\}, \tag{5.4}$$

where $\tilde{\mathcal{F}}^t_{\text{BEV}}$ is the unified BEV feature after memory fusion at time $t$, and $T_{\text{WM}}$ is the working memory capacity. Progressing towards time $t+1$, we first add the new feature and drop the oldest feature:

$$\mathcal{F}^{t+1}_{\text{WM}} = \{\tilde{\mathcal{F}}^{t-T_{\text{WM}}+1}_{\text{BEV}}, \tilde{\mathcal{F}}^{t-T_{\text{WM}}+2}_{\text{BEV}}, \ldots, \tilde{\mathcal{F}}^{t-1}_{\text{BEV}}, \tilde{\mathcal{F}}^{t}_{\text{BEV}}\}. \tag{5.5}$$

We also warp all the working memory features to align with $\mathcal{F}^{t+1}_{\text{BEV}}$:

$$\mathcal{F}^{t+1}_{\text{WM}} = \text{Warp}(\mathcal{F}^{t+1}_{\text{WM}}, T^{t+1}_t), \tag{5.6}$$

where $T^{t+1}_t$ is the transformation from $t$ to $t+1$. Additionally, at $t_0$, the beginning of a sequence, we initialize $\mathcal{F}^{t_0}_{\text{WM}}$ by repeating $\mathcal{F}^{t_0}_{\text{BEV}}$ for $T_{\text{WM}}$ times. After the memory fusion at $t_0$, we replace $\mathcal{F}^{t_0}_{\text{BEV}}$ with $\tilde{\mathcal{F}}^{t_0}_{\text{BEV}}$ and warp them to form $\mathcal{F}^{t_1}_{\text{BEV}}$. The following process of updating working memory follows the strategy mentioned above. Meanwhile, the working memory buffer also serves to store the temporal overlap heatmap. The propagation process for the heatmap is discussed in section 5.3.1.

#### 5.3.2.2 Working Memory Fusion

We propose a working memory fusion module to obtain a unified BEV feature $\tilde{\mathcal{F}}_{\text{BEV}}$ fused with temporal cues. The goal of the memory fusion block is to effectively utilize the temporal information in working memory features, aided by the temporal overlap heatmap, and to fuse them with the current BEV feature to obtain a unified BEV feature with rich temporal information.

Specifically, at time $t$, the input to this module consists of the working memory features $\mathcal{F}^t_{\text{WM}} \in \mathbb{R}^{C_{\text{WM}} \times H \times W}$, the temporal overlap heatmap $\mathbf{H}_t \in \mathbb{R}^{1 \times H \times W}$, and the BEV feature $\mathcal{F}^t_{\text{BEV}} \in \mathbb{R}^{C \times H \times W}$ obtained from the BEV feature encoder. Here $C$ is the feature channel dimension of the BEV feature encoder, $C_{\text{WM}} = T_{\text{WM}} \times C$, and $H$ and $W$ represent the spatial size of the BEV feature. The output of the working memory fusion module is the unified feature $\tilde{\mathcal{F}}^t_{\text{BEV}} \in \mathbb{R}^{C \times H \times W}$. In the working memory fusion module, the first step is to extract

low-level features from the temporal overlap heatmap:

$$\tilde{\mathbf{H}}_t = \text{sigmoid}(\text{Conv}_{\mathbf{H}}(\mathbf{H}_t)) \in \mathbb{R}^{C_{\mathbf{H}} \times H \times W}, \tag{5.7}$$

where $\text{Conv}_{\mathbf{H}}$ is a convolution block composed of three convolution layers with ReLU, and $C_{\mathbf{H}}$ is the feature dimension of the learned temporal overlap heatmap feature. We apply a sigmoid function to the output temporal overlap feature to bound its value. Subsequently, we use a convolutional memory fusion block to obtain the unified BEV feature $\tilde{\mathcal{F}}^t_{\text{BEV}}$ following:

$$\tilde{\mathcal{F}}^t_{\text{BEV}} = \text{LayerNorm}(\text{Conv}_{\text{Mem}}(\text{Concat}(\mathcal{F}^t_{\text{WM}}, \tilde{\mathbf{H}}_t, \mathcal{F}^t_{\text{BEV}}))), \tag{5.8}$$

where $\text{Conv}_{\text{Mem}}$ is composed of three convolution layers with ReLU. We follow [210] to apply layer normalization to improve training stability.

When designing the working memory fusion module, we keep the HD map construction task in mind. We observe that a unique feature for this task is that map elements, such as lane markings, can have elongated shapes, which inspires us to enhance the receptive field of the memory fusion module to let temporal features interact more effectively both temporally and spatially. Therefore, we apply dilated (atrous) convolution [17] in $\text{Conv}_{\text{Mem}}$ to further improve the working memory fusion module.

### 5.3.3 Training Loss

During training, the overall map loss $\mathcal{L}_{map}$ is defined as in [210]:

$$\mathcal{L}_{map} = \lambda_1 \mathcal{L}_{Focal} + \lambda_2 \mathcal{L}_{line} + \lambda_3 \mathcal{L}_{trans}, \tag{5.9}$$

where $\mathcal{L}_{Focal}$ is the classification matching cost, $\mathcal{L}_{line}$ is the polyline-wise matching cost, $\mathcal{L}_{trans}$ is the auxiliary transformation loss for temporal query propagation, and $\lambda_1$-$\lambda_3$ are pre-defined loss weights. We refer to [210] for more details about the loss function.

## 5.4 Experiments

### 5.4.1 Datasets and Splits

We evaluate our methods on the open-source nuScenes [8] and Argoverse2 [185] benchmarks. The nuScenes benchmark has an annotation keyframe rate of 2 Hz, with 6 synchronized cameras. Argoverse2 is annotated at 10 Hz with 7 ring cameras and 2 stereo cameras. We

Table 5.1: Performance comparison with baseline methods on the new nuScenes split under two bird's-eye-view perception ranges. AP and mAP are reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric within each perception range is **bolded**. Abbreviations: pedestrian crossing (ped), lane divider (div), and road boundary (bound). R50 denotes a ResNet-50 backbone. For StreamMapNet [210], we report our reproduced results, which match the results reported in the original paper.

| Range | Method | Backbone | Image Size (px) | Epoch | $AP_{ped}$ (%)↑ | $AP_{div}$ (%)↑ | $AP_{bound}$ (%)↑ | mAP (%)↑ |
|---|---|---|---|---|---|---|---|---|
| $60 \times 30\,m$ | VectorMapNet | R50 | $256 \times 480$ | 120 | 15.8 | 17.0 | 21.2 | 18.0 |
| | MapTR | R50 | $480 \times 800$ | 24 | 6.4 | 20.7 | 35.5 | 20.9 |
| | StreamMapNet | R50 | $480 \times 800$ | 24 | 30.7 | 29.9 | 41.8 | 34.1 |
| | MemFusionMap (ours) | R50 | $480 \times 800$ | 24 | **38.3** | **32.1** | **43.6** | **38.0** |
| $100 \times 50\,m$ | VectorMapNet | R50 | $256 \times 480$ | 120 | 12.0 | 8.1 | 6.3 | 8.8 |
| | MapTR | R50 | $480 \times 800$ | 24 | 8.3 | 16.0 | 20.0 | 14.8 |
| | StreamMapNet | R50 | $480 \times 800$ | 24 | 25.4 | 19.0 | 22.2 | 22.2 |
| | MemFusionMap (ours) | R50 | $480 \times 800$ | 24 | **30.2** | **22.6** | **29.9** | **27.6** |

follow [210] to unify the frame rate of the Argoverse2 dataset to 2 Hz, and use all 6 cameras of nuScenes and the 7 ring cameras of Argoverse2 as the input to the model.

As noted by [146, 210], the default splits in both nuScenes and Argoverse2 can be prone to overfitting as there is a large portion of overlap between training and validation sets. We highlight that the ultimate goal of the online HD map construction task is to develop a generalizable model that can adapt to unseen environments with robustness to potential map changes, instead of simply memorizing the training set. Therefore, we focus on evaluating MemFusionMap using the split proposed in [146] for nuScenes and the split proposed in [210] for Argoverse2. We refer to these non-overlapped splits as new splits and the default splits as original splits in the following sub-sections.

### 5.4.2 Evaluation Metrics

We follow [210, 172] to consider both a small perceptual range ($60 \times 30\,m$) and a large perceptual range ($100 \times 50\,m$). We mainly use the geofenced new splits proposed by [146, 210]. We use Average Precision (AP) as the evaluation metric. We calculate AP using distance thresholds of $\{0.5\,m, 1.0\,m, 1.5\,m\}$ for the small perceptual range and $\{1.0\,m, 1.5\,m, 2.0\,m\}$ for the large perceptual range. Following common practices in existing works, we evaluate three types of map elements: pedestrian crossings, lane dividers, and road boundaries. We calculate the mean AP (mAP) score of the three types of map elements.

### 5.4.3 Implementation Details

We build MemFusionMap upon the codebase of StreamMapNet [210]. We use ResNet-50 [47] as the image feature encoder. We use BEVFormer [93] as the BEV feature encoder to project

Table 5.2: Performance comparison with baseline methods on the new Argoverse2 split under two bird's-eye-view perception ranges. AP and mAP are reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric within each perception range is bolded. Abbreviations: pedestrian crossing (ped), lane divider (div), and road boundary (bound). R50 denotes a ResNet-50 backbone. For StreamMapNet [210], we report our reproduced results, which match the results reported in the original paper.

| Range | Method | Backbone | Image Size (px) | Epoch | $\text{AP}_{ped}$ (%)↑ | $\text{AP}_{div}$ (%)↑ | $\text{AP}_{bound}$ (%)↑ | mAP (%)↑ |
|---|---|---|---|---|---|---|---|---|
| $60 \times 30\,m$ | VectorMapNet | R50 | $384 \times 384$ | 120 | 35.6 | 34.9 | 37.8 | 36.1 |
| | MapTR | R50 | $608 \times 608$ | 30 | 48.1 | 50.4 | 55.0 | 51.1 |
| | StreamMapNet | R50 | $608 \times 608$ | 30 | 57.5 | 55.9 | 61.5 | 58.3 |
| | MemFusionMap (ours) | R50 | $608 \times 608$ | 30 | **59.3** | **57.2** | **65.1** | **60.6** |
| $100 \times 50\,m$ | VectorMapNet | R50 | $384 \times 384$ | 120 | 32.4 | 20.6 | 24.3 | 25.7 |
| | MapTR | R50 | $608 \times 608$ | 30 | 46.3 | 36.3 | 38.0 | 40.2 |
| | StreamMapNet | R50 | $608 \times 608$ | 30 | 60.2 | 45.5 | 49.0 | 51.5 |
| | MemFusionMap (ours) | R50 | $608 \times 608$ | 30 | **64.1** | **47.3** | **51.5** | **54.3** |

features from 2D to BEV space. $C$ is set as 256 across all experiments. We set $H$ as 50 and $W$ as 100. In the memory fusion module, we set $T_{\text{WM}}$ as 4 and $C_{\mathbf{H}}$ as 32. The model is trained on 8 NVIDIA V100 GPUs with a total batch size of 32. The learning rate is set to $5 \times 10^{-4}$, and an AdamW [109] optimizer is used. To train MemFusionMap, we apply the streaming training strategy [171, 43] to detach the memory tensor so gradients do not propagate back to previous frames. Following [133, 210], we apply a two-stage training procedure, where the input of the first stage is single-frame. It is followed by a temporal training stage, where the input is consecutive sequences obtained by dividing the raw training sequence into 2 random sequences.

We follow StreamMapNet [210] for the overall map loss:

$$\mathcal{L}_{map} = \lambda_1 \mathcal{L}_{Focal} + \lambda_2 \mathcal{L}_{line} + \lambda_3 \mathcal{L}_{trans}, \tag{5.10}$$

where $\mathcal{L}_{Focal}$ is the classification matching cost, $\mathcal{L}_{line}$ is the polyline-wise matching cost, and $\mathcal{L}_{trans}$ is the auxiliary transformation loss for temporal query propagation. We set $\lambda_1 = 5.0$, $\lambda_2 = 50.0$, and $\lambda_3 = 0.1$ for all experiments. To ensure fair comparison with StreamMapNet, we also follow its public training schedule: on nuScenes and Argoverse2, the single-frame warm-up stage lasts 4 and 5 epochs, respectively, before switching to the temporal training stage.

For the temporal overlap heatmap branch, we use a three-layer convolution block $\text{Conv}_{\mathbf{H}}$ with channel dimensions $1 \to 16 \to 16 \to 32$. The first two layers use $3 \times 3$ kernels with ReLU activation, and the final layer uses a $1 \times 1$ kernel to project the feature to $C_{\mathbf{H}} = 32$. In the working memory fusion module, the convolutional fusion block $\text{Conv}_{\text{Mem}}$ contains three $3 \times 3$ convolution layers with dilation $(2, 2)$ and channel dimensions $1312 \to 256 \to 256 \to 256$, followed by LayerNorm. The input channel size 1312 is computed as $T_{\text{WM}} \times C + C_{\mathbf{H}} + C =$

$4 \times 256 + 32 + 256$.

### 5.4.4 Quantitative Results

We compare MemFusionMap against state-of-the-art baselines across the nuScenes [8] and Argoverse2 [185] datasets with two perceptual ranges using the new splits [146, 210]. The baselines in the comprehensive comparison include StreamMapNet [210] and single-frame baselines, including VectorMapNet [104] and MapTR [97]. For the follow-up works of StreamMapNet (i.e., SQD-MapNet [172] and MapTracker [15]), we conduct separate comparisons. This is due to either the lack of support for the extended $100 \times 50\,m$ perception range in [15], which holds greater practical value for AV deployment, or the absence of open-source implementation and results evaluated using the new splits in [172]. We report the number of epochs to demonstrate the training efficiency and measure the runtime in frames per second (FPS) on a desktop equipped with an NVIDIA A6000 GPU.

We first show an overall comparison on the new split [146] of the nuScenes [8] benchmark in Table 5.1. At both ranges, MemFusionMap achieves significant improvement over baselines across all metrics, demonstrating the effectiveness of the proposed memory fusion architecture. Similarly, as shown in Table 5.2, MemFusionMap demonstrates superior performance over other methods across all categories on the Argoverse2 benchmark, validating the consistent improvement brought by MemFusionMap. We also highlight the greater improvement in the long perception range setting compared to the short range setting on both benchmarks, which is more applicable to real-world deployment. As MemFusionMap is built upon StreamMapNet and shares a similar training objective, we emphasize the advancement brought by MemFusionMap over StreamMapNet, which directly shows the improvement brought by the proposed working memory fusion framework over the previous GRU-based temporal fusion paradigm.

Furthermore, we show a comparison with open-source temporal baselines on mAP and runtime in Table 5.3. Though MemFusionMap significantly outperforms StreamMapNet [210] while maintaining a similarly fast runtime, it falls behind the recent state-of-the-art work MapTracker [15] in mAP. We recognize the contribution of MapTracker in formulating the HD map construction task as a tracking problem, and think merging MemFusionMap and MapTracker is an interesting future direction. However, it is worth mentioning that MapTracker benefits from additional ground truth of tracked map elements and requires time-consuming post-processing. This may have scalability and compatibility issues with large-scale databases that use an autolabeling process that cannot guarantee all frames are quality-assured, which could render a direct comparison on mAP inappropriate. Addi-

Table 5.3: Quantitative comparison of map prediction accuracy and runtime on the new nuScenes split at the $30\,m$ perception range. mAP is reported in percentage points (%), FPS is reported in frames per second, and ↑ indicates that higher is better. The best result for each metric is **bolded**. MapTracker [15] lacks support for the extended $50\,m$ range setting. SQD-MapNet [172] is not included because it is not open-source.

| Method | Epoch | mAP (%)↑ | FPS↑ |
|---|---|---|---|
| StreamMapNet [210] | 24 | 34.1 | **15.3** |
| MapTracker [15] | 72 | **40.3** | 11.6 |
| MemFusionMap (ours) | 24 | 38.0 | 15.1 |

Table 5.4: Performance comparison on the original splits of nuScenes and Argoverse2 at the $30\,m$ perception range. mAP is reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric within each dataset is **bolded**. * indicates that we follow the StreamMapNet code repository and train for 30 epochs on the original split. † indicates results reported in the SQD-MapNet [172] paper.

| Dataset | Method | Epoch | mAP (%)↑ |
|---|---|---|---|
| nuScenes | StreamMapNet [210] | 30* | 63.4 |
| | SQD-MapNet† [172] | 24 | 65.0 |
| | MemFusionMap (ours) | 30* | **65.2** |
| Argoverse2 | StreamMapNet† [210] | 30 | 61.5 |
| | SQD-MapNet† [172] | 30 | 63.3 |
| | MemFusionMap (ours) | 30 | **63.8** |

tionally, as shown in Table 5.3, MemFusionMap demonstrates strong performance without requiring a long training schedule, which is very valuable for integrating into large-scale databases. Compared with MapTracker [15], MemFusionMap also improves runtime significantly. Considering these factors, we still think MemFusionMap has unique benefits for the community.

We also show additional comparison on the original split (Table 5.4) for completeness since SQD-MapNet [172] does not report results on the new split and is not open-source. The evaluation is based on the $60 \times 30\,m$ range since SQD-MapNet uses a larger BEV spatial dimension for the extended range. Results in Table 5.4 demonstrate that MemFusionMap still outperforms the baselines, though we note that this split is prone to overfitting.

We additionally analyze convergence because rapid training is important for scaling online HD map construction to large databases. As shown in Figure 5.6, MemFusionMap reaches a strong performance level using fewer training epochs than the baseline, which we attribute to the improved temporal reasoning enabled by working memory fusion and the temporal overlap heatmap.

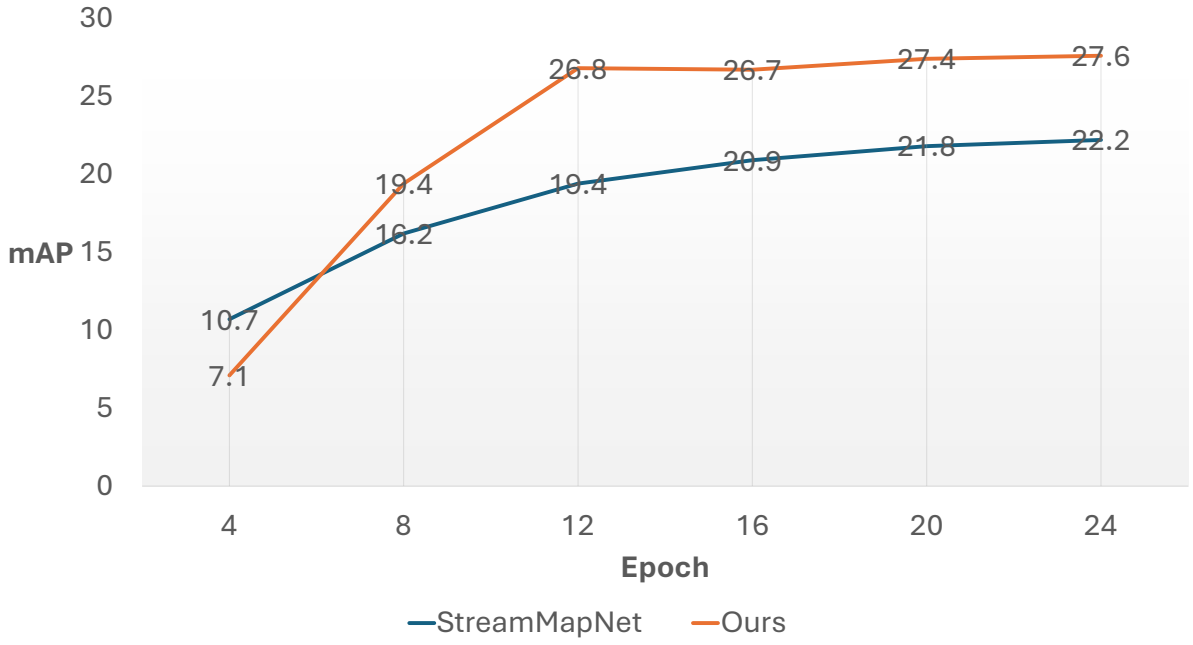



Figure 5.6: Convergence comparison on the new nuScenes split at the extended $50\,m$ range.

### 5.4.5 Ablation Studies

We conduct extensive ablation studies to further break down the performance improvement brought by MemFusionMap and justify our design choice. We conduct all of the ablation studies using the new split of nuScenes at $100 \times 50\,m$ range.

We show an ablation study in Table 5.5. We first remove the GRU-based temporal BEV fusion from StreamMapNet. We notice a notable performance drop, which shows the importance of temporal fusion for this task. We then test adding a convolutional working memory fusion module. It is observed that a simple convolution block can effectively improve temporal reasoning capability with the help of increased memory capacity by maintaining working memory features. The comparison between (a) and (c) also demonstrates the challenge of accumulating the entire history into a single memory latent. The improvement from (c) to (d) validates the importance of using dilated convolution for larger receptive field. Progressing from (d) to (e), the 1.7% improvement on an already strong model justifies the significance of our novel design of the temporal overlap heatmap.

Furthermore, we study the effect of varying the working memory capacity $T_{\text{WM}}$. As shown in Table 5.6, we find that $T_{\text{WM}} = 4$ is a sweet spot that delivers the best performance while also ensuring good runtime performance. We conjecture that the performance drop from further increasing $T_{\text{WM}}$ may be due to the unnecessary dilution of the current BEV feature

Table 5.5: Ablation study of the proposed components in MemFusionMap evaluated on the new nuScenes split at the $50\,m$ perception range. Starting from the StreamMapNet baseline in row (a), row (b) removes the GRU-based BEV temporal fusion module to evaluate the importance of temporal fusion. Row (c) adds the proposed convolutional working memory fusion module, row (d) further introduces dilated convolution to enlarge the receptive field, and row (e) adds the temporal overlap heatmap to guide memory aggregation. mAP is reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric is **bolded**.

| Index | Method | mAP (%)↑ |
|---|---|---|
| (a) | StreamMapNet | 22.2 |
| (b) | − GRU BEV fusion | 19.5 |
| (c) | + Working Memory Fusion | 25.5 |
| (d) | + Dilated convolution | 25.9 |
| (e) | + Temporal overlap heatmap | **27.6** |
| | MemFusionMap | **27.6** |

Table 5.6: Ablation study of the working memory capacity in MemFusionMap. $T_{\text{WM}}$ denotes the number of historical BEV feature frames maintained in working memory. mAP is reported in percentage points (%), FPS is reported in frames per second, and ↑ indicates that higher is better. The best result for each metric is **bolded**.

| Module | Memory Capacity $T_{\text{WM}}$ | mAP (%)↑ | FPS↑ |
|---|---|---|---|
| Memory Fusion | 1 | 24.3 | **15.4** |
| | 2 | 25.9 | 15.2 |
| | 4 | **27.6** | 15.1 |
| | 6 | 26.2 | 14.8 |
| | 8 | 27.0 | 14.7 |

when fusing with additional temporal information.

Additionally, we validate the design choice of dilation parameters of the convolution layers in the proposed working memory fusion module. Empirical results in Table 5.7 demonstrate that maintaining a symmetric dilation of $(2, 2)$ brings the most improvement.

We also study the non-linearity used to extract the temporal overlap heatmap feature. As shown in Table 5.8, using sigmoid performs better than ReLU. We conjecture that bounding the overlap feature to $[0, 1]$ improves its adaptability to the evaluation sequences, where each scene lasts the full 20s while the temporal training sequences are randomly split.

Table 5.7: Ablation study of the convolution dilation choice in the working memory fusion module. The convolution dilation is denoted as $(d_H, d_W)$, where $d_H$ and $d_W$ are the dilation factors along the height and width dimensions of the BEV feature map, respectively. mAP is reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric is **bolded**.

| Module | Conv Dilation $(d_H, d_W)$ | mAP (%)↑ |
|---|---|---|
| Memory Fusion | (1,1) | 26.4 |
| | (2,2) | **27.6** |
| | (2,3) | 26.4 |
| | (3,3) | 27.0 |

Table 5.8: Ablation study of the non-linear activation function used in the temporal overlap heatmap feature branch on the new nuScenes split at the $100 \times 50\,m$ BEV perception range. The compared activation functions are applied after the temporal overlap heatmap feature branch to produce the heatmap-based feature representation. mAP is reported in percentage points (%), and ↑ indicates that higher is better. The best result for each metric is **bolded**.

| Module | Non-linearity | mAP (%)↑ |
|---|---|---|
| Temporal Overlap Heatmap Feature | ReLU | 26.5 |
| | Sigmoid | **27.6** |

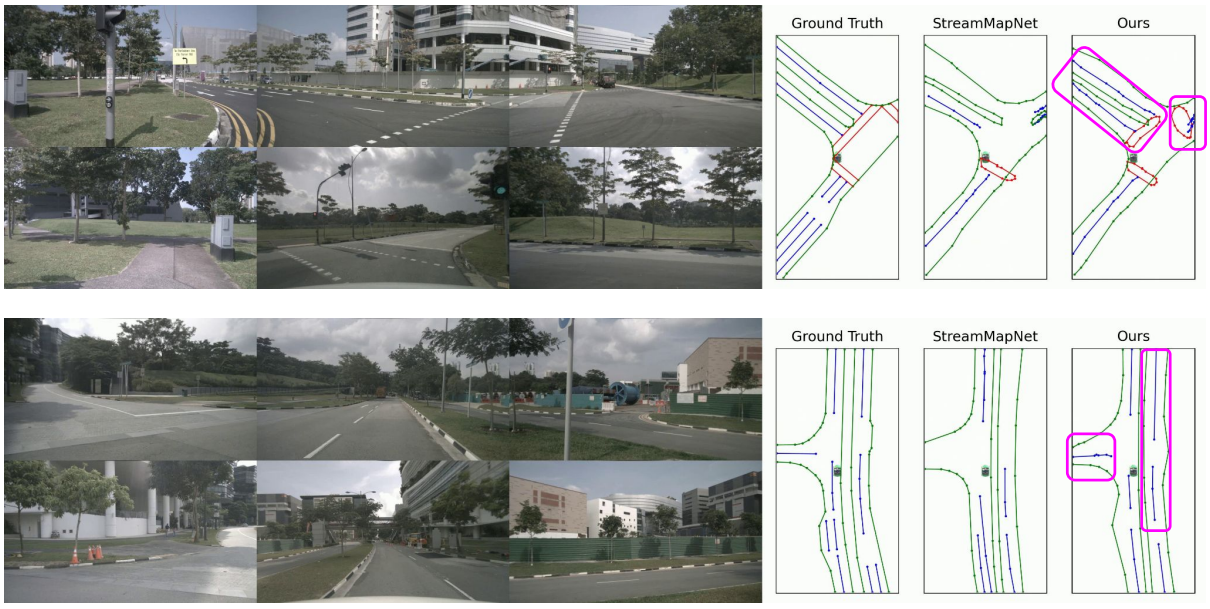



Figure 5.7: We show qualitative results on nuScenes [8] at the extended $50\,m$ range. We show the input multi-view images on the left. In the HD maps, green denotes road boundaries, red denotes pedestrian crossings, and blue denotes lane lines. Best viewed in color.

### 5.4.6 Qualitative Results

We present qualitative comparison of MemFusionMap and StreamMapNet on nuScenes [8] at the extended $100 \times 50\,m$ range, which is the most challenging configuration. We present several examples of MemFusionMap outperforming the baseline model, StreamMapNet [210], in Figure 5.7. Thanks to the proposed working memory fusion module with the temporal overlap heatmap, MemFusionMap predicts more accurate map elements with better geometry and fewer missed elements.

We further show two sequential examples in Figures 5.8 and 5.9 to illustrate how the improved memory capacity benefits temporal reasoning. In Figure 5.8, StreamMapNet starts to generate false predictions in the highlighted region at frames $t-1$ and $t$ after a bus passes through the scene and heavily occludes the relevant map element. Although the highlighted region is visible in earlier history frames, the accumulated temporal errors corrupt the single memory latent of the baseline. In contrast, MemFusionMap maintains access to multiple recent working-memory features and can still leverage the clearer historical observations to avoid these false positives.

We observe a similar pattern in Figure 5.9. A pedestrian crossing is persistently occluded by moving vehicles and is only intermittently visible across the sequence. This type of scene is particularly challenging for temporal fusion. MemFusionMap is able to preserve the pedestrian crossing consistently across frames, while StreamMapNet misses it in the final frames. These examples further validate that the proposed working memory fusion module improves robustness to long-duration occlusion and reduces error accumulation over time.

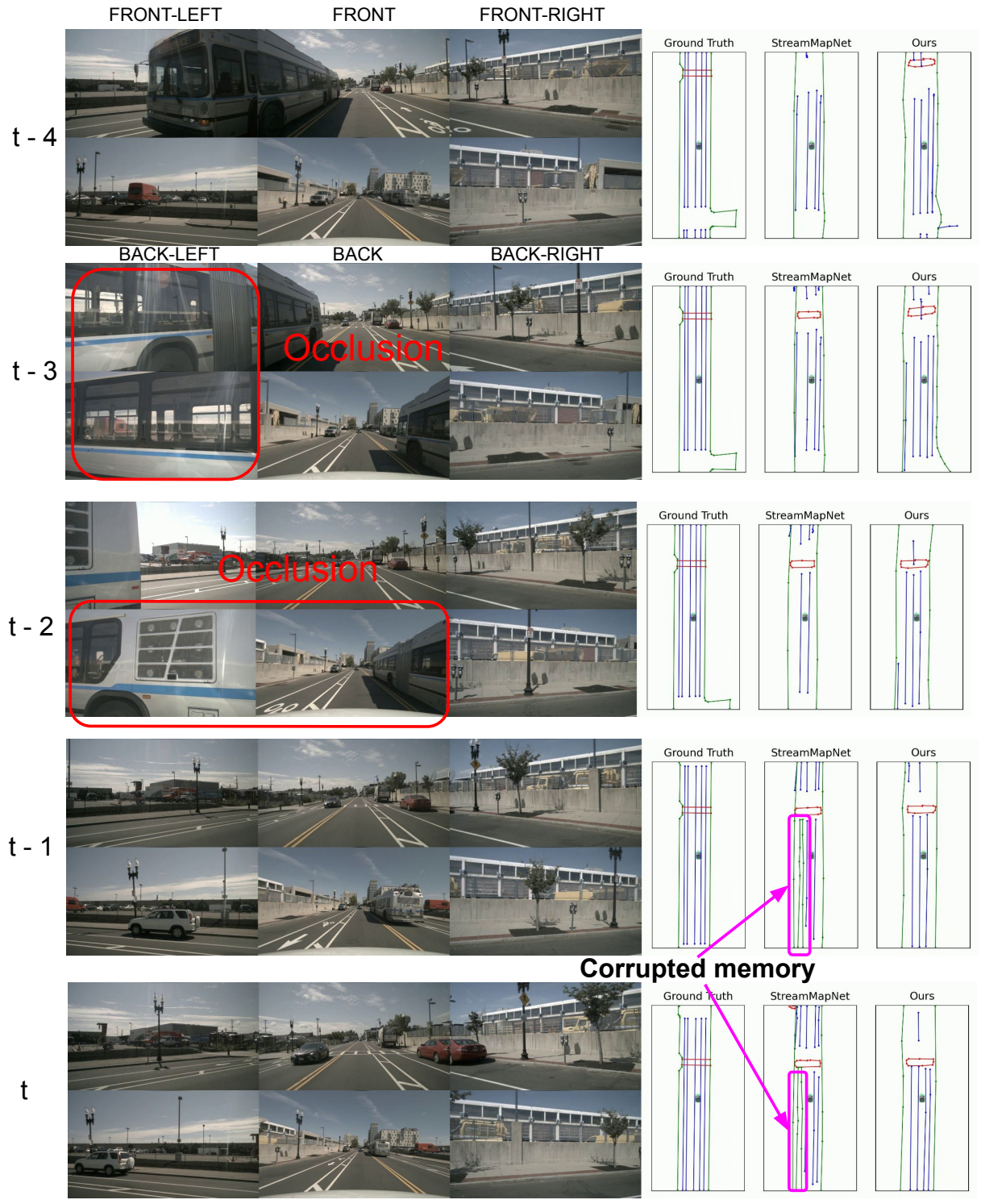



Figure 5.8: Sequential qualitative results on nuScenes at the extended $100 \times 50\,m$ range. We show a consecutive sequence of five frames. The highlighted region shows a case where MemFusionMap avoids false predictions that accumulate in the baseline under severe temporal occlusion. In the HD maps, green denotes road boundaries, red denotes pedestrian crossings, and blue denotes lane lines. Best viewed in color.

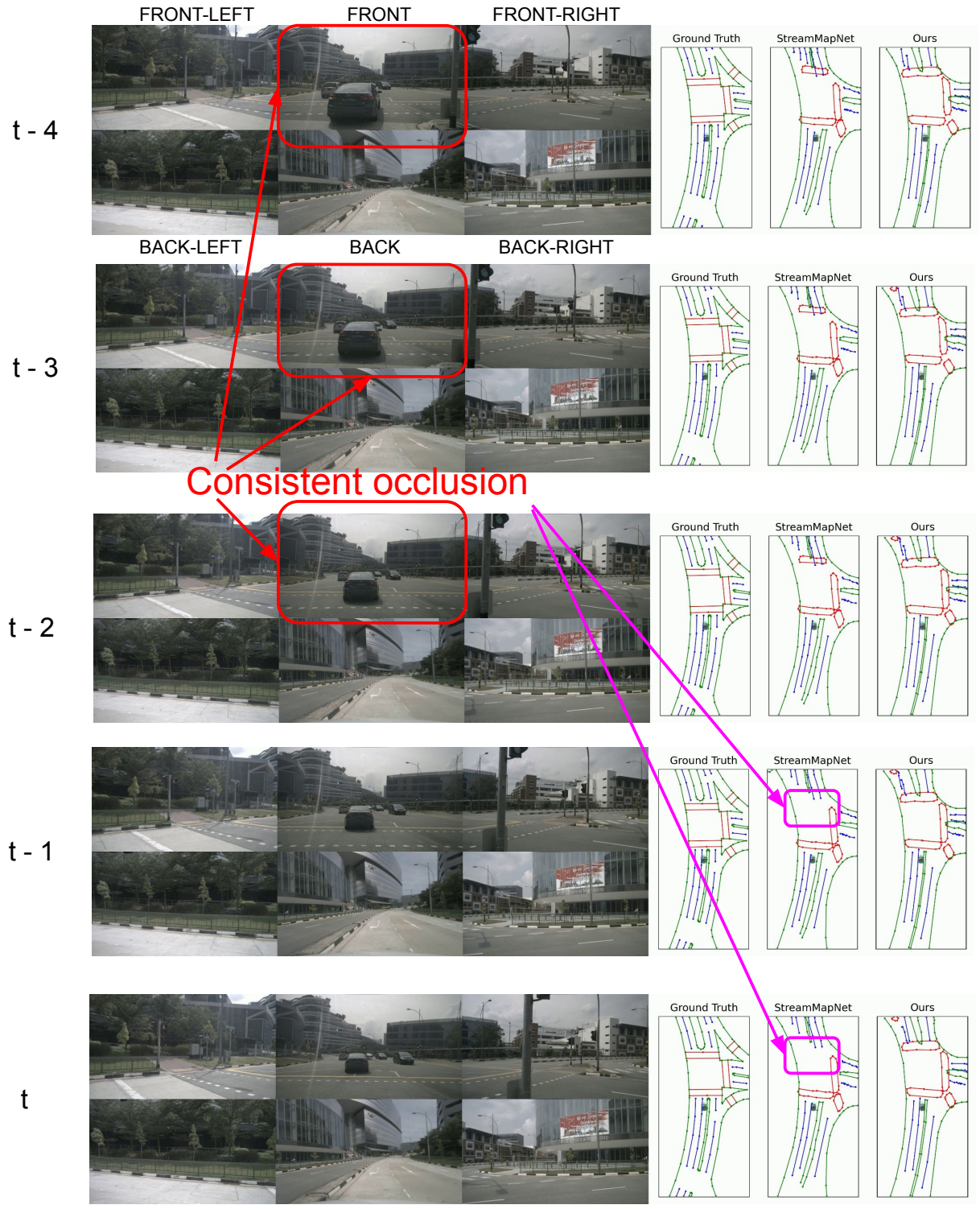



Figure 5.9: Sequential qualitative results on nuScenes at the extended $100 \times 50\,m$ range. We show a consecutive sequence of five frames. The highlighted region shows a case where MemFusionMap consistently recovers a pedestrian crossing under long-lasting occlusion. In the HD maps, green denotes road boundaries, red denotes pedestrian crossings, and blue denotes lane lines. Best viewed in color.

## 5.5 Conclusion

In this chapter, we have proposed MemFusionMap, a novel approach for effective online vectorized HD map construction with enhanced temporal reasoning capability. We propose a working memory fusion module to improve the memory capacity of the model. We also design a temporal overlap heatmap to aid the temporal reasoning process across history frames. Our proposed MemFusionMap demonstrates state-of-the-art performance on open-source benchmarks with the geofenced splits and maintains a fast runtime and versatile design. We hope MemFusionMap will inspire future research to leverage our proposed memory fusion module in other BEV perception tasks. We think integrating MemFusionMap in the MapTracker [15] framework to leverage the tracking paradigm to further enhance the performance and temporal consistency is also worth exploring.

# CHAPTER 6

# A Spatiotemporal Camera–Radar Backbone for Driving via Forecasting-Based Pretraining

## 6.1 Introduction

Autonomous vehicles (AVs) rely on perception systems to understand the surrounding environment and support downstream prediction and planning modules [8, 160, 52, 169, 183]. Common sensing modalities include camera, LiDAR, and radar, each providing distinct advantages and limitations [230, 223]. Cameras offer dense semantic cues, LiDAR provides accurate 3D geometry, and radar is robust to adverse weather and lighting while also measuring motion through Doppler velocity. Since no single sensor is sufficient across all conditions, multimodal fusion has received significant interest from the research community [176, 230].

Among different sensor configurations, LiDAR-camera (LC) fusion has achieved strong performance across many perception benchmarks [40, 160, 8, 87, 107, 18]. However, the cost and form factor of LiDAR can limit large-scale deployment in production vehicles [18, 223]. In contrast, camera-radar (CR) sensing offers a more scalable and lower-cost configuration that is already accessible on many vehicles with advanced driver assistance systems [155, 223]. As a result, CR fusion has attracted increasing attention in recent years, with demonstrated benefits for 3D object detection, scene segmentation, mapping, and occupancy prediction [121, 113, 74, 150, 110]. These advances suggest that CR fusion is a promising path toward practical and robust autonomous driving.

Despite this progress, including recent CR fusion networks, many AV perception models [135, 93, 107] still rely heavily on supervised training with precise 3D task annotations. Such annotations are expensive to obtain at scale, which motivates pretraining as a more scalable way to learn effective representations for downstream driving tasks from large amounts of unlabeled or weakly labeled data [200, 201, 203, 173]. Here, pretraining denotes an initial representation-learning stage performed before task-specific optimization: rather than training a model from scratch only with downstream labels, the backbone is first optimized

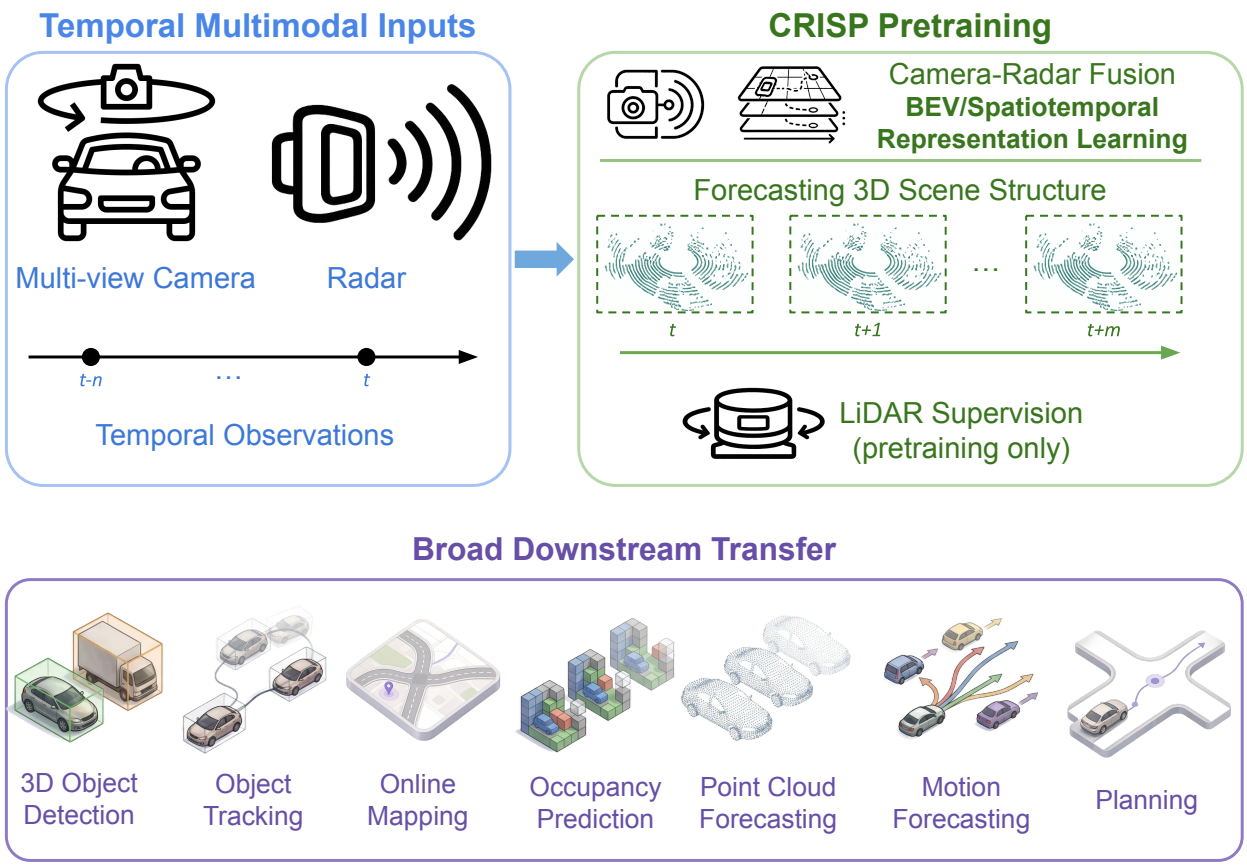



Figure 6.1: Overview of the motivation of CRISP. We use low-cost camera-radar inputs for inference, while leveraging LiDAR as privileged supervision during pretraining to learn a transferable multimodal driving backbone.

with a scalable objective and then finetuned by attaching downstream heads and updating the model with task-specific supervision [173]. Early autonomous driving pretraining methods mainly focus on reconstructing masked LiDAR points, voxels, or bird's-eye-view (BEV) features [48, 200, 102], whereas more recent multimodal methods exploit LiDAR as privileged supervision for camera-based or LC pretraining [132, 13, 201]. At the same time, forecasting-based pretraining and world-model learning have emerged as effective ways to encode scene dynamics for driving [67, 117, 118, 203, 173]. However, these lines of work are still dominated by camera-only (CO) or LC settings, leaving radar largely unexplored in transferable representation learning.

This gap is particularly important because radar is naturally suited for predictive scene modeling. Cameras provide dense semantic cues, but future motion must be inferred indirectly from appearance, geometry, and temporal changes, and can be affected by depth ambiguity, lighting, and visibility. Radar, in contrast, provides metric range and Doppler velocity

measurements, offering a direct physical cue about object motion. These measurements can anchor future scene prediction by complementing visual semantics with instantaneous motion evidence. Despite this suitability for forecasting, existing CR methods are still mostly trained for supervised downstream heads such as detection, segmentation, mapping, or occupancy prediction [121, 74, 71, 150, 110, 223]. Thus, the synergy between forecasting-based pretraining and CR fusion remains underexplored.

To this end, we propose **CRISP**, which is summarized in Figure 6.1. CRISP is designed to capture multimodal geometry, radar motion cues, temporal dynamics, and BEV semantics in a unified backbone, while preserving the practical advantage of LiDAR-free inference. Prior forecasting-based pretraining methods model temporal scene evolution effectively, but primarily in CO or LC settings; prior CR methods exploit low-cost multimodal sensing effectively, but usually through task-specific supervised training. Unlike cross-modal alignment or distillation methods that mainly use another modality to regularize same-frame feature learning [50, 6, 87], CRISP uses future LiDAR point clouds as privileged forecasting supervision. This encourages the CR backbone to learn not only same-frame geometry, but also multimodal interactions and temporal scene dynamics. LiDAR is used only during pretraining, while the learned backbone remains a practical CR system at inference time. By bridging camera, radar, and LiDAR during pretraining, CRISP provides a scalable representation learning framework to facilitate downstream perception, prediction, and planning tasks.

The main contributions of this chapter are as follows:

- We propose CRISP, a forecasting-based CR pretraining framework that learns transferable spatiotemporal BEV representations by predicting future LiDAR point clouds from historical CR observations. LiDAR is used only as privileged supervision during pretraining, while inference remains CR only.

- We design a CR spatiotemporal BEV backbone that integrates radar into representation learning through an enhanced radar encoder, radar-enhanced temporal self-attention, and multimodal feature rendering with modality innovation gating. These modules allow radar range and Doppler cues to condition temporal propagation, while enabling each BEV token to selectively compose dense visual semantics with sparse but motion-sensitive radar evidence through gated residual updates.

- We demonstrate on nuScenes [8] that CRISP improves long-horizon point cloud forecasting and transfers broadly to downstream autonomous driving tasks, including 3D object detection, tracking, online mapping, motion forecasting, future occupancy prediction, and planning, outperforming CO and CR baselines.

## 6.2 Related Work

### 6.2.1 Pretraining for Autonomous Driving

Pretraining has been extensively studied in computer vision and 3D representation learning, and has shown strong transferability across many downstream tasks [46, 209, 130]. Motivated by these advances, recent autonomous driving studies have developed pretraining objectives tailored to outdoor 3D scene understanding [95, 48, 200, 102, 173, 213]. These methods typically learn representations by reconstructing masked points, voxels, or BEV features, and have demonstrated that large-scale pretraining can improve downstream 3D perception [48, 200, 102]. However, most prior works focus on static reconstruction within the current frame, and are therefore less well-suited for learning temporal scene evolution and multimodal interactions jointly.

Multimodal pretraining for autonomous driving has also begun to attract attention [173]. PiMAE [13] studies interactive masked autoencoding on images and point clouds, while DD3D [132] and UniPAD [201] exploit paired LiDAR point clouds to supervise camera-based or multimodal networks through depth prediction or masked scene rendering. These methods demonstrate the value of privileged 3D supervision during pretraining. In parallel, radar-specific pretraining has been explored through contrastive objectives on unlabeled radar data paired with camera observations [44]. Nevertheless, most prior studies either pretrain a single modality or pretrain modalities separately before fusion [201], which limits the opportunity to learn fused spatiotemporal representations.

Recent work suggests that forecasting future 3D scene structure provides a stronger pretraining signal for autonomous driving than static reconstruction alone. Point cloud forecasting has been used as a proxy for 4D occupancy forecasting [67], while UniWorld [117] and DriveWorld [118] formulate world-model-style pretraining through future occupancy prediction in spatiotemporal scenes. ViDAR [203] further shows that forecasting future LiDAR point clouds from historical camera observations can improve a broad range of downstream tasks. More recently, LRS4Fusion [127] extends forecasting-based pretraining to sparse LC fusion, demonstrating the promise of pretraining a multimodal backbone for long-range perception.

Despite these advances, existing forecasting-based pretraining methods are still dominated by CO or LC settings. This leaves radar largely unused in transferable representation learning, even though radar provides direct range and Doppler motion measurements that are especially relevant for predicting scene dynamics. LC pretraining benefits from strong metric geometry, but it does not exploit the low-cost motion sensing available in CR systems. In contrast, CRISP targets joint CR pretraining: the backbone learns fused multimodal repre-

sentations from historical camera and radar observations by forecasting future LiDAR point clouds. This formulation uses LiDAR only as privileged supervision during pretraining, while encouraging the deployed CR backbone to encode cross-modal geometry, radar motion cues, and temporal scene evolution in a unified representation.

### 6.2.2 Camera-Radar Fusion for Autonomous Driving

CR fusion has gained increasing attention in autonomous driving because the two modalities provide complementary sensing capabilities and improve robustness under adverse weather and lighting conditions [155, 181, 205]. Early CR methods primarily focus on 3D object detection by associating sparse radar measurements with image features. CenterFusion [121] detects image-space object centers and links them with radar points through frustum association to improve 3D detection and velocity estimation. CramNet [57] further addresses the geometric ambiguity between camera rays and radar returns through ray-constrained cross-attention, and MVFusion [190] strengthens cross-modal interactions through semantic alignment and radar-guided transformer fusion. These methods demonstrate the value of radar cues, but they are mostly optimized for detection-specific supervision rather than for learning a general multimodal backbone.

More recently, BEV representations have become a unified spatial framework for integrating camera, LiDAR, and radar features [135, 93, 107, 51]. Benefiting from the BEV formulation, CR fusion has expanded from object detection to broader driving tasks. RCBEVDet [101] and CRN [74] learn stronger CR BEV features for 3D perception, while BEV-Guided Fusion [113] and BEVCar [150] demonstrate that CR fusion is also beneficial for scene segmentation, mapping, and velocity-aware BEV understanding. In addition, CRT-Fusion [71] highlights the importance of temporal cues for CR-based 3D detection, LiCROcc [110] shows that radar can contribute to dense semantic occupancy prediction when guided by LiDAR and camera information, and CRKD [223] illustrates the effectiveness of using targeted sensing configurations to improve CR perception through cross-modality knowledge distillation. Most recently, SpaRC-AD [189] extends CR fusion to an end-to-end multi-task driving framework, showing that radar can benefit not only detection, but also tracking, motion prediction, and planning.

Despite the rapid progress of CR fusion, most existing methods are still trained with task-specific labels for detection, segmentation, occupancy prediction, or end-to-end driving tasks [74, 101, 71, 150, 110, 223, 189]. Temporal modeling is typically introduced to improve a particular downstream head, and LiDAR is mainly used through teacher-student supervision or distillation. As a result, it remains unclear how to learn a reusable CR backbone that can

support a wide range of downstream autonomous driving tasks. Our work addresses this gap by using forecasting-based pretraining to learn a unified CR backbone before task-specific finetuning.

### 6.2.3 Forecasting-Based World Models for Driving

Driving world models aim to learn predictive representations of the driving environment: given historical observations, the model should infer the current scene state and anticipate how the 3D scene may evolve in the future. Recent autonomous driving systems increasingly rely on such predictive representations. Temporal BEV encoders such as BEVFormer [93] show that aggregating historical observations is critical for stable 3D scene understanding, while planning-oriented frameworks such as UniAD [52] demonstrate that a shared BEV representation can support perception, motion prediction, and planning in a unified pipeline.

This predictive perspective is closely related to recent driving world models [173]. Point Cloud Forecasting as a Proxy for 4D Occupancy Forecasting [67] shows that future 3D scene evolution can be learned from unlabeled sequences through predictive geometry. UniWorld [117] and DriveWorld [118] formulate autonomous driving pretraining as spatiotemporal world modeling through future occupancy prediction, while ViDAR [203] demonstrates that forecasting future LiDAR point clouds from historical camera observations improves downstream perception, prediction, and planning. More recently, HERMES [229] proposes a unified driving world model for simultaneous 3D scene understanding and generation, using BEV representations together with large language model (LLM)-based world queries to incorporate contextual world knowledge into future scene prediction. These works suggest that forecasting-based objectives provide a natural training signal for learning representations that encode scene geometry, temporal evolution, and downstream-relevant dynamics.

However, existing predictive world models and forecasting-based pretraining methods mainly focus on visual, LiDAR, or language-augmented inputs, and do not study how radar can be incorporated into a transferable multimodal backbone. CRISP bridges this gap by using future LiDAR point cloud forecasting as the pretraining objective for a CR backbone. The learned representation is therefore encouraged to encode multimodal geometry, radar motion cues, temporal dynamics, and BEV semantics in a form that transfers not only to perception, but also to downstream prediction and planning tasks [52]. This distinguishes CRISP from prior CR fusion networks optimized for supervised task-specific heads, as well as from prior driving world models that do not learn joint CR representations.

## 6.3 Preliminaries

We briefly review the two components that CRISP modifies most directly: BEVFormer [93] for spatiotemporal BEV encoding and ViDAR [203] for forecasting-based pretraining.

### 6.3.1 BEVFormer

BEVFormer [93] is an influential CO framework for learning BEV representations from multi-view images. Rather than explicitly constructing a dense 3D volume, BEVFormer uses a set of grid-shaped BEV queries and transformer attention to aggregate information from surrounding cameras and past frames. Its unified BEV representation was shown to support multiple autonomous driving perception tasks, including 3D object detection and semantic map segmentation, and later became a common BEV encoding backbone for end-to-end driving systems such as UniAD [52].

Concretely, BEVFormer represents the scene with a grid of learnable BEV queries and refines them through stacked transformer layers that alternate temporal self-attention, camera spatial cross-attention (SCA), and feed-forward updates. Given multi-view image features $\{F_t^i\}_{i=1}^{N_{\text{view}}}$ at time $t$, the BEV encoder aggregates historical context from previous BEV memory and current spatial evidence from the cameras into a BEV representation.

#### 6.3.1.1 Camera Spatial Cross-Attention

The key view-transformation step in BEVFormer is SCA, which avoids densely lifting all image features into a 3D volume. For a BEV query $Q_p$ at grid location $p$, BEVFormer associates the query with a vertical pillar, samples $N_{\text{ref}}$ 3D reference points along that pillar, projects these points into each camera view, and attends only to image features from views in which the projected points are valid:

$$\text{SCA}(Q_p, F_t) = \frac{1}{|\mathcal{V}_{\text{hit}}|} \sum_{i \in \mathcal{V}_{\text{hit}}} \text{DA}\big(Q_p, \mathcal{P}_p^i, F_t^i\big)\,, \tag{6.1}$$

where $\text{SCA}(\cdot)$ denotes camera spatial cross-attention, $Q_p$ is the BEV query at location $p$, and $F_t = \{F_t^i\}_{i=1}^{N_{\text{view}}}$ denotes the multi-view image features at time $t$. $\mathcal{P}_p^i$ denotes the image-plane projections, onto camera $i$, of the $N_{\text{ref}}$ 3D reference points sampled along the pillar associated with $Q_p$; $\mathcal{V}_{\text{hit}}$ is the set of camera views where at least one such projected point lies inside the image; and DA denotes deformable attention applied to the projected reference points and the image feature map $F_t^i$. This sparse, geometry-guided attention makes the BEV encoder camera-aware while avoiding the cost of dense image-to-volume lifting.

#### 6.3.1.2 Standard Temporal Self-Attention (TSA)

BEVFormer incorporates temporal context by attending to ego-motion-compensated historical BEV memory. Given current BEV queries $Q_t$ and aligned historical memory $B_{t-1}^{\text{ego}}$, temporal self-attention (TSA) predicts deformable sampling offsets and attention weights from the temporal BEV queue:

$$\begin{aligned} Z_t &= [B_{t-1}^{\text{ego}}, Q_t], \\ \Delta p_t,\, a_t &= f_{\text{tsa}}(Z_t), \\ \tilde{B}_t &= \text{MSDeformAttn}(Z_t, p + \Delta p_t, a_t), \end{aligned} \tag{6.2}$$

where $Z_t = [B_{t-1}^{\text{ego}}, Q_t]$ is the temporal BEV queue formed by concatenating the ego-aligned historical memory and the current query, $\text{MSDeformAttn}(\cdot)$ denotes multi-scale deformable attention, $f_{\text{tsa}}(\cdot)$ predicts temporal-attention parameters, $p$ denotes the BEV reference locations, $\Delta p_t$ denotes the learned sampling offsets, and $a_t$ denotes the deformable attention weights. In the standard formulation, the attended features from the temporal queue are fused by a fixed average, producing the temporally aggregated BEV feature $\bar{B}_t$.

Although ego-motion compensation aligns static scene structure, dynamic objects may still occupy different BEV locations across frames. Standard TSA must therefore infer residual object motion implicitly from camera-derived BEV queries. In CRISP, radar does not need to be injected explicitly into every temporal operation. Instead, radar information is first blended into the query state through gated query initialization, producing radar-aware BEV queries. These queries are then used by the standard temporal attention machinery to predict sampling offsets, attention weights, and temporal aggregation. As a result, the offset prediction and temporal fusion become radar-informed through the query representation itself, while preserving the original TSA interface.

### 6.3.2 ViDAR

ViDAR [203] formulates visual autonomous driving pretraining as future point cloud forecasting: given historical multi-view camera observations, the model predicts future LiDAR point clouds. This objective encourages the visual BEV encoder to learn representations that jointly encode semantics, 3D geometry, and temporal dynamics.

Architecturally, ViDAR builds on BEVFormer [93] by using it as the history encoder. The BEVFormer encoder aggregates multi-frame, multi-view camera features into a spatiotemporal BEV representation, which is then passed through a latent rendering operator and an autoregressive future prediction decoder. The future decoder iteratively predicts point

clouds at future timestamps from the encoded historical representation. CRISP keeps this future prediction decoder and the forecasting objective unchanged, using them as predictive supervision for the learned BEV features.

The main distinction is the treatment of the intermediate BEV representation. ViDAR introduces latent rendering to transform camera-derived BEV features into a 3D geometric latent space before future prediction. This is effective for CO pretraining because the rendering process imposes geometric structure on features inferred from images. However, the operator is built around a ray-projection-oriented prior, which is less suitable for further processing of CR fused features. In CRISP, radar already supplies metric BEV occupancy and motion cues, and these cues should interact with camera evidence during BEV formation rather than after a camera-centric rendering step. Therefore, CRISP preserves ViDAR's predictive decoder but replaces latent rendering with multimodal feature rendering, where camera and radar information are composed inside the shared BEV backbone through modality-aware attention and gated fusion.

## 6.4 Methodology

### 6.4.1 Overview of CRISP

CRISP is a CR spatiotemporal BEV backbone pretrained through future point cloud forecasting. It learns a joint CR representation directly: cameras contribute dense visual semantics, while radar contributes sparse metric range and radial-velocity cues for motion-aware temporal reasoning. Rather than pretraining a CO BEV encoder and adding radar only during downstream finetuning, CRISP moves CR interaction inside the BEV backbone and optimizes the resulting representation with LiDAR-supervised forecasting.

As shown in Figure 6.2, CRISP takes historical multi-view images and radar sweeps as input and encodes them into a unified spatiotemporal BEV feature. The image stream provides dense visual context, while the radar stream provides sparse but metric dynamic cues. CRISP fuses these complementary signals inside the BEV backbone before applying the forecasting objective, so that the pretrained representation is optimized for both semantic scene understanding and motion-aware prediction.

The architecture consists of an image encoder, an enhanced radar encoder, a CR spatiotemporal BEV backbone, and a future prediction decoder used only during pretraining. The image encoder extracts multi-view image features, and the radar encoder converts radar sweeps into BEV-aligned radar features. The shared backbone then produces the unified BEV representation through two radar-aware operations. First, radar-enhanced temporal

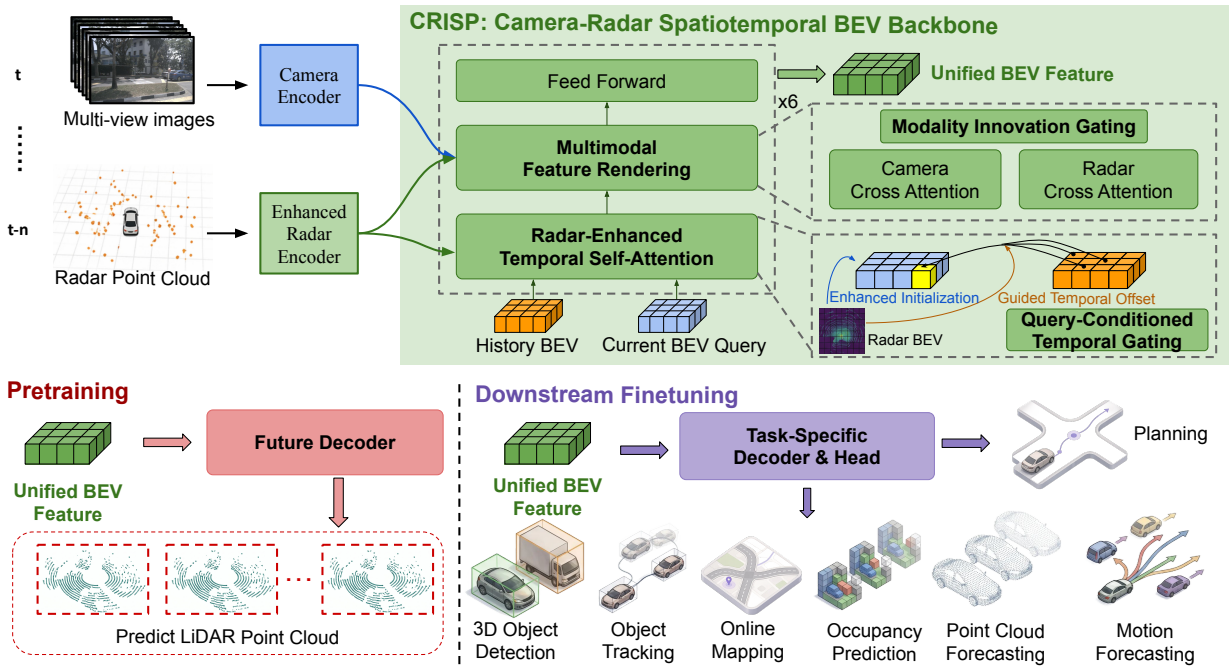



Figure 6.2: Overview of CRISP. Historical multi-view images and aggregated radar observations are encoded into a shared CR BEV backbone. Radar first passes through ego-aware radar encoding, then enters the backbone through radar-enhanced temporal propagation and multimodal feature rendering. During pretraining, a future occupancy decoder is supervised by future LiDAR; during downstream finetuning, only the pretrained CR backbone is retained.

self-attention injects radar information through gated query initialization, making the BEV queries radar-aware before temporal offsets and attention weights are predicted. Second, multimodal feature rendering composes camera and radar evidence through modality-specific cross-attention and adaptive gating.

During pretraining, the unified BEV feature is passed to a future prediction decoder that predicts current and future scene occupancy supervised by LiDAR point clouds. LiDAR serves only as privileged supervision and is not required at inference time. During downstream finetuning, the future decoder is removed and the pretrained CRISP backbone is connected to task-specific heads for perception, prediction, and planning. This design uses forecasting to teach the backbone how to combine visual semantics with radar dynamics, while preserving the benefit of CR-only inference.

### 6.4.2 Enhanced Radar Encoder

Radar returns provide complementary geometric and motion cues for BEV perception, but their measured Doppler velocities are inherently ego-centric and therefore depend on the

instantaneous motion of the sensing vehicle. Directly encoding radar points without this context can make the learned representation ambiguous: similar Doppler patterns may correspond to different object motions under different ego velocities, yaw rates, or accelerations. To address this, we enhance the radar encoder with two lightweight conditioning mechanisms: ego-motion-aware feature modulation and residual spatial calibration. The former injects global vehicle-state information into the radar BEV feature map, while the latter improves feature-level compatibility between radar and camera BEV representations before temporal fusion.

#### 6.4.2.1 Ego-Motion Awareness

Automotive radar measures relative motion with respect to the ego vehicle. Consequently, Doppler-related cues should be interpreted jointly with the current ego state rather than as standalone point attributes. Given the pillar-based radar BEV feature map $F_t^{\text{pill}}$ and the ego-motion information vector $c_t$, we first encode the ego state using a lightweight multilayer perceptron (MLP) and broadcast the resulting embedding across the BEV grid:

$$\tilde{F}_t^{\text{rad}} = F_t^{\text{pill}} + \text{Broadcast}(\text{MLP}(c_t)) \,. \tag{6.3}$$

This additive conditioning can be interpreted as an ego-motion-dependent feature bias that allows the radar branch to adapt its representation according to the vehicle's instantaneous motion. The conditioned feature map $\tilde{F}_t^{\text{rad}}$ is then passed to the subsequent BEV processing stages, enabling temporal radar features to be formed from ego-aware representations.

#### 6.4.2.2 Residual Spatial Calibration

Although radar and camera features are represented in a common BEV coordinate frame, their feature distributions remain substantially different due to differences in sensing geometry, sparsity, noise characteristics, and semantic content. To reduce this modality gap, we introduce a residual spatial calibrator after ego-motion injection. We first project $\tilde{F}_t^{\text{rad}}$ to a lower-dimensional latent space, concatenate it with a BEV positional encoding $P_{\text{bev}}$, and predict a location-aware correction:

$$F_t^{\text{rad}} = \tilde{F}_t^{\text{rad}} + \psi\Big(\text{Proj}\Big(\tilde{F}_t^{\text{rad}}\Big) \oplus P_{\text{bev}}\Big) \,, \tag{6.4}$$

where $\psi(\cdot)$ denotes the calibrator block and $\oplus$ denotes channel-wise concatenation. The residual design preserves the original radar representation while allowing the network to learn spatially varying refinements that improve compatibility with the camera-side BEV

features. We use $F_t^{\text{rad}}$ to denote the final ego-motion-aware and BEV-aligned radar feature used throughout the rest of this chapter.

### 6.4.3 Camera-Radar Spatiotemporal BEV Backbone

CRISP extracts unified spatiotemporal BEV features with a CR BEV encoder. Unlike late-fusion designs that append radar features only after BEV construction, CRISP integrates radar evidence directly inside the BEV encoder, where BEV tokens are temporally propagated and cross-modal evidence is composed. This design is motivated by two observations. First, radar provides geometric and motion-sensitive cues that are useful not only for final detection, but also for deciding how BEV memory should be propagated over time. Second, camera and radar observations have complementary failure modes: camera features are dense and semantically expressive, while radar features are sparse but provide robust range and motion measurements.

CRISP therefore replaces the camera-centric latent rendering stage used in ViDAR with an iterative CR BEV rendering process. Latent rendering is effective for CO pretraining because it encourages image-derived BEV features to acquire geometric structure before future prediction. However, in the CR setting, radar already provides metric BEV observations and motion cues. Passing fused features through a ray-projection-oriented latent rendering operator is therefore less natural than allowing radar and camera evidence to interact directly in the BEV backbone. CRISP preserves ViDAR's future prediction decoder and forecasting objective, but performs multimodal rendering inside the perception encoder.

The encoder contains $L = 6$ repeated layers. Let $Q_t^{(0)} \in \mathbb{R}^{N \times C}$ denote the initial BEV query at timestamp $t$, where $N = H_{\text{bev}} \times W_{\text{bev}}$ and $C$ is the channel dimension. Let $B_{t-1}^{\text{mem}} \in \mathbb{R}^{N \times C}$ denote the ego-motion-aligned BEV memory from the previous frame, let $F_t^{\text{cam}} = \{F_t^i\}_{i=1}^{N_{\text{view}}}$ denote the current multi-view camera features, and let $F_t^{\text{rad}} \in \mathbb{R}^{N \times C}$ denote the calibrated radar BEV feature. Each layer first applies radar-enhanced temporal self-attention, then performs multimodal feature rendering, and finally applies a position-wise feed-forward network (FFN), with residual connections and normalization after each stage. Since every layer has its own attention and fusion parameters, CR fusion is performed iteratively across the backbone. After six layers, the encoder outputs the unified spatiotemporal BEV feature $B_t^{\text{fused}} = Q_t^{(L)}$. We use $B_{t-1}^{\text{mem}}$ for the previous ego-aligned BEV memory consumed by temporal attention, $B_t^{\text{fused}}$ for the current fused BEV state emitted by the backbone, $\mathcal{B}_t$ for the queue of fused BEV states used by the forecasting decoder, and $\hat{B}_{t+\tau}$ for a predicted future BEV state.

A central design choice in CRISP is gated feature update, inspired by recent gated-

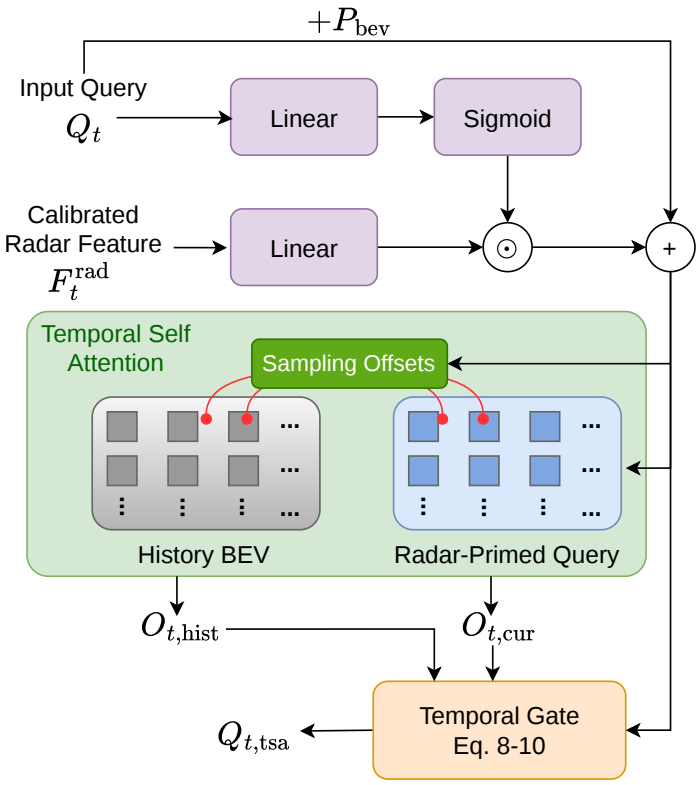



Figure 6.3: Radar-enhanced temporal self-attention. The calibrated radar BEV feature $F_t^{\text{rad}}$ primes the input BEV query $Q_t^{(l-1)}$ through the query-conditioned gate $g_{t,q}^{(l)}$, producing the radar-primed query $\hat{Q}_t^{(l)}$. Ego-aligned temporal deformable attention then attends to the historical BEV memory $B_{t-1}^{\text{mem}}$ and the current query branch. The temporal gate $g_{t,\text{temp}}^{(l)}$ combines current and historical outputs instead of using uniform averaging.

attention designs [143]. Instead of directly adding all attended camera or radar features to the BEV representation, the model predicts gates from the current BEV query and uses them to control each update. We refer to a residual update proposed by an attention branch as an *innovation*: it is new temporal or modality-specific evidence that may modify the current BEV state. In temporal self-attention, the gate balances current-frame features and historical BEV memory. In multimodal feature rendering, the gates control camera-derived and radar-derived innovations.

#### 6.4.3.1 Radar-Enhanced Temporal Self-Attention

As illustrated in Figure 6.3, radar-enhanced temporal self-attention introduces radar before temporal aggregation. The calibrated radar BEV feature first modulates the current BEV query through a learned gate, allowing radar range and motion cues to influence temporal attention. Temporal deformable attention then aggregates features from the ego-aligned

historical BEV memory and the current radar-primed query. Instead of averaging the two temporal branches (i.e., standard temporal self-attention), CRISP predicts a spatial gate that balances current-frame evidence and propagated history.

For the $l$-th layer, let $Q_t^{(l-1)}$ denote the input BEV query. We first project the calibrated radar feature into the BEV query space and predict a query-conditioned radar gate:

$$r_t^{(l)} = \mathrm{Linear}_r^{(l)}\left(F_t^{\mathrm{rad}}\right), \qquad g_{t,q}^{(l)} = \sigma\Big(\mathrm{Linear}_q^{(l)}\Big(Q_t^{(l-1)}\Big)\Big), \tag{6.5}$$

where $r_t^{(l)}$ denotes the projected radar feature used as the residual radar update, $g_{t,q}^{(l)}$ denotes the query-conditioned radar gate, $\sigma(\cdot)$ denotes the sigmoid activation, and $\mathrm{Linear}_r^{(l)}(\cdot)$ and $\mathrm{Linear}_q^{(l)}(\cdot)$ denote layer-specific affine projections. The current BEV query is then radar-primed through a gated residual update:

$$\hat{Q}_t^{(l)} = Q_t^{(l-1)} + g_{t,q}^{(l)} \odot r_t^{(l)}. \tag{6.6}$$

This *gated radar conditioning* allows radar geometry and motion cues to bias the query used for temporal reasoning, while preserving the original BEV query pathway. The radar-primed query is then combined with the BEV positional encoding:

$$Q_{t,p}^{(l)} = \hat{Q}_t^{(l)} + P_{\mathrm{bev}}. \tag{6.7}$$

The resulting query is used in ego-aligned deformable temporal attention over the previous BEV memory and the current BEV representation. Since the query has already been primed with radar features, the subsequent offset and attention-weight prediction becomes radar-aware through the query representation itself. This corresponds to the guided temporal offset in Figure 6.3: the offset predictor is guided by radar because it receives the radar-primed query. This allows radar range and motion cues to benefit temporal sampling naturally.

Let $O_{t,\mathrm{hist}}^{(l)}$ and $O_{t,\mathrm{cur}}^{(l)}$ denote the historical and current outputs of temporal deformable attention. CRISP replaces uniform temporal averaging with a *query-conditioned temporal gate*:

$$g_{t,\mathrm{temp}}^{(l)} = \sigma\Big(\phi_{\mathrm{temp}}^{(l)}(Q_{t,p}^{(l)})\Big). \tag{6.8}$$

Here, $\phi_{\mathrm{temp}}^{(l)}(\cdot)$ denotes a lightweight two-layer MLP. The gated temporal output is

$$O_{t,\mathrm{temp}}^{(l)} = g_{t,\mathrm{temp}}^{(l)} \odot O_{t,\mathrm{cur}}^{(l)} + \Big(1 - g_{t,\mathrm{temp}}^{(l)}\Big) \odot O_{t,\mathrm{hist}}^{(l)}. \tag{6.9}$$

Finally, the temporal output is projected back to the BEV query space and added to the

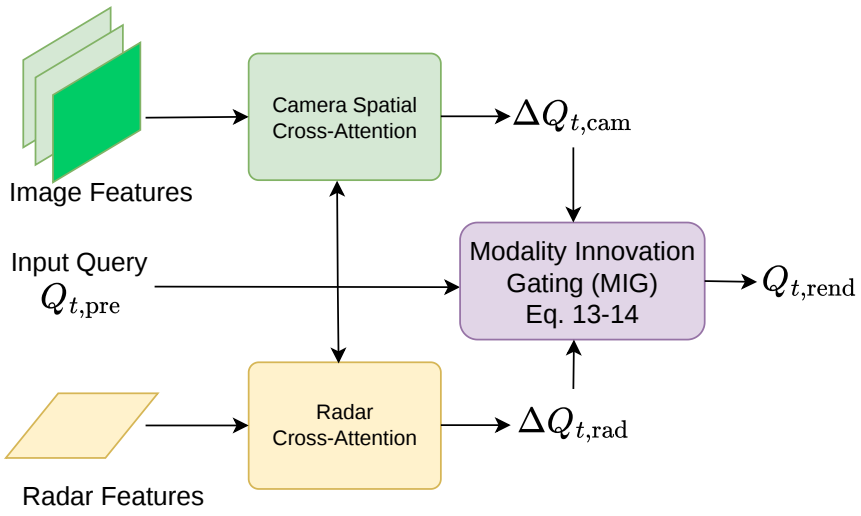



Figure 6.4: Multimodal feature rendering. Camera and radar cross-attention branches produce residual innovations, which are selectively admitted by the Modality Innovation Gate (MIG). The pre-attention BEV query is preserved through the residual path.

radar-primed query:

$$Q_{t,\text{tsa}}^{(l)} = \text{LN}\left(\hat{Q}_t^{(l)} + \text{Dropout}\left(\text{Linear}_o^{(l)}\left(O_{t,\text{temp}}^{(l)}\right)\right)\right). \tag{6.10}$$

Here, $\text{LN}(\cdot)$ denotes layer normalization.

This module provides a gated temporal update in BEV space. Temporal attention first extracts current and historical features, and the learned gate determines how they are fused. Because this fusion is repeated in every encoder layer, temporal information is progressively refined throughout the six-layer backbone.

#### 6.4.3.2 Multimodal Feature Rendering

After temporal aggregation, each BEV token must incorporate sensor evidence from both cameras and radar. As shown in Figure 6.4, CRISP replaces ViDAR's camera-centric latent rendering with an in-backbone multimodal rendering step. This choice is important in the CR setting: while latent rendering is useful when geometry must be inferred from images alone, radar already provides calibrated BEV measurements with range and motion cues. Therefore, CRISP directly renders camera and radar evidence into the BEV token at every encoder layer.

We implement this step with a *Modality Innovation Gate* (MIG). The camera and radar branches first produce modality-specific residual updates relative to the current BEV query. MIG predicts separate gates for these updates and controls how much camera and radar

information is fused into the BEV representation. Thus, each layer can rely on camera features, radar features, both, or neither, depending on the local scene context.

Let $Q^{(l)}_{t,\mathrm{pre}} = Q^{(l)}_{t,\mathrm{tsa}}$ denote the BEV query before cross-modal interaction. The camera branch uses BEVFormer-style spatial cross-attention to render multi-view image features into the BEV representation:

$$O^{(l)}_{t,\mathrm{cam}} = \mathrm{SCA}^{(l)}\left(Q^{(l)}_{t,\mathrm{pre}}, F^{\mathrm{cam}}_{t}\right), \qquad \Delta Q^{(l)}_{t,\mathrm{cam}} = W^{(l)}_{\mathrm{cam}} O^{(l)}_{t,\mathrm{cam}}. \tag{6.11}$$

Here, $\mathrm{SCA}(\cdot)$ samples image features from valid camera projections of BEV reference points. In parallel, the radar branch performs BEV-to-BEV deformable radar cross-attention (RCA) over the calibrated radar feature map:

$$O^{(l)}_{t,\mathrm{rad}} = \mathrm{RCA}^{(l)}\left(Q^{(l)}_{t,\mathrm{pre}}, F^{\mathrm{rad}}_{t}\right), \qquad \Delta Q^{(l)}_{t,\mathrm{rad}} = W^{(l)}_{\mathrm{rad}} O^{(l)}_{t,\mathrm{rad}}. \tag{6.12}$$

Since radar features are already represented in the BEV coordinate frame, $\mathrm{RCA}(\cdot)$ directly samples from the radar BEV feature field using 2D BEV reference points, avoiding the projection and visibility masking required by camera cross-attention.

MIG predicts independent query-conditioned gates for the two modality innovations:

$$g^{(l)}_{t,\mathrm{cam}} = \sigma\left(\phi^{(l)}_{\mathrm{cam}}(Q^{(l)}_{t,\mathrm{pre}})\right), \qquad g^{(l)}_{t,\mathrm{rad}} = \sigma\left(\phi^{(l)}_{\mathrm{rad}}(Q^{(l)}_{t,\mathrm{pre}})\right). \tag{6.13}$$

Here, $\phi^{(l)}_{\mathrm{cam}}(\cdot)$ and $\phi^{(l)}_{\mathrm{rad}}(\cdot)$ denote layer-specific lightweight MLPs that predict the camera and radar gates, respectively. The gates are implemented group-wise over the channel dimension and are broadcast within each group. For notation simplicity, we write the gated fusion at the token level:

$$Q^{(l)}_{t,\mathrm{rend}} = Q^{(l)}_{t,\mathrm{pre}} + g^{(l)}_{t,\mathrm{cam}} \odot \Delta Q^{(l)}_{t,\mathrm{cam}} + g^{(l)}_{t,\mathrm{rad}} \odot \Delta Q^{(l)}_{t,\mathrm{rad}}. \tag{6.14}$$

The residual path $Q^{(l)}_{t,\mathrm{pre}}$ is preserved explicitly, while MIG controls the camera and radar updates applied to it.

The camera and radar gates are independent rather than competitive (e.g., softmax-normalized). This is important because the two sensors are complementary rather than mutually exclusive. In well-observed regions, both modalities can be useful; for distant or dynamic objects, radar may provide more reliable range and motion cues; for semantic structure, camera features may dominate; and when the existing BEV feature is already confident, both updates can be suppressed. Since each layer has its own MIG parameters, the backbone can adjust the fusion behavior at different depths.

Finally, the rendered feature is passed through a position-wise feed-forward network:

$$Q_t^{(l)} = \mathrm{LN}\Big(Q_{t,\mathrm{rend}}^{(l)} + \mathrm{FFN}^{(l)}\Big(Q_{t,\mathrm{rend}}^{(l)}\Big)\Big). \tag{6.15}$$

Here, $\mathrm{LN}(\cdot)$ denotes layer normalization.

In this way, multimodal feature rendering extends gated feature update from temporal fusion to sensor fusion. Rather than directly adding all camera and radar attention outputs, CRISP selectively fuses modality updates according to the current BEV representation.

### 6.4.4 Forecasting-Based Pretraining Objective

CRISP adopts the forecasting-based pretraining objective of ViDAR [203] to provide predictive geometric supervision for the learned BEV representation. We do not introduce a new forecasting loss; instead, we use the ViDAR-style occupancy prediction decoder to supervise the proposed CR BEV backbone. This allows the backbone to learn geometry, free space, and temporal dynamics from LiDAR point clouds during pretraining, while requiring only camera and radar inputs at inference time.

#### 6.4.4.1 Occupancy Prediction Head

Following ViDAR [203], we attach an autoregressive occupancy prediction head only during pretraining. Let $\mathcal{B}_t = \{B_{t-T+1}^{\mathrm{fused}}, \ldots, B_t^{\mathrm{fused}}\}$ denote the queue of multimodal BEV states produced by the CR backbone. At future step $\tau$, the decoder predicts a future BEV state conditioned on the previously predicted BEV states and a learned embedding of the relative future ego-motion:

$$\hat{B}_{t+\tau} = \Psi_{\mathrm{pred}}(\mathcal{B}_{t+\tau-1}, e_{\mathrm{ego}}(\Delta T_{t\to t+\tau}^{\mathrm{ego}})), \quad \tau = 1, \ldots, T_{\mathrm{fut}}. \tag{6.16}$$

Here, $\Delta T_{t\to t+\tau}^{\mathrm{ego}}$ denotes the relative ego-motion from the current frame to the target future frame. During pretraining, it is obtained from the recorded ego trajectory. When the prediction head is used for point cloud forecasting evaluation, we follow ViDAR [203] and condition all compared methods on the same future ego-motion transform used only to define the future ego-centric coordinate frame. This does not include future sensor observations or future object states.

In our configuration, the prediction decoder follows ViDAR's transformer-based design, but operates on the multimodal BEV representation produced by CRISP. The decoder pre-

dicts occupancy logits directly over the BEV grid and vertical bins:

$$\hat{O}_{t+\tau} \in \mathbb{R}^{H_{\text{bev}} \times W_{\text{bev}} \times Z}, \tag{6.17}$$

where $Z = 16$ is the number of vertical occupancy bins. We follow ViDAR [203] to convert the predicted occupancy volume into a forecast point set for point cloud forecasting evaluation. In addition to future occupancy prediction, we also use auxiliary current-frame occupancy prediction to stabilize training, following ViDAR [203].

#### 6.4.4.2 Pretraining Loss

LiDAR point clouds are used as privileged supervision for the predicted occupancy volumes. Following ViDAR [203], we supervise the prediction decoder with a sparse LiDAR occupancy classification loss and a dense voxel occupancy reconstruction loss:

$$\mathcal{L}_{\text{pretrain}} = \sum_{\tau=0}^{T_{\text{fut}}} w_\tau \left( \mathcal{L}_{\text{occ}}^{(\tau)} + \lambda_{\text{dense}} \mathcal{L}_{\text{dense}}^{(\tau)} \right), \tag{6.18}$$

where $\tau = 0$ denotes the auxiliary current-frame prediction and $\tau > 0$ denotes future predictions, $w_\tau$ is the loss weight for prediction step $\tau$, and $\lambda_{\text{dense}}$ controls the contribution of the dense voxel loss. The sparse occupancy loss $\mathcal{L}_{\text{occ}}^{(\tau)}$ encourages the predicted occupancy field to assign high probability to LiDAR-observed surface locations and low probability to competing samples along the corresponding sensor line, following the ViDAR formulation. The dense loss $\mathcal{L}_{\text{dense}}^{(\tau)}$ further supervises the reconstructed voxel occupancy field to encourage spatially coherent geometry.

### 6.4.5 Downstream Finetuning

After pretraining, we discard the occupancy prediction head and retain the camera branch, radar encoder, and CR spatiotemporal BEV backbone as initialization for downstream tasks. Task-specific heads are then attached to the pretrained BEV representation and optimized with the corresponding downstream supervision. The resulting representation is unified, temporally aware, and directly compatible with standard BEV-centric perception, prediction, or planning heads [93, 52]. Since LiDAR is used only as privileged supervision during pretraining, the finetuned model remains a practical CR system at inference time.

## 6.5 Experiments

### 6.5.1 Experimental Setup

#### 6.5.1.1 Dataset

We conduct experiments on nuScenes [8], a large-scale autonomous driving dataset with synchronized multi-view cameras, LiDAR, radar, ego-motion, and annotations for perception-centric evaluation. Its synchronized camera–radar streams make it well suited for studying LiDAR-free multimodal representation learning. During pretraining, the model takes historical CR sequences as input, while LiDAR point clouds are used only as privileged supervision for occupancy forecasting. During downstream finetuning and inference, LiDAR is not used; the model operates only with camera and radar inputs.

#### 6.5.1.2 Implementation Details

All main results (Section 6.5.2) use the full CRISP configuration described in Section 6.4. The camera encoder and the spatiotemporal BEV backbone mainly follow common practice in [93, 52, 203]. Specifically, the image backbone (i.e., ResNet-101 [47]) is initialized from the checkpoint trained in [174] and a feature pyramid network (FPN) [99] neck is used to produce four levels of image features. The BEV feature dimension is set to $C = 256$, and the BEV grid resolution is $200 \times 200$ over the range $[-51.2, 51.2]$m in both the longitudinal and lateral directions. The BEV backbone contains six encoder layers. For radar input, we use 6 radar sweeps. Radar points are voxelized with voxel size $[0.512, 0.512, 8]$ and encoded by the proposed enhanced radar encoder built on PointPillars [79]. The resulting radar BEV feature is aligned to the same $200 \times 200$ BEV grid as the camera-side BEV representation. Radar cross-attention is enabled in all six encoder layers, and the Modality Innovation Gate uses 8 channel groups for group-wise gated fusion, matching the number of heads used by deformable attention.

For forecasting-based pretraining, CRISP uses the ViDAR-style occupancy prediction head described in Section 6.4. The prediction decoder has three transformer layers and predicts occupancy logits over 16 vertical bins. During training, we supervise the current frame and three future frames using the dense loss weight $\lambda_{\text{dense}}$. The supervised frame weights are set to 1 for the current frame and the three future frames. During forecasting evaluation, the model is rolled out to six future frames.

We pretrain the CRISP backbone for 24 epochs using a learning rate of $2 \times 10^{-4}$. Experiments are conducted on 8 NVIDIA A100 GPUs with one sample per GPU. After pretraining, we use the pretrained CRISP model as the initialization for downstream models (i.e., BEV-

Former [93] and UniAD [52]). Specifically, we retain the camera encoder, radar encoder, and CR spatiotemporal BEV backbone, and connect the resulting unified BEV feature to task-specific heads. Following ViDAR [203], we initialize UniAD training with the weights trained for the detection task. We use the same training schedule for BEVFormer and UniAD as ViDAR for a fair comparison.

#### 6.5.1.3 Evaluation Protocol

We evaluate CRISP from two complementary perspectives. First, following [67, 203], we directly evaluate point cloud forecasting quality to assess whether the pretrained representation captures future 3D geometry and scene dynamics. We report Chamfer Distance (CD) at future timestamps from 0.5s to 3.0s.

Second, we transfer the pretrained backbone to downstream autonomous driving tasks under the UniAD-style evaluation protocol [52, 203]. For 3D object detection, we report mean Average Precision (mAP) and nuScenes Detection Score (NDS). For multi-object tracking, we report Average Multi-Object Tracking Accuracy (AMOTA), Average Multi-Object Tracking Precision (AMOTP), and Recall. For map segmentation, we report intersection-over-union (IoU) for lane, drivable area, divider, and pedestrian crossing categories. For motion forecasting, we report minimum average displacement error (minADE), minimum final displacement error (minFDE), Miss Rate (MR), and End-to-end Prediction Accuracy (EPA). minADE and minFDE measure the best-mode average and final displacement errors, MR measures the fraction of missed agents, and EPA summarizes end-to-end prediction quality. For future occupancy prediction, we report IoU and Video Panoptic Quality (VPQ) over near and far regions, where IoU measures voxel overlap and VPQ measures spatiotemporal occupancy quality. For planning, we report average $\mathcal{L}_2$ displacement error and average collision rate over the 3s planning horizon. Together, these metrics evaluate whether the pretrained CRISP backbone transfers to perception, prediction, and planning.

### 6.5.2 Main Results

We evaluate CRISP on point cloud forecasting and a broad set of downstream autonomous driving tasks, including 3D detection, tracking, map segmentation, motion forecasting, future occupancy prediction, and planning. We report three variants of our method:

- **CRISP**: the pretrained CR backbone evaluated with the occupancy forecasting head for future point cloud prediction.
- **BEVFormer-CRISP**: BEVFormer with its original detection head retained and its

Table 6.1: Point cloud forecasting on nuScenes following the ViDAR protocol [203]. We report Chamfer Distance (CD) for future timestamps from 0.5s to 3.0s. The best result is shown in **bold** and the second-best result is underlined. † indicates a language-augmented world model [229] with an LLM-based component and additional text supervision, which is different from the sensor-only pretraining setting of the other listed methods.

| History Horizon | Method | Modality | Chamfer Distance ($m^2$) ↓ | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 0.5s | 1.0s | 1.5s | 2.0s | 2.5s | 3.0s |
| 0s | HERMES† [229] | C | – | 0.78 | – | 0.95 | – | 1.17 |
| | LRS4Fusion [127] | C+L | **0.35** | **0.57** | 0.75 | 0.96 | 1.21 | 1.51 |
| | CRISP (ours) | C+R | 0.52 | 0.59 | **0.67** | **0.74** | **0.82** | **0.91** |
| 1s | 4D-Occ [67] | L | 1.26 | 1.88 | – | – | – | – |
| | ViDAR [203] | C | 1.11 | 1.25 | 1.40 | 1.57 | 1.76 | 1.97 |
| | LRS4Fusion [127] | C+L | **0.31** | **0.48** | 0.64 | 0.79 | 0.99 | 1.25 |
| | CRISP (ours) | C+R | 0.47 | 0.53 | **0.60** | **0.68** | **0.76** | **0.86** |
| 3s | 4D-Occ [67] | L | 0.91 | 1.13 | 1.30 | 1.53 | 1.72 | 2.11 |
| | ViDAR [203] | C | 1.01 | 1.12 | 1.25 | 1.38 | 1.54 | 1.73 |
| | LRS4Fusion [127] | C+L | **0.33** | **0.47** | 0.61 | 0.77 | 0.97 | 1.23 |
| | CRISP (ours) | C+R | 0.44 | 0.51 | **0.58** | **0.65** | **0.74** | **0.84** |

BEV backbone replaced by the pretrained CRISP backbone.

- **UniAD-CRISP**: UniAD with its original task heads retained and its BEV backbone replaced by the pretrained CRISP backbone.

#### 6.5.2.1 Point Cloud Forecasting

Table 6.1 evaluates predictive 3D geometry under the ViDAR point cloud forecasting protocol [203]. ViDAR establishes this setting by pretraining a CO BEV encoder to forecast future LiDAR point clouds, providing a strong baseline for vision-based predictive representation learning. CRISP extends this paradigm to CR pretraining and demonstrates a clear advantage in long-horizon forecasting. While LRS4Fusion [127] benefits from LC inputs and performs best at short horizons, CRISP consistently overtakes it from 1.5s onward across all history settings. This result is particularly notable because LiDAR typically provides much stronger metric geometry than radar; nevertheless, the proposed CR backbone achieves more accurate long-horizon scene evolution. The trend suggests that CRISP does not simply ap-

Table 6.2: 3D detection results. All metrics are reported in percentage. The best result is shown in **bold** and the second-best result is underlined.

| Method | Modality | Pretrain | mAP ↑ | NDS ↑ |
|---|---|---|---|---|
| UVTR [87] | C | - | 39.2 | 48.8 |
| | | UniPAD [201] | 42.8 | 50.2 |
| | | VisionPAD [213] | 43.1 | 50.4 |
| BEVFormer [93] | C | - | 41.5 | 51.7 |
| | | ViDAR [203] | 45.8 | 54.8 |
| SpaRC-AD [189] | C+R | - | 46.6 | 57.0 |
| BEVFormer-CRISP | C+R | - | 49.8 | 59.1 |
| | | CRISP | **53.2** | **61.3** |

pend radar as an auxiliary cue, but uses radar-enhanced temporal attention and multimodal feature rendering to learn motion-aware BEV representations for future prediction. Under the 0s history setting, CRISP also outperforms HERMES [229] at the reported horizons, even though HERMES uses an LLM-based language-augmented world-model component and additional text supervision. Overall, these results highlight the central benefit of CRISP: accurate long-horizon 3D forecasting from a practical CR system compatible with LiDAR-free inference.

#### 6.5.2.2 Downstream Finetuning

We next study whether the backbone pretrained through forecasting transfers to downstream autonomous driving tasks. Our main CO baselines are BEVFormer [93] and UniAD [52], together with their ViDAR-pretrained counterparts. We also compare with SpaRC-AD [189], a strong CR multi-task model. For CRISP transfer, we replace the BEV backbone of BEVFormer or UniAD with the pretrained CRISP backbone while keeping the original downstream heads unchanged.

**3D Object Detection.** Table 6.2 reports 3D detection results on the nuScenes *val* set. In addition to BEVFormer-based baselines, we include UVTR [87], a unified CO 3D perception model, and its pretrained variants UniPAD [201] and VisionPAD [213]. BEVFormer-CRISP achieves 53.2 mAP and 61.3 NDS, outperforming both CO pretraining baselines and the CR

Table 6.3: Online map segmentation results. All IoU values are percentages. The best result is shown in **bold** and the second-best result is underlined.

| Method | Modality | Pretrain | IoU ↑ | | | |
|---|---|---|---|---|---|---|
| | | | Lane | Drivable | Divider | Crossing |
| UniAD [52] | C | - | 31.3 | 69.1 | 25.7 | 13.8 |
| | | ViDAR [203] | 35.2 | **73.9** | 29.1 | **17.1** |
| UniAD-CRISP | C+R | - | 32.4 | 69.8 | 26.7 | 15.1 |
| | | CRISP | **35.4** | 72.7 | **29.3** | 16.6 |

Table 6.4: Online tracking results. AMOTA is reported as a percentage, AMOTP in meters, and Recall as a ratio. The best result is shown in **bold** and the second-best result is underlined.

| Method | Modality | Pretrain | AMOTA ↑ | AMOTP ↓ | Recall ↑ |
|---|---|---|---|---|---|
| UniAD [52] | C | - | 35.9 | 1.32 | 0.48 |
| | | ViDAR [203] | 43.46 | 1.25 | 0.54 |
| SpaRC-AD [189] | C+R | - | 46.9 | 1.13 | 0.55 |
| UniAD-CRISP | C+R | - | 44.6 | 1.17 | 0.56 |
| | | CRISP | **48.6** | **1.12** | **0.57** |

SpaRC-AD baseline. Compared with ViDAR-pretrained BEVFormer, the gain suggests that CR forecasting-based pretraining provides a stronger BEV initialization for object-centric 3D perception. The improvement over SpaRC-AD further indicates that the benefit is not simply from adding radar, but from pretraining a reusable CR backbone before detection finetuning.

**Online Map Segmentation.** Table 6.3 reports online map segmentation results. UniAD-CRISP achieves the best lane and divider IoU, while remaining competitive on drivable area and crossing. The mixed trend is reasonable: map segmentation is dominated by mostly static scene elements, where camera semantics and spatial layout already provide strong cues, and radar contributes less directly than in dynamic-object tasks. Nevertheless, CRISP still improves over CO UniAD across all categories and matches or exceeds ViDAR on the finer lane and divider classes, suggesting that CR pretraining does not harm static BEV structure while primarily benefiting tasks that require motion and geometry.

**Online Tracking.** Table 6.4 evaluates multi-object tracking. UniAD-CRISP improves AMOTA, AMOTP, and Recall over both CO baselines and SpaRC-AD. This is consistent with the role of radar in the backbone: tracking depends not only on detecting objects in each frame, but also on maintaining stable object states across time. Radar range and motion cues provide complementary evidence for dynamic agents, helping the temporal BEV representation preserve object continuity under motion, occlusion, and appearance ambiguity.

**Motion Forecasting.** Table 6.5 reports motion forecasting results. UniAD-CRISP achieves the best minADE, minFDE, and EPA, while remaining second-best on MR. This indicates that CRISP improves the accuracy of the selected trajectory modes and the overall end-to-end prediction score. The result is aligned with the pretraining objective: forecasting future 3D structure encourages the backbone to encode dynamic scene evolution, and radar provides direct motion observations that are useful for downstream agent prediction.

**Future Occupancy Prediction.** Table 6.6 evaluates dense future occupancy prediction. UniAD-CRISP achieves the best results across near and far regions for both VPQ and IoU, with especially strong gains in the far field. This is where radar is particularly valuable: CO depth estimates become less reliable at long range, while radar provides metric range measurements and motion-sensitive returns. Since future occupancy prediction requires both spatial geometry and temporal evolution, the combination of LiDAR-supervised forecasting pretraining and radar-aware BEV encoding transfers naturally to this task.

**Planning.** Table 6.7 reports planning performance on the nuScenes *val* set. UniAD is explicitly designed as a planning-oriented framework, so improvements here are especially meaningful. UniAD-CRISP reduces both average $\mathcal{L}_2$ error and collision rate, outperforming ViDAR-pretrained UniAD and SpaRC-AD. The lower collision rate suggests that the improved CR BEV representation is not only beneficial for intermediate perception metrics, but also for open-loop safety-related planning metrics. Overall, the downstream results show that CRISP transfers most strongly to tasks where geometry, temporal consistency, and object dynamics are central, while remaining competitive on more static BEV understanding tasks such as map segmentation.

Table 6.5: Motion forecasting results. minADE and minFDE are reported in meters. The best result is shown in **bold** and the second-best result is underlined.

| Method | Modality | Pretrain | minADE ↓ | minFDE ↓ | MR ↓ | EPA ↑ |
|---|---|---|---|---|---|---|
| UniAD [52] | C | - | 0.71 | 1.02 | 0.151 | 0.456 |
| | | ViDAR [203] | 0.65 | 0.97 | 0.147 | 0.513 |
| SpaRC-AD [189] | C+R | - | 0.58 | 0.93 | **0.121** | 0.530 |
| UniAD-CRISP | C+R | - | 0.54 | 0.87 | 0.139 | 0.545 |
| | | CRISP | **0.53** | **0.85** | 0.134 | **0.571** |

Table 6.6: Future occupancy prediction results. All values are percentages. The best result is shown in **bold** and the second-best result is underlined.

| Method | Modality | Pretrain | VPQ-n ↑ | VPQ-f ↑ | IoU-n ↑ | IoU-f ↑ |
|---|---|---|---|---|---|---|
| UniAD [52] | C | - | 54.7 | 33.5 | 63.4 | 40.2 |
| | | ViDAR [203] | 58.1 | 37.8 | 65.7 | 42.6 |
| UniAD-CRISP | C+R | - | 60.8 | 45.5 | 67.7 | 46.6 |
| | | CRISP | **62.7** | **47.5** | **69.1** | **48.0** |

### 6.5.3 Label-Efficiency of Pretraining

We further evaluate whether the pretrained CRISP backbone improves label efficiency during downstream finetuning. We use the same BEVFormer-CRISP detection architecture and train it for 12 epochs with different fractions of the nuScenes training split: 1/4, 1/2, and the full set. We compare two initialization settings: training from scratch and initializing from CRISP pretraining. The scratch baseline still uses the same FCOS3D-pretrained camera encoder as in our main experiments, so the comparison directly highlights the benefit of CRISP backbone pretraining for camera-radar detection.

As shown in Figure 6.5, CRISP consistently improves both mAP and NDS across all training-set sizes. The gain is especially clear in the low-label regime: with only 1/4 or 1/2 of the training data, CRISP improves NDS and mAP by more than 3 percentage points. The improvement remains stable as more annotations are used, reaching 59.9 NDS and 51.6 mAP with the full training set. These results show that the pretrained CRISP backbone provides a stronger initialization for downstream detection, reducing the dependence on dense 3D box annotations while still improving full-data performance.

Table 6.7: End-to-end planning results. Avg. $\mathcal{L}_2$ is reported in meters, and Avg. Collision is reported as a percentage. The best result is shown in **bold** and the second-best result is underlined.

| Method | Modality | Pretrain | Avg. $\mathcal{L}_2$ (m) ↓ | Avg. Collision (%) ↓ |
|---|---|---|---|---|
| UniAD [52] | C | - | 1.03 | 0.31 |
| UniAD [52] | C | ViDAR [203] | 0.94 | 0.20 |
| SpaRC-AD [189] | C+R | - | 0.93 | 0.23 |
| UniAD-CRISP | C+R | - | 0.85 | 0.20 |
| | | CRISP | **0.80** | **0.15** |

## 6.5.4 Ablation Study

### 6.5.4.1 Implementation Details

To reduce GPU memory usage, we mainly follow the lightweight BEVFormer-small setting [93] for ablation experiments. Compared with the full CRISP model, the ablation setting uses a smaller $160 \times 160$ BEV grid, single-scale image features extracted from images resized to $0.8\times$ of the full input resolution, and a 3-layer BEV encoder, while keeping the training recipe consistent across variants. For efficiency, ablation pretraining uses a fixed 1/8 subset of the nuScenes *train* split, and detection finetuning uses a fixed 1/4 subset. All ablation models are trained with the same loss configuration as the full model, and still evaluated on the full nuScenes validation split. We train all ablation models using 4 GPUs.

The ablations are designed to assess both the contribution of each CRISP component and the effectiveness of our specific design choices in the radar encoder, radar-enhanced temporal self-attention, and multimodal feature rendering. Unless otherwise noted, we evaluate forecasting with Chamfer Distance (CD) and Average Euclidean Error (AEE), and evaluate downstream detection with mean Average Precision (mAP), nuScenes Detection Score (NDS), and mean Average Velocity Error (mAVE). CD measures set-level similarity between predicted and ground-truth point clouds using squared nearest-neighbor distances, while AEE measures metric prediction accuracy in meters after associating predictions with corresponding ground-truth viewing directions. For detection, mAP and NDS capture overall object detection quality, while mAVE measures velocity estimation error in meters per second, which is closely related to dynamic scene understanding.




Figure 6.5: Label-efficiency of CRISP pretraining on nuScenes 3D detection. We train the same BEVFormer-CRISP detection model for 12 epochs using 1/4, 1/2, and the full nuScenes training split. The scratch baseline uses the same FCOS3D-pretrained camera encoder but does not use CRISP backbone pretraining. CRISP improves both NDS and mAP across all data scales, with clear gains in the low-label regime.

#### 6.5.4.2 Component Ablation

Table 6.8 studies the cumulative contribution of the main CRISP components. Adding radar-enhanced temporal self-attention gives a large forecasting gain over ViDAR, confirming that radar motion cues are useful when injected before temporal aggregation. However, downstream transfer remains incomplete: mAVE is still high, suggesting that radar-aware temporal propagation alone does not fully provide motion-consistent BEV features for detection. Replacing ViDAR's camera-centric latent rendering with multimodal feature rendering addresses this issue. As discussed in Section 6.4.3.2, latent rendering is suitable when geometry must be inferred from images, but in CRISP radar already provides calibrated metric BEV features with range and motion cues; therefore, camera and radar evidence should be composed directly inside the BEV backbone. This change produces the largest down-

Table 6.8: Main ablation study of the proposed CRISP modules. Starting from the ViDAR baseline, each subsequent row cumulatively adds one additional component on top of the previous row: radar-enhanced TSA, multimodal feature rendering, and the enhanced radar encoder. mAP and NDS are reported in percentage points (%), and mAVE is reported in meters per second (m/s). The best result for each metric is **bolded**.

| Variant | CD ($m^2$) ↓ | | | AEE (m) ↓ | | | Detection Finetuning | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1s | 2s | 3s | 1s | 2s | 3s | mAP ↑ | NDS ↑ | mAVE ↓ |
| ViDAR [203] | 1.42 | 1.65 | 1.95 | 2.67 | 2.81 | 2.98 | 31.3 | 34.9 | 1.19 |
| + radar-enhanced TSA | 0.97 | 1.22 | 1.54 | 1.78 | 1.93 | 2.14 | 37.8 | 39.8 | 1.23 |
| + multimodal feature rendering | 0.90 | 1.12 | 1.43 | 1.68 | 1.83 | 2.01 | 41.1 | 48.8 | **0.49** |
| + enhanced radar encoder | **0.88** | **1.10** | **1.41** | **1.66** | **1.78** | **1.93** | **41.8** | **49.5** | **0.49** |

stream improvement, especially in NDS and mAVE. The enhanced radar encoder then gives a smaller but consistent final gain, validating the full design of ego-aware radar encoding, radar-enhanced temporal propagation, and iterative multimodal evidence rendering.

#### 6.5.4.3 Radar Encoder Design

Table 6.9 ablates the enhanced radar encoder. The PointPillars baseline already provides a reasonable radar BEV feature, but it does not explicitly account for the ego-centric nature of radar motion measurements. Adding ego-motion awareness improves AEE and slightly improves NDS, suggesting that ego-conditioned radar features help interpret Doppler cues more consistently; however, mAP slightly drops and mAVE worsens, so the effect is not uniformly positive for detection. Residual spatial calibration then improves CD and gives the best or tied-best detection metrics, indicating that aligning radar BEV features with the camera-side BEV representation is useful before radar participates in temporal attention and multimodal rendering.

#### 6.5.4.4 Radar-Enhanced Temporal Self-Attention

Table 6.10 evaluates the two design choices in radar-enhanced temporal self-attention. Gated radar conditioning mostly improves over standard TSA, especially on AEE and detection metrics, although CD at 3s slightly worsens. This shows that radar-primed BEV queries make temporal propagation more effective for many, but not all, metrics. Adding temporal gating gives the strongest aggregate result, especially on AEE and downstream velocity estimation. This supports the view that temporal aggregation should not uniformly average current and historical branches; instead, each BEV token should decide how much current

Table 6.9: Ablation study of the radar encoder design. Starting from the PointPillars radar encoder, each subsequent row cumulatively adds one additional component on top of the previous row: ego-motion awareness and residual spatial calibration. All other CRISP modules are kept the same for fair comparison. mAP and NDS are reported in percentage points (%), and mAVE is reported in meters per second (m/s). The best result for each metric is **bolded**.

| Variant | CD ($m^2$) ↓ | | | AEE (m) ↓ | | | Detection Finetuning | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1s | 2s | 3s | 1s | 2s | 3s | mAP ↑ | NDS ↑ | mAVE ↓ |
| PointPillars [79] | 0.90 | 1.12 | 1.43 | 1.68 | 1.82 | 2.01 | 41.1 | 48.8 | **0.49** |
| + Ego-motion awareness | 0.90 | 1.12 | 1.43 | **1.66** | **1.78** | **1.92** | 41.0 | 49.0 | 0.51 |
| + Residual spatial calibration | **0.88** | **1.10** | **1.41** | **1.66** | **1.78** | 1.93 | **41.8** | **49.5** | **0.49** |

evidence and propagated memory to use.

#### 6.5.4.5 Multimodal Feature Rendering

Table 6.11 compares different ways to compose camera and radar evidence after temporal aggregation. Adaptive scalar weighting is competitive for forecasting, including the best 3s CD, but weaker for downstream transfer, while full gating degrades detection performance, suggesting that gating complete modality features can disturb the existing BEV state. Innovation gating achieves the best downstream transfer while maintaining competitive forecasting because it gates only the modality-specific residual updates proposed by camera and radar cross-attention. This preserves the current BEV representation while allowing each layer to selectively admit useful sensor evidence, which is the intended behavior of the MIG.

Overall, the ablations show that CRISP's improvements come from introducing radar at multiple stages of the BEV state update process. Radar-enhanced TSA improves temporal propagation, multimodal feature rendering provides the main transfer gain by replacing camera-centric latent rendering, and the enhanced radar encoder improves the quality of the radar evidence used throughout the backbone.

### 6.5.5 Qualitative Analysis

We visualize future point cloud prediction results to better understand what CRISP learns from CR forecasting pretraining. In all examples, the model only observes historical CR inputs; future camera frames are shown only for visualization. Predicted point clouds are compared with ground truth over future horizons, with CRISP and ViDAR shown side by side.

Table 6.10: Ablation study of design choices in radar-enhanced temporal self-attention. Starting from standard TSA, each subsequent row cumulatively adds gated radar conditioning and temporal gating. All other CRISP modules are kept the same for fair comparison. mAP and NDS are reported in percentage points (%), and mAVE is reported in meters per second (m/s). The best result for each metric is **bolded**.

| Variant | CD ($m^2$) ↓ | | | AEE (m) ↓ | | | Detection Finetuning | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1s | 2s | 3s | 1s | 2s | 3s | mAP ↑ | NDS ↑ | mAVE ↓ |
| Standard TSA | 0.91 | 1.12 | **1.41** | 1.69 | 1.82 | 1.99 | 40.8 | 48.6 | 0.52 |
| + Gated radar conditioning | 0.89 | 1.12 | 1.43 | 1.67 | 1.80 | 1.95 | 41.7 | 49.4 | 0.52 |
| + Temporal Gating | **0.88** | **1.10** | **1.41** | **1.66** | **1.78** | **1.93** | **41.8** | **49.5** | **0.49** |

Table 6.11: Ablation study of modality fusion design in multimodal feature rendering. All other CRISP modules are kept the same for fair comparison. mAP and NDS are reported in percentage points (%), and mAVE is reported in meters per second (m/s). The best result for each metric is **bolded**.

| Variant | CD ($m^2$) ↓ | | | AEE (m) ↓ | | | Detection Finetuning | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1s | 2s | 3s | 1s | 2s | 3s | mAP ↑ | NDS ↑ | mAVE ↓ |
| Adaptive scalar weight [94] | 0.90 | 1.12 | **1.40** | 1.67 | 1.79 | 1.95 | 41.2 | 48.9 | 0.53 |
| Full Gating | 0.90 | 1.12 | 1.46 | 1.67 | 1.79 | 1.95 | 40.5 | 47.7 | 0.57 |
| Innovation Gating | **0.88** | **1.10** | 1.41 | **1.66** | **1.78** | **1.93** | **41.8** | **49.5** | **0.49** |

Figures 6.6 and 6.7 illustrate two representative strengths of CRISP. In the low-light scene of Figure 6.6, the CO ViDAR baseline loses the forward vehicle over time and exhibits large spatial drift, while CRISP maintains a more stable prediction of the vehicle trajectory. This case highlights the value of radar motion cues under challenging illumination, where visual appearance alone is unreliable. In Figure 6.7, ViDAR propagates an erroneous structure from a billboard-like image region into the predicted geometry. CRISP better preserves the physical scene layout, suggesting that radar-enhanced BEV features help anchor the future prediction to metric scene structure rather than purely visual texture.

Figure 6.8 shows a remaining failure mode. The highlighted vehicle is stopped at a red light, but the traffic signal is small and difficult to infer from the input views. Although CRISP predicts the vehicle geometry more accurately than ViDAR at early horizons, its prediction begins to drift after 2s. This suggests that when future behavior depends on high-level traffic semantics or agent intent, geometry and motion priors alone may be insufficient. These examples indicate that CRISP improves future world-geometry prediction in visually

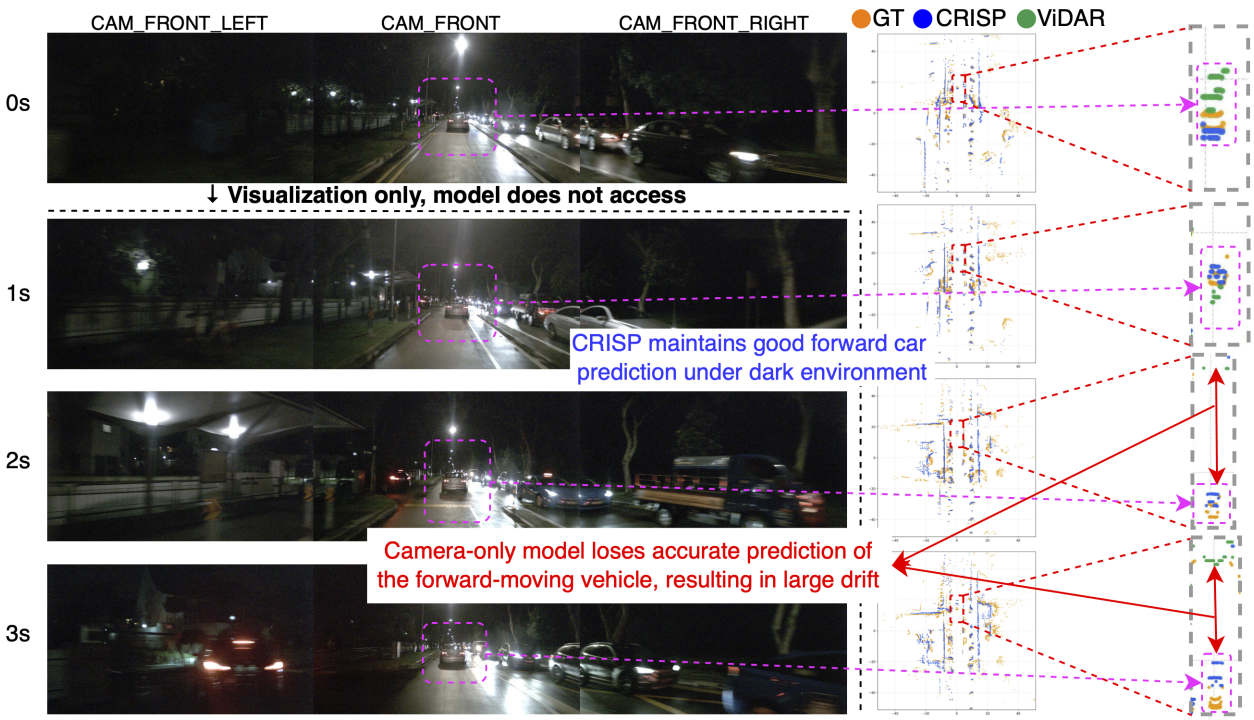



Figure 6.6: Qualitative comparison between CRISP and ViDAR for future world-geometry prediction in a low-light scene. We show the front camera views at the current frame (0s) and future timestamps for visualization only; future images are not provided to the model. On the right, predicted point clouds are compared with ground truth. In this example, the CO ViDAR baseline loses the forward vehicle over time and exhibits large spatial drift, while CRISP maintains a more accurate prediction of the vehicle trajectory. This illustrates the benefit of radar motion cues for forecasting dynamic objects under challenging illumination, which is important for downstream trajectory evaluation and safe planning around leading vehicles.

ambiguous and motion-heavy scenes, while also motivating future work on incorporating stronger traffic-rule, signal-state, or intent-level reasoning.

## 6.6 Discussion

CRISP frames CR fusion as predictive representation learning rather than task-specific supervised fusion. By forecasting future LiDAR point clouds from historical CR observations, the model learns a reusable spatiotemporal BEV representation that encodes geometry, motion, and cross-modal interactions, while still requiring only camera and radar at inference time. This connects two previously separate directions: forecasting-based world-model pretraining for driving and practical CR fusion for deployable autonomy. The downstream results further suggest that the learned representation is broadly useful beyond the pretext task, with

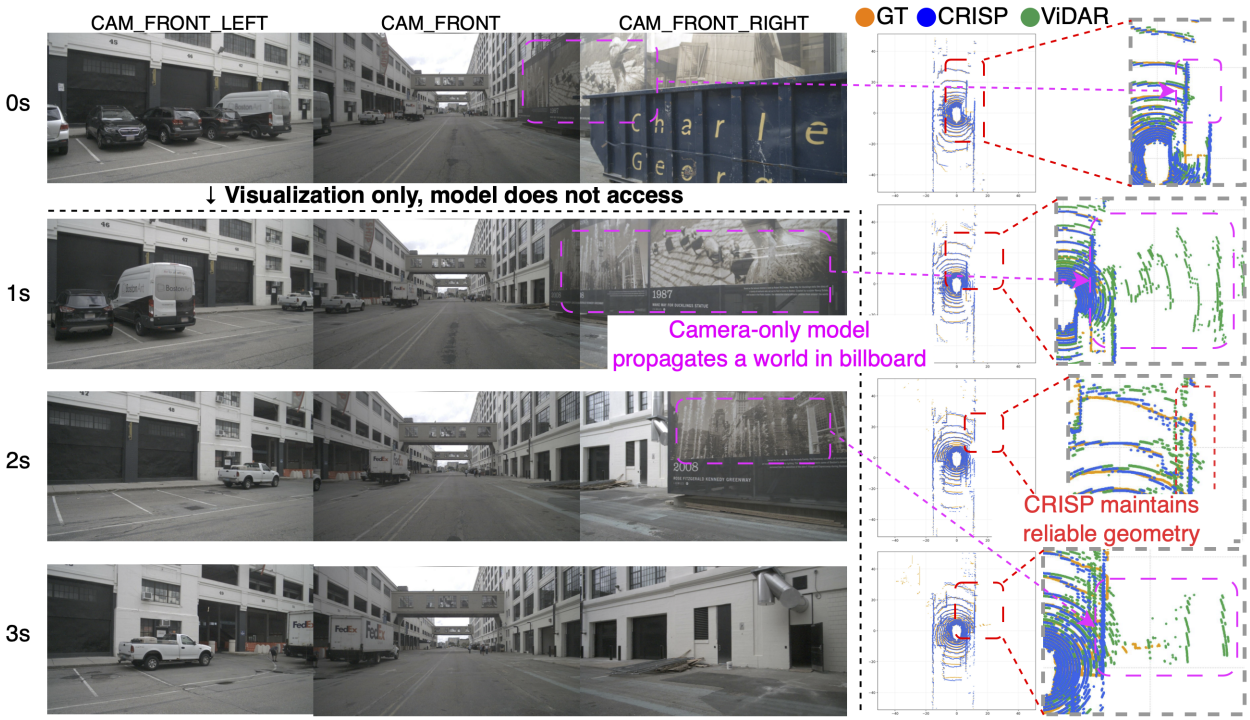



Figure 6.7: Qualitative comparison between CRISP and ViDAR for future world-geometry prediction. We show the front camera views at the current frame (0s) and future timestamps for visualization only; future images are not provided to the model. On the right, predicted point clouds are compared with ground truth, where CRISP better preserves the scene geometry over time. In this example, the CO ViDAR baseline propagates an erroneous structure from a billboard region, while CRISP maintains more reliable geometry.

the strongest gains appearing in tasks that depend on geometry, temporal consistency, and agent motion, such as detection, tracking, occupancy prediction, motion forecasting, and planning. The more modest gains on mostly static map elements also suggest that radar contributes most when dynamic scene evolution is important.

Despite these benefits, CRISP still has important failure modes. First, the forecasting objective is data-driven and does not explicitly model causality, traffic rules, or agent intent (Figure 6.8). In common situations, predicting likely future scene structure is useful; however, in rare safety-critical cases, a model may favor statistically common continuations from the training data even when they are inappropriate for the current scene. For example, near an occluded intersection with a red traffic light, the model may still predict forward traffic flow if similar visual-geometric observations in the training set usually contain moving vehicles. Robust prediction therefore requires higher-level understanding of signals, right-of-way, occlusion, map context, and interactive intent, which are not directly captured by the current objective.

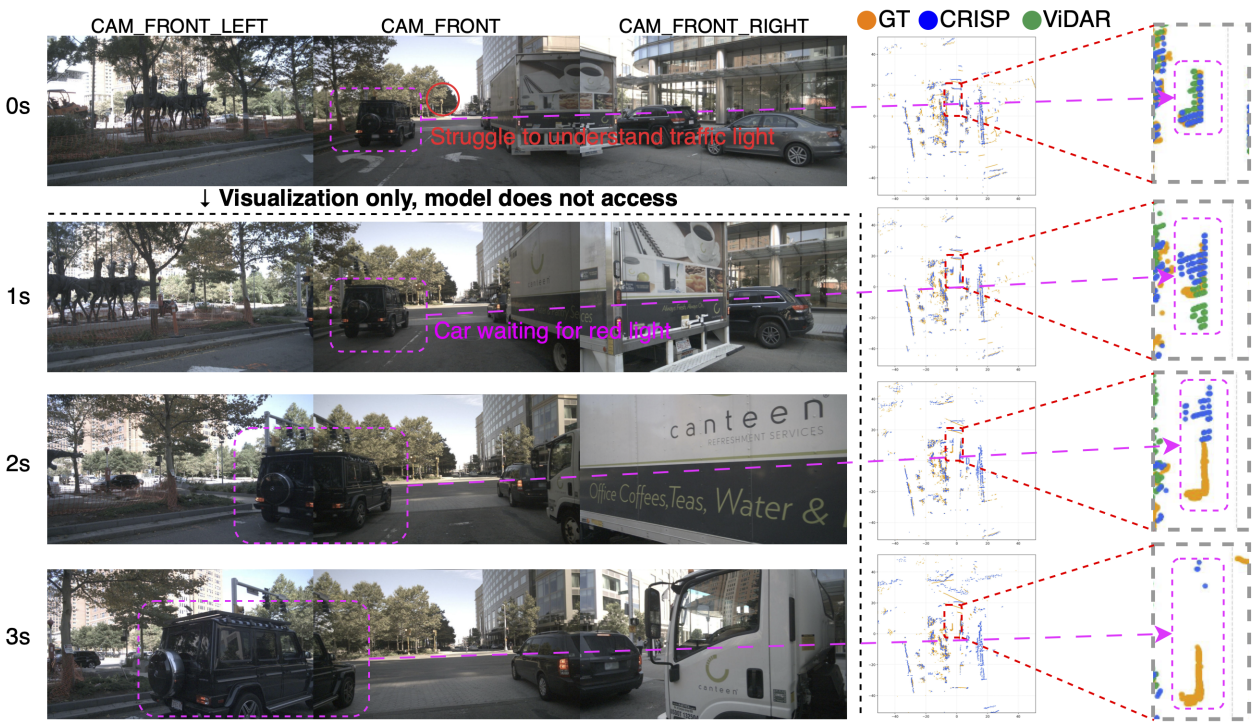



Figure 6.8: Failure case analysis for future world-geometry prediction. We show the front camera views at the current frame (0s) and future timestamps for visualization only; future images are not provided to the model. On the right, predicted point clouds are compared with ground truth. This scene requires reasoning about a traffic light that is small and difficult to observe from the input images: the highlighted vehicle remains stopped for the red light, while the road ahead is visually open. CRISP predicts the vehicle geometry more accurately than the CO ViDAR baseline at early horizons, but its prediction begins to drift after 2s. This suggests that, when high-level traffic-signal intent is ambiguous, the forecasting model may rely more on learned motion priors from pretraining than on the underlying traffic-control state.

Second, CRISP inherits the limitations of radar sensing. Radar provides valuable range and Doppler cues, but it is sparse, noisy, semantically weak, and vulnerable to clutter, multipath, small objects, static objects, calibration errors, and sensor degradation. The proposed gates can reduce unhelpful modality updates, but they do not guarantee calibrated uncertainty or reliable behavior when radar evidence is missing or contradictory. Future CR backbones should therefore reason not only about how to use radar, but also about when radar should be trusted.

Third, the current study is limited by dataset scale and diversity. CRISP is evaluated on nuScenes, one of the few public datasets with synchronized camera, radar, and LiDAR, but it remains far smaller and less diverse than the scale needed for driving foundation models. Cross-dataset pretraining is a natural next step. Large planning datasets such as nuPlan [9]

illustrate the value of scale, but the community still lacks comparably large public datasets with synchronized radar streams. We hope this motivates future releases that include radar together with camera, LiDAR, ego-motion, maps, and planning annotations, enabling CR pretraining at foundation-model scale.

Several research directions follow from these limitations. One is to transfer CRISP-style pretraining to broader architectures, including other BEV encoders, occupancy networks, end-to-end planning systems, and diffusion-based world models. This would help distinguish architecture-specific gains from the general value of CR predictive pretraining. Another direction is to move from deterministic geometric forecasting toward uncertainty-aware and semantics-aware world modeling. Future objectives could combine future geometry with map context, traffic-light states, agent intent, rule annotations, language, and counterfactual ego trajectories, allowing the model to represent multiple plausible futures instead of a single likely one. Such uncertainty is especially important in occluded intersections, unprotected turns, pedestrian interactions, and other safety-critical scenarios.

## 6.7 Conclusion

We presented CRISP, a forecasting-based pretraining framework for transferable camera-radar representation learning in autonomous driving. CRISP uses historical multi-view images and radar sweeps to predict future LiDAR point clouds, using LiDAR only as privileged supervision during pretraining. After pretraining, the forecasting decoder is removed, leaving a practical LiDAR-free CR backbone for downstream inference. By moving CR interaction inside the spatiotemporal BEV encoder, CRISP allows radar cues to guide temporal propagation and multimodal BEV feature formation rather than serving only as a late-fusion signal. CRISP combines an enhanced radar encoder, radar-enhanced temporal self-attention, and multimodal feature rendering with modality innovation gating to jointly encode visual semantics, radar motion cues, BEV geometry, and temporal dynamics. Experiments on nuScenes show that CRISP improves future point cloud forecasting and transfers broadly to 3D detection, tracking, online mapping, motion forecasting, future occupancy prediction, and planning, with especially strong gains on tasks requiring geometry and dynamic scene reasoning. Overall, CRISP shows that CR fusion can be studied as a scalable predictive pretraining problem, and we hope it encourages larger-scale CR pretraining, broader backbone transfer, and semantics- and uncertainty-aware driving world models.

# CHAPTER 7

# Conclusion

## 7.1 Conclusion

Through five key contributions, this dissertation has advanced the state-of-the-art in robotic perception across underwater robotics and autonomous driving. First, **TURTLMap** introduces a real-time, low-cost underwater localization and dense mapping system capable of robust operation in wave-disturbed, low-texture environments. Second, **LiRaFusion** demonstrates how adaptive LiDAR-Radar fusion can improve autonomous driving perception, particularly in challenging long-range and adverse-weather scenarios. Third, **CRKD** addresses the scalability challenge of autonomous driving perception by transferring knowledge from a high-performance LiDAR-Camera (LC) teacher to an efficient Camera-Radar (CR) student, enabling stronger perception under a more deployable sensor configuration. Fourth, **MemFusionMap** improves temporal reasoning for bird's-eye view (BEV) perception by incorporating working memory and adaptive history fusion for road scene understanding. Finally, **CRISP** develops a spatiotemporal CR pretraining framework that learns a transferable 3D foundation backbone by forecasting future LiDAR point clouds from historical camera and radar observations. By integrating radar-enhanced temporal reasoning and multimodal feature rendering, CRISP improves downstream perception, prediction, occupancy forecasting, and planning while requiring only CR inputs during deployment.

Together, these contributions establish a coherent research direction toward scalable and robust robotic perception. Across different platforms and sensing configurations, this dissertation shows how intelligent fusion, knowledge transfer, temporal modeling, and predictive pretraining can help robots build richer representations of complex real-world environments. While improving perception accuracy and robustness, this work emphasizes practical deployment constraints such as sensor cost, scalability, compatibility, and operation under challenging conditions. These principles provide a foundation for future robotic systems that can perceive, predict, and act reliably in dynamic open-world environments.

## 7.2 Future Directions

This dissertation has studied robotic perception under a central theme: how robots can build reliable, scalable, and temporally coherent representations of complex real-world environments from imperfect sensor observations. The proposed systems address this challenge from complementary perspectives, including low-cost underwater localization and mapping, adaptive multimodal fusion, cross-modality knowledge transfer, memory-based temporal reasoning, and predictive CR pretraining. Building on these contributions, future work can move from platform-specific perception systems toward sensor-adaptive perception models that support broader classes of robots, sensing configurations, and environments.

In particular, future work can be organized around three directions. First, unified multimodal-temporal backbones can provide a common representation interface for heterogeneous robot sensors. Second, semantics-aware temporal memory can help robots maintain coherent world models beyond short-term feature aggregation. Finally, these components can support the longer-term goal of general-purpose robot perception across diverse platforms and deployment conditions.

### 7.2.1 Unified Multimodal-Temporal Backbones for Heterogeneous Robot Sensors

A major future direction is to develop unified perception backbones that can support heterogeneous robot sensors across different platforms. The goal is not to require every robot to use the same sensor suite, but to build a sensor-adaptive architecture in which modality-specific encoders or adapters map different sensor inputs into a shared spatial-temporal representation. Such a framework could support sensors such as cameras, radar, LiDAR, sonar, inertial sensors, and other platform-specific measurements, while exposing common representations such as BEV features, 3D occupancy, free space, scene flow, semantic maps, or local geometric maps.

A concrete architecture could consist of three components: sensor-specific adapters, a shared multimodal-temporal backbone, and task-specific prediction heads. The adapters would convert each available sensor modality into a common feature space. The shared backbone would integrate spatial structure and short-horizon temporal cues across the available sensors. The prediction heads would then support downstream tasks such as detection, mapping, localization, prediction, occupancy forecasting, or planning. This structure separates the sensor configuration from the core representation, allowing different robots to reuse the same backbone while changing only the input adapters and task heads required by their deployment setting.

This framework also provides a practical path for transferring perception models from data-rich domains to data-limited robotics domains. For example, autonomous driving datasets could be used to train strong LiDAR-Camera-Radar representations, which could then be adapted or distilled to surface marine robots with similar sensor configurations. In domains where the sensor configuration differs more substantially, such as underwater robotics, the transferable knowledge may come less from the raw sensor measurements and more from the fusion principles: uncertainty-aware weighting, geometric alignment, temporal aggregation, and predictive pretraining. In this way, cross-domain and cross-modal transfer become concrete use cases of a unified sensor-adaptive backbone rather than separate future directions.

Training such a backbone could combine masked sensor modeling, cross-modal alignment, future prediction, and privileged supervision from richer training-time sensors or teacher models. During deployment, however, each robot would rely only on the sensors available on its own platform. This would extend the dissertation's core ideas — multimodal fusion, knowledge transfer, temporal modeling, and predictive pretraining — toward a more general perception framework for robotics.

### 7.2.2 Semantics-Aware Temporal Memory for Dynamic Field Environments

While unified multimodal-temporal backbones address how different sensors can be integrated into a shared representation, robust autonomy also requires deciding what information should persist over time. Short-horizon temporal fusion can improve perception, but robots operating in dynamic and partially observable environments need memory mechanisms that determine what to remember, what to update, and what to discard.

MemFusionMap advances this direction by introducing working memory for BEV perception, while CRISP shows that future-predictive objectives can encourage representations that encode scene dynamics. Future work can build on these ideas by developing memory-augmented perception systems whose read, write, and forgetting operations are guided by uncertainty, semantics, motion, and task relevance. Rather than storing all past features uniformly, the memory should preserve information that remains useful for understanding the scene while suppressing transient artifacts and outdated observations.

A concrete future system could maintain a structured world memory that supports future occupancy prediction, object motion forecasting, free-space reasoning, and planning. Such a memory could be updated from multimodal observations, queried by downstream prediction or planning modules, and trained with objectives that encourage temporal consistency and

future predictability. This would move temporal perception from feature aggregation toward persistent world modeling.

### 7.2.3 Toward General-Purpose Robot Perception

Together, these directions point toward robotic perception systems that are not tied to a single platform, sensor configuration, or task. Unified multimodal-temporal backbones provide a general architecture for integrating heterogeneous robot sensors, while semantics-aware temporal memory provides a mechanism for maintaining coherent world models under uncertainty, occlusion, and dynamic change.

By extending the ideas developed in this dissertation — multimodal fusion, knowledge transfer, temporal memory, and predictive pretraining — future work can move toward general-purpose perception systems that support robust localization, mapping, perception, prediction, and planning across diverse robotic platforms and real-world environments.

# APPENDIX A

# Software

The following appendix describes the open-source software developed throughout this thesis.

Open-source implementations of the proposed algorithms are provided as follows:

- Codebase for OceanSim, supporting underwater robot perception simulation: https://github.com/umfieldrobotics/OceanSim
- Codebase for TURTLMap (Chapter 2): https://github.com/umfieldrobotics/TURTLMap
- Codebase for LiRaFusion (Chapter 3): https://github.com/Song-Jingyu/LiRaFusion
- Codebase for CRKD (Chapter 4): https://github.com/Song-Jingyu/CRKD
- Codebase for MemFusionMap (Chapter 5): https://github.com/Song-Jingyu/MemFusionMap

# BIBLIOGRAPHY


[1] CloudCompare (version 2.11) [GPL software], 2023.

[2] Onur Bagoren, Marc Micatka, Katherine A. Skinner, and Aaron Marburg. PUGS: Perceptual uncertainty for grasp selection in underwater environments. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11278–11285, 2025.

[3] 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 (CVPR)*, pages 1080–1089, 2022.

[4] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438, 2020.

[5] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11682–11692, 2020.

[6] Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, Abdulaziz Almuzairee, Senthil Yogamani, and Fatih Porikli. X-Align: Cross-modal cross-view alignment for bird's-eye-view segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3287–3297, 2023.

[7] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[8] 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 (CVPR)*, pages 11621–11631, 2020.

[9] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

[10] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, 2020.

[12] Spencer Carmichael, Austin Buchan, Mani Ramanagopal, Radhika Ravi, Ram Vasudevan, and Katherine A. Skinner. Dataset and benchmark: Novel sensors for autonomous vehicle perception. *The International Journal of Robotics Research*, 44(3):355–365, 2025.

[13] Anthony Chen, Kevin Zhang, Renrui Zhang, Zihan Wang, Yuheng Lu, Yandong Guo, and Shanghang Zhang. PiMAE: Point cloud and image interactive masked autoencoders for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5291–5301, 2023.

[14] Jiacheng Chen, Ruizhi Deng, and Yasutaka Furukawa. PolyDiffuse: Polygonal shape reconstruction via guided set diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 1863–1888, 2023.

[15] Jiacheng Chen, Yuefan Wu, Jiaqi Tan, Hang Ma, and Yasutaka Furukawa. MapTracker: Tracking with strided memory fusion for consistent vector HD mapping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 90–107, 2024.

[16] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, et al. PersFormer: 3D lane detection via perspective transformer and the OpenLane benchmark. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 550–567, 2022.

[17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[18] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. FUTR3D: A unified sensor fusion framework for 3D detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 172–181, June 2023.

[19] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. BEVDistill: Cross-modal BEV distillation for multi-view 3D object detection. In *International Conference on Learning Representations (ICLR)*, 2023.

[20] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 640–658, 2022.

[21] Zhiyu Chong, Xinzhu Ma, Hong Zhang, Yuxin Yue, Haojie Li, Zhihui Wang, and Wanli Ouyang. MonoDistill: Learning spatial features for monocular 3D object detection. In *International Conference on Learning Representations (ICLR)*, 2022.

[22] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[23] Adam Conner-Simons. To self-drive in the snow, look under the road. MIT News, 2020. Photo courtesy of the researchers.

[24] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020.

[25] Nelson Cowan. The magical mystery four: How is working memory capacity limited, and why? *Current Directions in Psychological Science*, 19(1):51–57, 2010.

[26] Tung Dang, Marco Tranzatto, Shehryar Khattak, Frank Mascarich, Kostas Alexis, and Marco Hutter. Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics*, 37(8):1363–1388, 2020.

[27] Frank Dellaert and GTSAM Contributors. borglab/gtsam, May 2022.

[28] Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1–2):1–139, August 2017.

[29] Wenjie Ding, Limeng Qiao, Xi Qiu, and Chi Zhang. PivotNet: Vectorized pivot learning for end-to-end HD map construction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3672–3682, 2023.

[30] Jason M. Doherty, Clement Belletier, Stephen Rhodes, Agnieszka Jaroslawska, Pierre Barrouillet, Valerie Camos, Nelson Cowan, Moshe Naveh-Benjamin, and Robert H. Logie. Dual-task costs in working memory: An adversarial collaboration. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 45(9):1529, 2019.

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

[32] Yuki Endo, Ehsan Javanmardi, and Shunsuke Kamijo. Analysis of occlusion effects for map-based self-localization in urban areas. *Sensors*, 21(15):5196, 2021.

[33] Ryan M. Eustice, Hanumant Singh, John J. Leonard, and Matthew R. Walter. Visually mapping the RMS titanic: Conservative covariance estimates for SLAM information filters. *The International Journal of Robotics Research*, 25(12):1223–1242, 2006.

[34] Cameron Fabbri, Md Jahidul Islam, and Junaed Sattar. Enhancing underwater imagery using generative adversarial networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7159–7165, 2018.

[35] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

[36] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17062–17070, 2022.

[37] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.

[38] Paul Furgale, Timothy D. Barfoot, and Gabe Sibley. Continuous-time batch estimation using temporal basis functions. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2088–2095, 2012.

[39] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1280–1286, 2013.

[40] 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.

[41] Xunjiang Gu, Guanyu Song, Igor Gilitschenski, Marco Pavone, and Boris Ivanovic. Producing and leveraging online map uncertainty in trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14521–14530, 2024.

[42] Xiaoyang Guo, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. LIGA-Stereo: Learning LiDAR geometry aware representations for stereo-based 3D detector. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3133–3143, 2021.

[43] Chunrui Han, Jinrong Yang, Jianjian Sun, Zheng Ge, Runpei Dong, Hongyu Zhou, Weixin Mao, Yuang Peng, and Xiangyu Zhang. Exploring recurrent long-term temporal fusion for multi-view 3D perception. *IEEE Robotics and Automation Letters*, 9(7):6544–6551, 2024.

[44] Yiduo Hao, Sohrab Madani, Junfeng Guan, Mohammed Alloulah, Saurabh Gupta, and Haitham Hassanieh. Bootstrapping autonomous driving radars with self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15012–15023, 2024.

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

[46] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[48] Georg Hess, Johan Jaxing, Elias Svensson, David Hagerman, Christoffer Petersson, and Lennart Svensson. Masked autoencoder for self-supervised pre-training on LiDAR point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 350–359, 2023.

[49] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[50] Yu Hong, Hang Dai, and Yong Ding. Cross-modality knowledge distillation network for monocular 3D object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–104, 2022.

[51] Haotian Hu, Fanyi Wang, Jingwen Su, Yaonong Wang, Laifeng Hu, Weiye Fang, Jingwei Xu, and Zhiwang Zhang. EA-LSS: Edge-aware lift-splat-shot framework for 3D BEV object detection. *arXiv preprint arXiv:2303.17895*, 2023.

[52] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17853–17862, 2023.

[53] Junjie Huang and Guan Huang. BEVDet4D: Exploit temporal cues in multi-camera 3D object detection. *arXiv preprint arXiv:2203.17054*, 2022.

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

[55] Peixiang Huang, Li Liu, Renrui Zhang, Song Zhang, Xinli Xu, Baichao Wang, and Guoyi Liu. TiG-BEV: Multi-view BEV 3D object detection via target inner-geometry learning. *arXiv preprint arXiv:2212.13979*, 2022.

[56] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. EPNet: Enhancing point features with image semantics for 3D object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–52, 2020.

[57] Jyh-Jing Hwang, Henrik Kretzschmar, Joshua Manela, Sean Rafferty, Nicholas Armstrong-Crews, Tiffany Chen, and Dragomir Anguelov. CramNet: Camera-radar fusion with ray-constrained cross-attention for robust 3D object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 388–405, 2022.

[58] Eduardo Iscar, Katherine A. Skinner, and Matthew Johnson-Roberson. Multi-view 3D reconstruction in underwater environments: Evaluation and benchmark. In *OCEANS 2017 Anchorage*, pages 1–8, 2017.

[59] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[60] Maani Ghaffari Jadidi, William A. Clark, Anthony M. Bloch, Ryan Eustice, and Jessy W. Grizzle. Continuous direct sparse visual odometry from RGB-D images. In *Proceedings of Robotics: Science and Systems (RSS)*, 2019.

[61] Vijay John and Seiichi Mita. RVNet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments. In *Pacific-Rim Symposium on Image and Video Technology*, pages 351–364, 2019.

[62] Matthew Johnson-Roberson, Mitch Bryson, Ariell Friedman, Oscar Pizarro, Giancarlo Troni, Paul Ozog, and Jon C. Henderson. High-resolution underwater robotic vision-based mapping and three-dimensional reconstruction for archaeology. *Journal of Field Robotics*, 34(4):625–643, 2017.

[63] Bharat Joshi, Chanaka Bandara, Ioannis Poulakakis, Herbert G. Tanner, and Ioannis Rekleitis. Hybrid visual inertial odometry for robust underwater estimation. In *OCEANS 2023 MTS/IEEE U.S. Gulf Coast*, pages 1–7, 2023.

[64] Bharat Joshi, Sharmin Rahman, Michail Kalaitzakis, Brennan Cain, James Johnson, Marios Xanthidis, Nare Karapetyan, Alan Hernandez, Alberto Quattrini Li, Nikolaos Vitzilaios, et al. Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7227–7233, 2019.

[65] Bo Ju, Zhikang Zou, Xiaoqing Ye, Minyue Jiang, Xiao Tan, Errui Ding, and Jingdong Wang. Paint and distill: Boosting 3D object detection with semantic passing network. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pages 5639–5648, 2022.

[66] Shehryar Khattak, Christos Papachristos, and Kostas Alexis. Keyframe-based direct thermal–inertial odometry. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3563–3569, 2019.

[67] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point cloud forecasting as a proxy for 4D occupancy forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1116–1124, 2023.

[68] Ayoung Kim and Ryan M. Eustice. Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Transactions on Robotics*, 29(3):719–733, 2013.

[69] Jaekyum Kim, Junho Koh, Yecheol Kim, Jaehyung Choi, Youngbae Hwang, and Jun Won Choi. Robust deep multi-modal learning based on gated information fusion network. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 90–106, 2019.

[70] Jisong Kim, Minjae Seong, Geonho Bang, Dongsuk Kum, and Jun Won Choi. RCM-Fusion: Radar-Camera multi-level fusion for 3D object detection. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 18236–18242, 2024.

[71] Jisong Kim, Minjae Seong, and Jun Won Choi. CRT-Fusion: Camera, radar, temporal fusion using motion information for 3D object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 108625–108648, 2024.

[72] Youngseok Kim, Jun Won Choi, and Dongsuk Kum. GRIF Net: Gated region of interest fusion network for robust 3D object detection from radar point cloud and monocular image. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10857–10864, 2020.

[73] Youngseok Kim, Sanmin Kim, Jun Won Choi, and Dongsuk Kum. Craft: Camera-radar 3D object detection with spatio-contextual fusion transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1160–1168, 2023.

[74] Youngseok Kim, Juyeb Shin, Sanmin Kim, In-Jae Lee, Jun Won Choi, and Dongsuk Kum. CRN: Camera radar net for accurate, robust, efficient 3D perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17569–17580, 2023.

[75] James C. Kinsey, Ryan M. Eustice, and Louis L. Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference on Manoeuvring and Control of Marine Craft*, volume 88, pages 1–12, 2006.

[76] Marvin Klingner, Shubhankar Borse, Varun Ravi Kumar, Behnaz Rezaei, Venkatraman Narayanan, Senthil Yogamani, and Fatih Porikli. X3KD: Knowledge distillation across modalities, tasks and stages for multi-camera 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13343–13353, 2023.

[77] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, 2012.

[78] Mathieu Labbé and François Michaud. RTAB-Map as an open-source LiDAR and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019.

[79] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, 2019.

[80] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[81] Kai Lei, Zhan Chen, Shuman Jia, and Xiaoteng Zhang. HVDetFusion: A simple and robust camera-radar fusion framework. *arXiv preprint arXiv:2307.11323*, 2023.

[82] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[83] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M. Ni, and Lei Zhang. DN-DETR: Accelerate DETR training by introducing query DeNoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13619–13627, 2022.

[84] Jianing Li, Ming Lu, Jiaming Liu, Yandong Guo, Yuan Du, Li Du, and Shanghang Zhang. BEV-LGKD: A unified LiDAR-guided knowledge distillation framework for multi-view BEV 3D object detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 2489–2498, 2023.

[85] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. HDMapNet: An online HD map construction and evaluation framework. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4628–4634, 2022.

[86] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[87] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3D object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[88] Yanwei Li, Xiaojuan Qi, Yukang Chen, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Voxel field fusion for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1119, 2022.

[89] Yao Li, Jiajun Deng, Yu Zhang, Jianmin Ji, Houqiang Li, and Yanyong Zhang. ezfusion: A close look at the integration of LiDAR, millimeter-wave radar, and camera for accurate 3D object detection and tracking. *IEEE Robotics and Automation Letters*, 7(4):11182–11189, 2022.

[90] 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 (AAAI)*, volume 37, pages 1477–1485, 2023.

[91] Yu-Jhe Li, Jinhyung Park, Matthew O'Toole, and Kris Kitani. Modality-agnostic learning for radar-LiDAR fusion in vehicle detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 908–917, June 2022.

[92] Zhihui Li, Pengfei Xu, Xiaojun Chang, Luyao Yang, Yuanyuan Zhang, Lina Yao, and Xiaojiang Chen. When object detection meets knowledge distillation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10555–10579, 2023.

[93] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–18, 2022.

[94] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from LiDAR-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):2020–2036, 2025.

[95] Hanxue Liang, Chenhan Jiang, Dapeng Feng, Xin Chen, Hang Xu, Xiaodan Liang, Wei Zhang, Zhenguo Li, and Luc Van Gool. Exploring geometry-aware contrast and clustering harmonization for self-supervised 3D object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3293–3302, 2021.

[96] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[97] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. MapTR: Structured modeling and learning for online vectorized HD map construction. In *International Conference on Learning Representations (ICLR)*, 2023.

[98] Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. MapTRv2: An end-to-end framework for online

vectorized HD map construction. *International Journal of Computer Vision*, pages 1–23, 2024.

[99] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, July 2017.

[100] Tzu-Yuan Lin, Tingjun Li, Wenzhe Tong, and Maani Ghaffari. Proprioceptive invariant robot state estimation. *arXiv preprint arXiv:2311.04320*, 2023.

[101] Zhiwei Lin, Zhe Liu, Zhongyu Xia, Xinhao Wang, Yongtao Wang, Shengxiang Qi, Yang Dong, Nan Dong, Le Zhang, and Ce Zhu. RCBEVDet: Radar-camera fusion in bird's eye view for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14928–14937, 2024.

[102] Zhiwei Lin, Yongtao Wang, Shengxiang Qi, Nan Dong, and Ming-Hsuan Yang. BEV-MAE: Bird's eye view masked autoencoders for point cloud pre-training in autonomous driving scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 3531–3539, 2024.

[103] Xiaolu Liu, Song Wang, Wentong Li, Ruizi Yang, Junbo Chen, and Jianke Zhu. MGMap: Mask-guided learning for online vectorized HD map construction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14812–14821, 2024.

[104] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. VectorMapNet: End-to-end vectorized HD map learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 22352–22369, 2023.

[105] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

[106] Zhe Liu, Xiaoqing Ye, Xiao Tan, Errui Ding, and Xiang Bai. StereoDistill: Pick the cream from LiDAR for distilling stereo-based 3D object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1790–1798, 2023.

[107] 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, 2023.

[108] Zihao Liu, Xiaoyu Zhang, Guangwei Liu, Ji Zhao, and Ningyi Xu. Leveraging enhanced queries of point sets for vectorized map construction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 461–477, 2024.

[109] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[110] Yukai Ma, Jianbiao Mei, Xuemeng Yang, Licheng Wen, Weihua Xu, Jiangning Zhang, Xingxing Zuo, Botian Shi, and Yong Liu. LiCROcc: Teach radar for accurate semantic occupancy prediction using LiDAR and camera. *IEEE Robotics and Automation Letters*, 10(1):852–859, 2025.

[111] Anas Mahmoud, Jordan S. K. Hu, and Steven L. Waslander. Dense voxel fusion for 3D object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 663–672, 2023.

[112] Angelos Mallios, Eduard Vidal, Ricard Campos, and Marc Carreras. Underwater caves sonar data set. *The International Journal of Robotics Research*, 36(12):1247–1251, 2017.

[113] Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. BEV-guided multi-modality fusion for driving perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21960–21969, 2023.

[114] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3D object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, 131(8):1909–1963, 2023.

[115] Tamas Matuszka, Ivan Barton, Ádám Butykai, Péter Hajas, Dávid Kiss, Domonkos Kovács, Sándor Kunsági-Máté, Péter Lengyel, Gábor Németh, Levente Pető, Dezső Ribli, Dávid Szeghy, Szabolcs Vajna, and Balint Viktor Varga. aiMotive dataset: A multimodal dataset for robust autonomous driving with long-range perception. In *International Conference on Learning Representations (ICLR) 2023 Workshop on Scene Representations for Autonomous Driving*, 2023.

[116] Jérôme Maye, Paul Furgale, and Roland Siegwart. Self-supervised calibration for robotic systems. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 473–480, 2013.

[117] Chen Min, Dawei Zhao, Liang Xiao, Yiming Nie, and Bin Dai. UniWorld: Autonomous driving pre-training via world models. *arXiv preprint arXiv:2308.07234*, 2023.

[118] Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, Liping Jing, Yiming Nie, and Bin Dai. DriveWorld: 4D pre-trained scene understanding via world models for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15522–15533, 2024.

[119] Mohammadreza Mostajabi, Ching Ming Wang, Darsh Ranjan, and Gilbert Hsyu. High-resolution radar dataset for semi-supervised learning of dynamic objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 450–457, June 2020.

[120] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[121] Ramin Nabati and Hairong Qi. CenterFusion: Center-based radar and camera fusion for 3D object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1527–1536, 2021.

[122] Felix Nobis, Ehsan Shafiei, Phillip Karle, Johannes Betz, and Markus Lienkamp. Radar voxel fusion for 3D object detection. *Applied Sciences*, 11(12):5598, 2021.

[123] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1366–1373, 2017.

[124] Kenton Oliver, Weilin Hou, and Song Wang. Image feature detection and matching in underwater conditions. In *Ocean Sensing and Monitoring II*, volume 7678, pages 185–196, 2010.

[125] Open Robotics. Robot Operating System (ROS), 2020.

[126] Anshul Paigwar, David Sierra-Gonzalez, Ozgur Erkent, and Christian Laugier. Frustum-PointPillars: A multi-stage approach for 3D object detection using RGB camera and LiDAR. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2926–2935, 2021.

[127] Edoardo Palladin, Samuel Brucker, Filippo Ghilotti, Praveen Narayanan, Mario Bijelic, and Felix Heide. Self-supervised sparse sensor fusion for long range perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27498–27509, 2025.

[128] Su Pang, Daniel Morris, and Hayder Radha. CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10386–10393, 2020.

[129] Su Pang, Daniel Morris, and Hayder Radha. Fast-CLOCs: Fast camera-LiDAR object candidates fusion for 3D object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 187–196, 2022.

[130] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 604–621, 2022.

[131] Ziqi Pang, Jie Li, Pavel Tokmakov, Dian Chen, Sergey Zagoruyko, and Yu-Xiong Wang. Standing between past and future: Spatio-temporal modeling for multi-camera 3D multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17928–17938, 2023.

[132] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-LiDAR needed for monocular 3D object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3142–3152, 2021.

[133] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris M. Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3D object detection. In *International Conference on Learning Representations (ICLR)*, 2022.

[134] Scott Drew Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Hong Eng, Daniela Rus, and Marcelo H. Ang. Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1):6, 2017.

[135] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 194–210, 2020.

[136] Easton R. Potokar, Kalin Norman, and Joshua G. Mangelson. Invariant extended kalman filtering for underwater navigation. *IEEE Robotics and Automation Letters*, 6(3):5792–5799, 2021.

[137] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. Robust multimodal vehicle detection in foggy weather using complementary LiDAR and radar signals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 444–453, June 2021.

[138] Rui Qian, Xin Lai, and Xirong Li. 3D object detection for autonomous driving: a survey. *Pattern Recognition*, page 108796, 2022.

[139] Donghao Qiao and Farhana Zulkernine. Adaptive feature fusion for cooperative perception using LiDAR point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1186–1195, January 2023.

[140] Limeng Qiao, Wenjie Ding, Xi Qiu, and Chi Zhang. End-to-end vectorized HD-map construction with piecewise bezier curve. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13218–13228, 2023.

[141] Hongmao Qin, Xi Wang, Guangcai Wang, Manjiang Hu, Yougang Bian, Xiaohui Qin, and Rongjun Ding. A novel INS/USBL/DVL integrated navigation scheme against complex underwater environment. *Ocean Engineering*, 286:115485, 2023.

[142] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[143] Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[144] Sharmin Rahman, Alberto Quattrini Li, and Ioannis Rekleitis. SVIn2: An underwater SLAM system using sonar, visual, inertial, and depth sensor. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1861–1868, 2019.

[145] Sharmin Rahman, Alberto Quattrini Li, and Ioannis Rekleitis. SVIn2: A multi-sensor fusion-based underwater SLAM system. *The International Journal of Robotics Research*, 41(11-12):1022–1042, 2022.

[146] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11135–11144, 2020.

[147] David M. Rosen, Kevin J. Doherty, Antonio Terán Espinoza, and John J. Leonard. Advances in inference and representation for simultaneous localization and mapping. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):215–242, 2021.

[148] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696, 2020.

[149] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-LiDAR self-supervised distillation for autonomous driving data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9881–9891, 2022.

[150] Jonas Schramm, Niclas Vödisch, Kürsat Petek, B. Ravi Kiran, Senthil Yogamani, Wolfram Burgard, and Abhinav Valada. BEVCar: Camera-radar fusion for BEV map and object segmentation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1435–1442, 2024.

[151] Advaith V. Sethuraman, Anja Sheppard, Onur Bagoren, Christopher Pinnow, Jamey Anderson, Timothy C. Havens, and Katherine A. Skinner. Machine learning for shipwreck segmentation from side scan sonar imagery: Dataset and benchmark. *The International Journal of Robotics Research*, 44(3):341–354, 2025.

[152] Advaith Venkatramanan Sethuraman, Manikandasriram Srinivasan Ramanagopal, and Katherine A. Skinner. WaterNeRF: Neural radiance fields for underwater scenes. In *OCEANS 2023 MTS/IEEE U.S. Gulf Coast*, pages 1–7, 2023.

[153] Tixiao Shan and Brendan Englot. LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765, 2018.

[154] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. RADIATE: A radar dataset for automotive perception in bad weather. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2021.

[155] Kun Shi, Shibo He, Zhenyu Shi, Anjun Chen, Zehui Xiong, Jiming Chen, and Jun Luo. Radar and camera fusion for object detection and tracking: A comprehensive survey. *arXiv preprint arXiv:2410.19872*, 2024.

[156] Vishwanath A. Sindagi, Yin Zhou, and Oncel Tuzel. MVX-Net: Multimodal voxelnet for 3D object detection. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7276–7282, 2019.

[157] Merrill I. Skolnik. *Radar handbook*. 2008.

[158] Jingyu Song, Onur Bagoren, and Katherine A. Skinner. Uncertainty-aware acoustic localization and mapping for underwater robots. In *OCEANS 2023 Limerick*, pages 1–9, 2023.

[159] Jingyu Song, Lingjun Zhao, and Katherine A. Skinner. LiRaFusion: Deep adaptive LiDAR-radar fusion for 3D object detection. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 18250–18257, 2024.

[160] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020.

[161] Alexander Thoms, Gabriel Earle, Nicholas Charron, and Sriram Narasimhan. Tightly coupled, graph-based DVL/IMU fusion and decoupled mapping for SLAM-centric maritime infrastructure inspection. *IEEE Journal of Oceanic Engineering*, 48(3):663–676, 2023.

[162] Ignacio Torroba, Marco Cella, Aldo Terán, Niklas Rolleberg, and John Folkesson. Online stochastic variational gaussian process mapping for large-scale bathymetric SLAM in real time. *IEEE Robotics and Automation Letters*, 8(6):3150–3157, 2023.

[163] Michael Ulrich, Sascha Braun, Daniel Köhler, Daniel Niederlöhner, Florian Faion, Claudius Gläser, and Holger Blume. Improved orientation estimation and detection with hybrid object detection networks for automotive radar. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 111–117, 2022.

[164] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[165] 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 (CVPR)*, pages 4603–4611, June 2020.

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

[167] Leichen Wang, Tianbai Chen, Carsten Anklam, and Bastian Goldluecke. High dimensional frustum PointNet for 3D object detection from camera, LiDAR, and radar. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1621–1628, 2020.

[168] LeiChen Wang and Bastian Goldluecke. Sparse-PointNet: See further in autonomous vehicles. *IEEE Robotics and Automation Letters*, 6(4):7049–7056, 2021.

[169] Li Wang, Xinyu Zhang, Ziying Song, Jiangfeng Bi, Guoxin Zhang, Haiyue Wei, Liyao Tang, Lei Yang, Jun Li, Caiyan Jia, and Lijun Zhao. Multi-modal 3D object detection in autonomous driving: A survey and taxonomy. *IEEE Transactions on Intelligent Vehicles*, 8(7):3781–3798, 2023.

[170] Ruihao Wang, Jian Qin, Kaiying Li, Yaochen Li, Dong Cao, and Jintao Xu. BEV-LaneDet: An efficient 3D lane detection based on virtual camera via key-points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1002–1011, 2023.

[171] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3D object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3621–3631, 2023.

[172] Shuo Wang, Fan Jia, Weixin Mao, Yingfei Liu, Yucheng Zhao, Zehui Chen, Tiancai Wang, Chi Zhang, Xiangyu Zhang, and Feng Zhao. Stream query denoising for vectorized HD-map construction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 203–220, 2024.

[173] Song Wang, Lingdong Kong, Xiaolu Liu, Hao Shi, Wentong Li, Jianke Zhu, and Steven C. H. Hoi. Forging spatial intelligence: A roadmap of multi-modal data pre-training for autonomous systems. *arXiv preprint arXiv:2512.24385*, 2025.

[174] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3D object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 913–922, 2021.

[175] Weihan Wang, Bharat Joshi, Nathaniel Burgdorfer, Konstantinos Batsos, Alberto Quattrini Li, Philippos Mordohai, and Ioannis Rekleitis. Real-time dense 3D mapping of underwater environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5184–5191, 2023.

[176] Yingjie Wang, Qiuyu Mao, Hanqi Zhu, Jiajun Deng, Yu Zhang, Jianmin Ji, Houqiang Li, and Yanyong Zhang. Multi-modal 3D object detection in autonomous driving: a survey. *International Journal of Computer Vision*, pages 1–31, 2023.

[177] Yizhou Wang, Zhongyu Jiang, Yudong Li, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. RODNet: A real-time radar object detection network cross-supervised by camera-radar fused object 3D localization. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):954–967, 2021.

[178] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin M. Solomon. DETR3D: 3D object detection from multi-view images via 3D-to-2D queries. In *Conference on Robot Learning (CoRL)*, pages 180–191, 2021.

[179] Yue Wang and Justin M. Solomon. Object DGCNN: 3D object detection using dynamic graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[180] Zeyu Wang, Dingwen Li, Chenxu Luo, Cihang Xie, and Xiaodong Yang. DistillBEV: Boosting multi-camera 3D object detection with cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8603–8612, 2023.

[181] Zhangjing Wang, Yu Wu, and Qingqing Niu. Multi-sensor fusion in automated driving: A survey. *IEEE Access*, 8:2847–2868, 2020.

[182] Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. LiDAR distillation: Bridging the beam-induced domain gap for 3D object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 179–195, 2022.

[183] Li-Hua Wen and Kang-Hyun Jo. Deep learning-based perception systems for autonomous driving: A comprehensive survey. *Neurocomputing*, 489:255–270, 2022.

[184] Stefan B. Williams, Oscar R. Pizarro, Michael V. Jakuba, Craig R. Johnson, Neville S. Barrett, Russell C. Babcock, Gary A. Kendrick, Peter D. Steinberg, Andrew J. Heyward, Peter J. Doherty, Ian Mahon, Matthew Johnson-Roberson, Daniel Steinberg, and Ariell Friedman. Monitoring of benthic reference sites: Using an autonomous underwater vehicle. *IEEE Robotics and Automation Magazine*, 19(1):73–84, 2012.

[185] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[186] Joey Wilson, Yuewei Fu, Arthur Zhang, Jingyu Song, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. Convolutional bayesian kernel inference for 3D semantic mapping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8364–8370, 2023.

[187] Joey Wilson, Jingyu Song, Yuewei Fu, Arthur Zhang, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. MotionSC: Data set and network for real-time semantic mapping in dynamic environments. *IEEE Robotics and Automation Letters*, 7(3):8439–8446, 2022.

[188] David Wisth, Marco Camurri, and Maurice Fallon. VILENS: Visual, inertial, LiDAR, and leg odometry for all-terrain legged robots. *IEEE Transactions on Robotics*, 39(1):309–326, 2023.

[189] Philipp Wolters, Johannes Gilg, Torben Teepe, and Gerhard Rigoll. SpaRC-AD: A baseline for radar-camera fusion in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1831–1841, October 2025.

[190] Zizhang Wu, Guilian Chen, Yuanzhu Gan, Lei Wang, and Jian Pu. MVFusion: Multi-view 3D object detection with semantic-aligned radar and camera fusion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2766–2773, 2023.

[191] Xinyu Xie and Jarrod Moss. Working memory capacity predicts more effective problem space exploration. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44, 2022.

[192] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. PointFusion: Deep sensor fusion for 3D bounding box estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253, June 2018.

[193] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Zhou Bin, and Liangjun Zhang. FusionPainting: Multimodal fusion with adaptive attention for 3D object detection. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 3047–3054, 2021.

[194] Shida Xu, Tomasz Luczynski, Jonatan Scharff Willners, Ziyang Hong, Kaicheng Zhang, Yvan R. Petillot, and Sen Wang. Underwater visual acoustic SLAM with extrinsic calibration. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7647–7652, 2021.

[195] Zhenhua Xu, Kwan-Yee K. Wong, and Hengshuang Zhao. InsMapper: Exploring inner-instance information for vectorized HD mapping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 296–312, 2024.

[196] Zirui Xu, George P. Kontoudis, and Kyriakos G. Vamvoudakis. Online and robust intermittent motion planning in dynamic and changing environments. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):17425–17439, 2024.

[197] 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 (ICCV)*, pages 18222–18232, 2023.

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

[199] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. RadarNet: Exploiting radar for robust perception of dynamic objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 496–512, 2020.

[200] Honghui Yang, Tong He, Jiaheng Liu, Hua Chen, Boxi Wu, Binbin Lin, Xiaofei He, and Wanli Ouyang. GD-MAE: generative decoder for MAE pre-training on LiDAR point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9403–9414, 2023.

[201] Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, Shixiang Tang, Hengshuang Zhao, Qibo Qiu, Binbin Lin, et al. UniPAD: A universal pre-training paradigm for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15238–15250, 2024.

[202] Jihan Yang, Shaoshuai Shi, Runyu Ding, Zhe Wang, and Xiaojuan Qi. Towards efficient 3D object detection with knowledge distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[203] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14673–14684, 2024.

[204] Zeyu Yang, Jiaqi Chen, Zhenwei Miao, Wei Li, Xiatian Zhu, and Li Zhang. DeepInteraction: 3D object detection via modality interaction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[205] Shanliang Yao, Runwei Guan, Xiaoyu Huang, Zhuoxiao Li, Xiangyu Sha, Yong Yue, Eng Gee Lim, Hyungjoon Seo, Ka Lok Man, Xiaohui Zhu, and Yutao Yue. Radar-camera fusion for object detection and semantic segmentation in autonomous driving: A comprehensive review. *IEEE Transactions on Intelligent Vehicles*, 9(1):2094–2128, 2024.

[206] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3D object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, June 2021.

[207] Jin Hyeok Yoo, Yecheol Kim, Jisong Kim, and Jun Won Choi. 3D-CVF: Generating joint camera and LiDAR features using cross-view spatial feature fusion for 3D object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2020.

[208] Xihang Yu, Sangli Teng, Theodor Chakhachiro, Wenzhe Tong, Tingjun Li, Tzu-Yuan Lin, Sarah Koehler, Manuel Ahumada, Jeffrey M. Walls, and Maani Ghaffari. Fully proprioceptive slip-velocity-aware state estimation for mobile robots via invariant kalman filtering and disturbance observer. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8096–8103, 2023.

[209] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-training 3D point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19313–19322, 2022.

[210] Tianyuan Yuan, Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. StreamMapNet: Streaming mapping network for vectorized online HD map construction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7356–7365, 2024.

[211] Jia Zeng, Li Chen, Hanming Deng, Lewei Lu, Junchi Yan, Yu Qiao, and Hongyang Li. Distilling focal knowledge from imperfect expert for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 992–1001, 2023.

[212] Arthur Zhang, Chaitanya Eranki, Christina Zhang, Ji-Hwan Park, Raymond Hong, Pranav Kalyani, Lochana Kalyanaraman, Arsh Gamare, Arnav Bagad, Maria Esteva, and Joydeep Biswas. Toward robust robot 3-D perception in urban environments: The UT campus object dataset. *IEEE Transactions on Robotics*, 40:3322–3340, 2024.

[213] Haiming Zhang, Wending Zhou, Yiyao Zhu, Xu Yan, Jiantao Gao, Dongfeng Bai, Yingjie Cai, Bingbing Liu, Shuguang Cui, and Zhen Li. VisionPAD: A vision-centric pre-training paradigm for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17165–17175, 2025.

[214] Ji Zhang, Sanjiv Singh, et al. LOAM: LiDAR odometry and mapping in real-time. In *Proceedings of Robotics: Science and Systems (RSS)*, volume 2, pages 1–9, 2014.

[215] Linfeng Zhang, Runpei Dong, Hung-Shuo Tai, and Kaisheng Ma. PointDistiller: Structured knowledge distillation towards efficient and compact 3D detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21791–21801, 2023.

[216] Linfeng Zhang, Yukang Shi, Hung-Shuo Tai, Zhipeng Zhang, Yuan He, Ke Wang, and Kaisheng Ma. Structured knowledge distillation towards efficient and compact multi-view 3D detection. *arXiv preprint arXiv:2211.08398*, 2022.

[217] Ray Zhang, Tzu-Yuan Lin, Chien Erh Lin, Steven A. Parkison, William Clark, Jessy W. Grizzle, Ryan M. Eustice, and Maani Ghaffari. A new framework for registration of semantic point clouds from stereo and RGB-D cameras. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12214–12221, 2021.

[218] Tianyi Zhang and Matthew Johnson-Roberson. Beyond NeRF underwater: Learning neural reflectance fields for true color correction of marine imagery. *IEEE Robotics and Automation Letters*, 8(10):6467–6474, 2023.

[219] Tianyi Zhang, Weiming Zhi, Braden Meyers, Nelson Durrant, Kaining Huang, Joshua Mangelson, Corina Barbalata, and Matthew Johnson-Roberson. RecGS: Removing water caustic with recurrent gaussian splatting. *IEEE Robotics and Automation Letters*, 10(1):668–675, 2025.

[220] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7244–7251, 2018.

[221] Haimei Zhao, Qiming Zhang, Shanshan Zhao, Jing Zhang, and Dacheng Tao. BEVSimDet: Simulated multi-modal distillation in bird's-eye view for multi-view 3D object detection. *arXiv preprint arXiv:2303.16818*, 2023.

[222] Lin Zhao, Mingxi Zhou, and Brice Loose. Tightly-coupled visual-DVL-inertial odometry for robot-based ice-water boundary exploration. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7127–7134, 2023.

[223] Lingjun Zhao, Jingyu Song, and Katherine A. Skinner. CRKD: Enhanced camera-radar object detection with cross-modality knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15470–15480, 2024.

[224] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 30:5264–5276, 2021.

[225] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13760–13769, 2022.

[226] Junbao Zhou, Ziqi Pang, and Yu-Xiong Wang. RMem: Restricted memory banks improve video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18602–18611, 2024.

[227] Shengchao Zhou, Weizhou Liu, Chen Hu, Shuchang Zhou, and Chao Ma. UniDistill: A universal cross-modality knowledge distillation framework for 3D object detection in bird's-eye view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5116–5125, 2023.

[228] Taohua Zhou, Junjie Chen, Yining Shi, Kun Jiang, Mengmeng Yang, and Diange Yang. Bridging the view disparity between radar and camera features for multi-modal fusion 3D object detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1523–1535, 2023.

[229] Xin Zhou, Dingkang Liang, Sifan Tu, Xiwu Chen, Yikang Ding, Dingyuan Zhang, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. HERMES: A unified self-driving world model for simultaneous 3D scene understanding and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27817–27827, 2025.

[230] Yi Zhou, Lulu Liu, Haocheng Zhao, Miguel López-Benítez, Limin Yu, and Yutao Yue. Towards deep radar perception for autonomous driving: Datasets, methods, and challenges. *Sensors*, 22(11):4208, 2022.

[231] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018.