SSMRadNet : A Sample-wise State-Space Framework for Efficient and Ultra-Light Radar Segmentation and Object Detection

Anuvab Sen, Mir Sayeed Mohammad, Saibal Mukhopadhyay
Georgia Institute of Technology, Atlanta, Georgia, USA
asen74@gatech.edu, mirsayeedmohammad@gatech.edu, saibal.mukhopadhyay@ece.gatech.edu

Abstract

We introduce SSMRadNet, the first multi-scale State Space Model (SSM) based detector for Frequency Modulated Continuous Wave (FMCW) radar that sequentially processes raw ADC samples through two SSMs. One SSM learns a chirp-wise feature by sequentially processing samples from all receiver channels within one chirp, and a second SSM learns a representation of a frame by sequentially processing chirp-wise features. The latent representations of a radar frame are decoded to perform segmentation and detection tasks. Comprehensive evaluations on the RADIal dataset show SSMRadNet has ***10-33× fewer parameters*** *and* ***60-88× less computation (GFLOPs)*** *while being* ***3.7× faster*** *than state-of-the-art transformer and convolution-based radar detectors at competitive performance for segmentation tasks.*

1. Introduction

Frequency Modulated Continuous Wave (FMCW) radars improve the perception of autonomous systems under adverse conditions [16, 34]. A FMCW radar frame is represented by a $C \times S \times N_{R_x}$ tensor of digitized samples (analog-to-digital-converters (ADC) cube), where C, N_{R_x}, and S are the number of chirps per frame, ADCs (receiver channels), and samples per chirp, respectively. Higher N_{R_x}, C, and sampling bandwidth (more S per chirp) improve angular, velocity, and range resolutions, respectively, but also increase computational demand on perception algorithms.

Traditionally, radar perception models generate sparse point clouds from a 3D Range-Azimuth-Doppler (RAD) tensor, which are computed by applying multi-stage FFTs on ADC cubes. The point clouds are processed by Convolutional Neural Networks (CNN) for end tasks (Figure 1,Table 1) [12, 16, 18, 20–22, 25, 25, 32]. However, creation and processing of point clouds introduce computation and latency bottlenecks.

To address the point cloud challenges, recent works have demonstrated convolution, attention, and recurrent network based approaches to learn features directly from ADC cube (Table 1) [1, 5, 11, 13, 28, 30, 32, 35]. As an example, Rebut et al. pioneered FFT-RadNet[17] that avoids computing the full 3D range-azimuth-Doppler tensor by using a CNN that learns to recover object angles from the 2D range-Doppler input, thus eliminating one FFT stage and reducing computation. As an alternative, ADCNet [33] learns representations directly from the ADC cube using CNN based approaches further reducing computation. However, parameter, computation, and memory complexity of 3D CNN increases non-linearly with dimension of the ADC cube. T-FFT-RadNet family [5, 11] proposed transformer based methods that interpret samples within the ADC cube as tokens and replace CNNs with attention. The design shows improved performance (accuracy), but as the computational cost of self-attention scales quadratically, processing long radar sample-by-chirp sequences is computationally expensive (Table 1).

ChirpNet pioneered a sequential method where digitized ADC samples from all receivers for a chirp are encoded into a feature vector, and features across chirps are learned using a recurrent network [24]. The design shows lower computation, parameters, and latency than prior methods, but with a lower perception accuracy (Table 1).

To further reduce computation while maintaining accuracy, we propose *SSMRadNet*, a novel radar processing framework that models (synchronous) samples from all receivers (N_{Rx} antennas) as sequential tokens and processes using a multi-scale state-space model (SSM)(Figure 1). The SSM can capture long-duration dependencies, and its computation scales linearly with sequence length, instead of quadratically, as in transformers [4, 6, 7, 10, 14]. Although, SSM shows superior performance than transformers on vision tasks[6, 7], SSM for radar processing have not been demonstrated yet.

SSMRadNet has one SSM (i.e., states) to process ADC

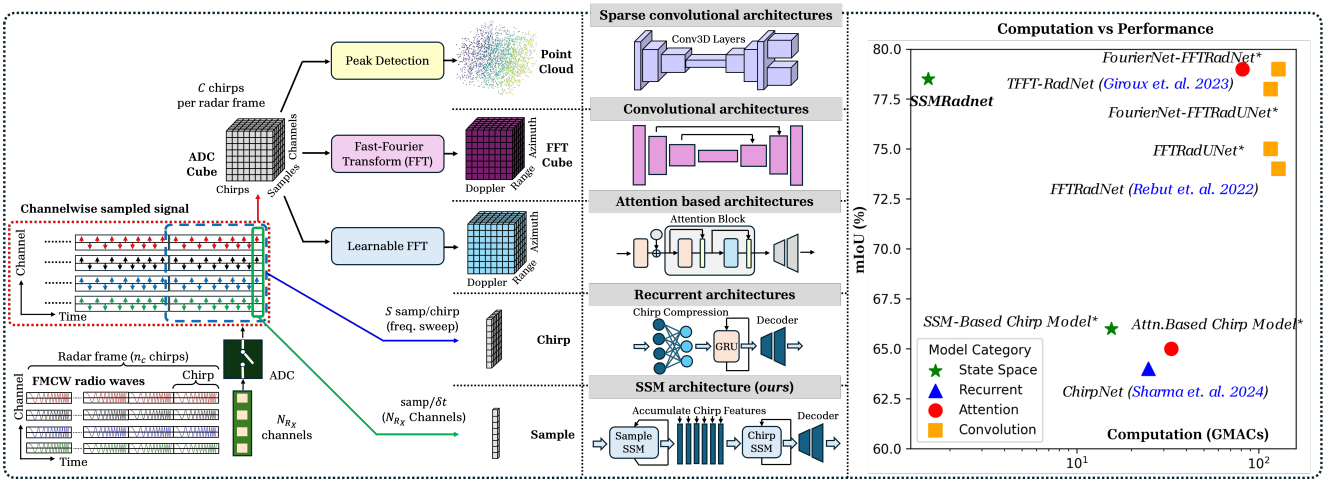

Figure 1. **Motivation from Prior Works:** Traditional convolutional[17], attention-based[5] and recurrent[24] networks are computationally expensive because they rely on a large volume of data (*left*). Our state space modeling approach scales linearly in computation with increased radar resolution ($C \times S \times N_{R_X}$) because of our sample by sample processing, achieving state of the art performance for a fraction of the computation (*right*). (*hybrid models; refer Table 4).

Table 1. Radar object detection architectures; Trade-offs: ✓ (desirable), ✗ (undesirable), • (moderate).

Class	Model	Input Type	Feature Shape	Processing Core	Params (M)	Computation (GMACs)	Latency
Convolution	Point-cloud Conv Detectors [31]	Point Cloud (CFAR)	Sparse 3D tensor	CNN / FPN over sparse voxel / BEV grids	High ✗	High ✗	High ✗
	FFT-RadNet [17]	Range–Doppler (RD)	2D tensor	CNN encoder + FPN angle regression	High ✗	High ✗	High ✗
	ADCNet (Conv) [33]	Raw ADC → RD	2D tensor	Learnable signal proc + Conv blocks	High ✗	High ✗	High ✗
Attention	T-FFT-RadNet (Swin) [5, 11]	RD / RAD / raw ADC	2D / 3D tensor	Hierarchical Swin self attention + heads	High ✗	High ✗ (quadratic)	High ✗
	RadarFormer [2]	RA / RD (merged)	2D tensor	Channel–chirp–time merging transformer	High ✗	High ✗	High ✗
Recurrent	ChirpNet [24]	Raw ADC chirp / (R_x)	2D tensor	Sequential GRU + lightweight MLP	Mid •	Mid •	Mid •
SSM	SSMRadNet (ours)	Raw ADC sample / chirp	1D tensor	Sample- & Chirp-wise SSM's	Low ✓	Low ✓	Low ✓

samples within a chirp (the fast-time axis). The learned representations from successive chirps are processed sequentially using a second SSM (the slow-time axis) to create a latent map for a frame. The final latent map is decoded for downstream detection or segmentation.

We demonstrate SSMRadNet on two radar vision datasets, namely, RADIAL & RADICAL. As illustrated in Figure 1, for the segmentation task on RADIal, *SSMRadNet* demonstrates more than $60\times$ reduction in computation but at a competitive performance. The computational advantage originates from eliminating FFT processing, 3D convolution, and quadratic self-attention bottleneck in transformers. Compared to ChirpNet, *SSMRadNet*, shows more than $15\times$ computation reduction in computation with a substantially better accuracy, thanks to sequentially learning the temporal features between successive ADC samples within a chirp.

The key contributions of this paper are as follows :
(1) SSM for Radar Processing : Our method is the first multi-scale SSM framework for radar processing framework to learn features from sequentially obtained ADC samples modeled as tokens.
(2) Sample-by-sample radar detection: We propose the first sample-by-sample radar data processing architecture

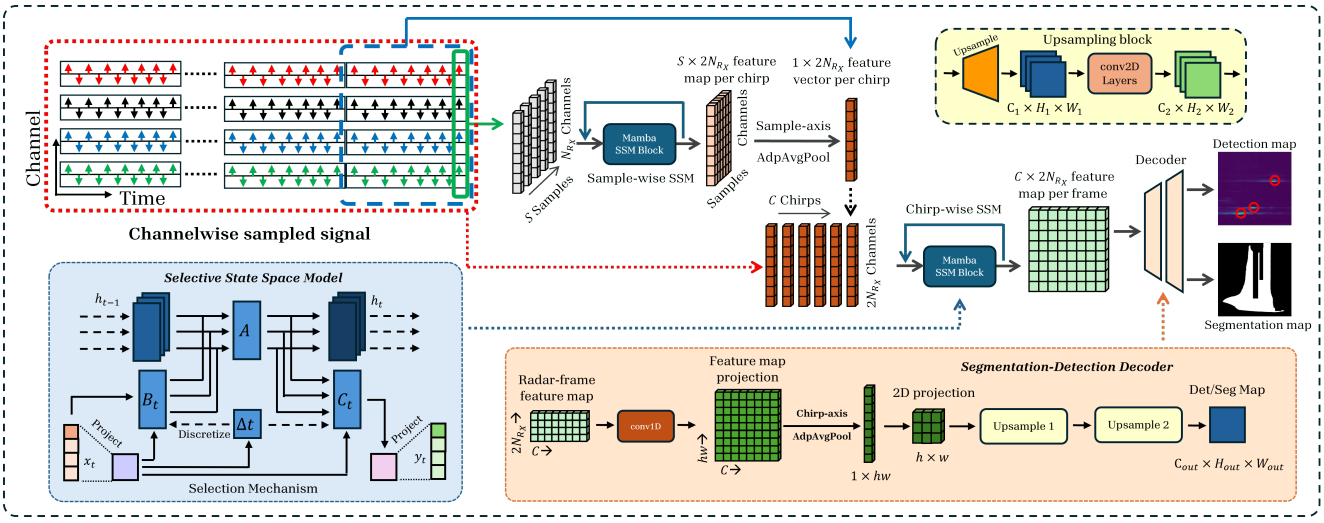

Figure 2. **SSMRadNet Architecture:** Raw complex ADC samples from N_{R_X}-channel feed into sample-SSM block. Sample states/chirp are aggregated into chirp feature maps and temporally pooled into token embeddings. Each chirp token then passes through chirp-SSM blocks to capture inter-chirp dynamics (motion, velocity). Their outputs are aggregated into radar-frame features and decoded—via a channel projection and parallel upsampling+convolutional layers—into detection and segmentation heads.

for segmentation and detection eliminating the memory and latency of buffering ADC cubes.

(3) Compute efficiency at competitive performance: Our model has significantly lower parameters, computation, and processing latency compared to prior works, but maintains comparable segmentation and detection performance.

2. Proposed Model Architecture

SSMRadNet is an end-to-end neural architecture that takes raw radar ADC data as input and produces a bird's-eye-view occupancy map as output. Figure 2 illustrates the overall design. The key idea is to model the radar data in sequential form, preserving the temporal and channel information. We decompose the problem into three stages: (1) Per-sample embedding, converting each fast-time ADC sample (across antennas) into a token representation; (2) SSM sequence modeling, which applies state-space model (Mamba) blocks first along the sample dimension of each chirp (intra-chirp modeling) and then along the chirp dimension (inter-chirp modeling) to capture range and azimuth structure; and (3) spatial decoding, which projects and transforms the sequence outputs into a 2D spatial feature map for the free driving space segmentation and vehicle detection task.

2.1. Radar Input and Tokenisation

Raw ADC samples arrive continuously at a fixed "fast-time" rate. At each tick

$$t = 1, 2, \ldots, C \times S,$$

where - C = number of chirps per frame, - S = number of fast-time samples (range bins) per chirp,

The ADC hardware simultaneously digitizes all N_{Rx} receiver channels, producing

$$\mathbf{X}(t) = \left[X_{c,1}(k),\, X_{c,2}(k),\, \ldots,\, X_{c,N_{\mathrm{Rx}}}(k)\right]^{\top} \in \mathbb{C}^{N_{\mathrm{Rx}}},$$

$c = \lceil t/S \rceil$ (current chirp index),
$s = t - (c-1)\,S$ (fast-time sample index within chirp i).

Immediately upon acquisition, we embed each vector:

$$\mathbf{z}(t) = \mathrm{MLP}\big([\Re(\mathbf{X}(t)); \Im(\mathbf{X}(t))]\big) \in \mathbb{R}^{N_{R_X}} \quad (1)$$

so that no buffering beyond the current tick is required. These embeddings are passed to the intra-chirp SSM block sequentially.

2.2. Sample-wise Mamba Block (Intra-Chirp SSM)

We leverage the temporal correlations between individual ADC samples to extract relevant features. To capture fine-grained range-domain structure within each chirp, we employ a detailed MAMBASSM block that augments the raw token sequence with causal convolution. These tokens $\mathbf{z}_{c,s}$ are split into N_{R_X} channels and fed through a grouped causal convolution of width d_{conv}:

$$x^{(g)}_{\mathrm{conv},s} = \sum_{k=0}^{d_{\mathrm{conv}}-1} w_g[k]\, \mathbf{z}^{(g)}_{c,s-k} + b_g, \quad s = 1, \ldots, S \quad (2)$$

with zero-padding for $s < d_{\mathrm{conv}}$, weights w_g, bias b_g. The grouped causal convolution maintains a small (first in first out) FIFO buffer of the previous $d_{\mathrm{conv}} - 1$ token embeddings, computing each $x_{\mathrm{conv},s}$ in realtime.

Next, we linearly project each convolved vector $x_{\text{conv},s} \in \mathbb{R}^{N_{R_X}}$ into three modulation streams:

$$\mathbf{p}_s = W_p\, x_{\text{conv},s} = \left[D_{\text{dt},s},\, B_{\text{mod},s},\, C_{\text{mod},s}\right] \quad (3)$$

$D_{\text{dt},s},\, B_{\text{mod},s},\, C_{\text{mod},s} \in \mathbb{R}^{d_{\text{state}}}$, where d_{state} determines how many hidden units the SSM uses to accumulate information over the fast-time samples.

A learnable log-decay parameter $A_{\log} \in \mathbb{R}^{d_{\text{state}}}$ yields the per-step decay ("$\odot$" is element-wise multiplication):

$$A = -\exp\big(A_{\log}\big), \quad \text{decay}_t = \exp\big(dt_t \odot A\big), \quad (4)$$

Here, $A \in \mathbb{R}^{d_{\text{state}}}$ is the per-channel decay coefficient that governs how much information of the previous hidden state carries over at each step.

The internal SSM state $\mathbf{h}_s \in \mathbb{R}^{d_{\text{state}}}$ updates recursively:

$$\mathbf{h}_s \;=\; \mathbf{h}_{s-1} \odot \text{decay}_s \;+\; \tilde{x}_s \odot \big(dt_s \odot B_{\text{mod},s}\big), \quad (5)$$

with $\tilde{x}_s$ the appropriately reshaped convolved output. Finally, the block output $y_s \in \mathbb{R}^{d_{\text{state}}}$ aggregates the state via:

$$y_s \;=\; \sum_{j=1}^{d_{\text{state}}} \mathbf{h}_s^{(j)}\, C_{\text{mod},s}^{(j)} \;+\; D\,\tilde{x}_s, \quad (6)$$

where $D \in \mathbb{R}^{d_{\text{state}} \times N_{R_x}}$ is a learned skip-connection matrix.

State summarisation: We apply average pooling over the hidden states $\mathbf{h}_{0-S}$, effectively compressing the temporal sequence $(S \times N_{R_X})$ into a single vector $y_c \in \mathbb{R}^{N_{R_X}}$. This compact representation carrying the distilled intra-chirp information is again passed through an MLP for channel expansion to $y_{ce} \in \mathbb{R}^{2N_{R_X}}$ and passed to the Chirp-SSM block.

2.3. Chirp-wise Mamba Block (Inter-Chirp SSM)

We apply a similar SSM block from Section 2.2 to the sequence of chirp tokens $\mathbf{y}_{ce} \in \mathbb{R}^{2N_{R_X}}$ along the chirp axis:

$$\mathbf{u}_c = \text{MambaSSM}(\mathbf{y}_{ce}), \quad c = 1, \ldots, C$$

We take the output u_c for each chirp as they arrive sequentially. These outputs are accumulated over the radar frame to create the slow-time vs channel feature map $U \in \mathbb{R}^{C \times 2N_{R_X}}$.

$$\mathbf{U} \;=\; [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_C] \;\in\; \mathbb{R}^{C \times 2N_{R_X}}$$

This accumulated temporal feature map $\mathbf{U}$ captures temporal correlations across the chirps.

2.4. BEV Projection and Decoder

We use similar structured decoders for both segmentation and detection tasks. The decoder takes the chirp-channel output feature map $U \in \mathbb{R}^{C \times 2N_{R_X}}$ from Chirp-SSM and passes it through a conv1D layer for dimension expansion. Then it is passed through a pooling layer across the chirp axis to accumulate the temporal information. Finally, the pooled features are reshaped to obtain the initial BEV projection $(h \times w)$.

$$F_1 = \text{Conv1D}(U) \;\in\; \mathbb{R}^{C \times (h\,w)}$$
$$\bar{F}_1 = \text{AdpAvgPool}_C(F_1) \in \mathbb{R}^{hw}$$
$$F = \text{Reshape}\big(F_1; h, w\big) \;\in\; \mathbb{R}^{h \times w}$$

The 2D map is then passed through upsampling layer and conv2D blocks with activation for refining spatial details for segmentation/detection.

$$Z_U^{(1)} = \mathcal{U}\big(F\big), \quad Z^{(1)} = \text{SiLU}\big(\text{Conv2D}(U^{(1)})\big)$$

$$Z_U^{(2)} = \mathcal{U}\big(Z^{(1)}\big), \quad Z^{(2)} = \text{SiLU}\big(\text{Conv2D}(U^{(2)})\big)$$

2.5. Why Multi-Scale Temporal Modeling?

For an FMCW radar, chirp samples contain information on the round-trip delay of a radio wave, containing information on the range of objects. Sample-SSM models the range across all the channels. This is important for finding the free space in the scene. However, for detection task, modeling the chirps across the radar frame is important for identifying background radio reflections and the moving objects. Chirp-SSM block helps identify these variations to identify the objects of interest in the scene. This multi-scale SSM approach enables effective separation of static background and dynamic targets in complex radar scenes.

3. Experimental Setup

This section gives an insight into the two datasets used for our experiments (RADIal[17] and RaDICaL[8]). The metrics used for evaluation, baseline architectures for radar-only detection and the training settings.

3.1. Datasets

3.1.1. RaDICaL (Radar, Depth, IMU, Camera Dataset).

RaDICaL [8] comprises synchronized 77 GHz FMCW radar (4 Rx × 3 Tx), stereo RGB-D, and IMU measurements. Frame annotations were generated from synchronized camera images. To improve robustness, full-sized images were divided into tiles [26] for RetinaNet [9] inference & detections on each tile were merged to form a binary mask, to produce ground-truth data (Figure 3(a)). Prior works have utilized image space to bird's-eye

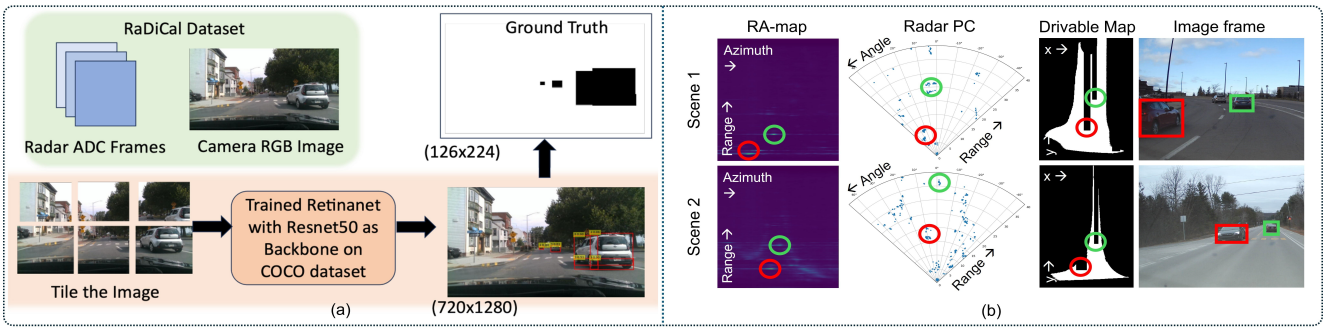

Figure 3. (a) **RaDICaL[8] Dataset:** Label generation from RGB frames (figure taken with permission from [24]). (b) **RADIal[17] Dataset:** FFT of raw ADC data produces range–Azimuth maps; CFAR yields radar point clouds; segmentation maps mark drivable (white) vs. non-drivable (black) areas; nearest and second-nearest vehicles are highlighted in red and green consecutively.

(BEV) view mapping to convert the detections into range-azimuth space. However, we allow model to learn these linear transformation during training. The primary advantages are : (a) elimination of manual calibration across radar configurations; and (b) seamless integration with image-space planning/control pipelines [23]; and (c) It eliminates the necessity to store the radar frame in the range-azimuth plane, thereby saving storage and computational resources.

3.1.2. RADIal (HD Radar Multi-Task Dataset).

RADIal[17] comprises roughly two hours of synchronized driving data—front-facing RGB video, a 16-beam LiDAR and a 77 GHz imaging radar (12 Tx × 16 Rx = 192 virtual channels)—collected across 91 sequences (city streets, highways and rural roads). Of its 25 000 radar frames, 8252 are semi-automatically annotated (Figure 3(b)) with per-vehicle centroids projected into the radar's polar (range R, azimuth A) and Cartesian (X,Y) coordinates, plus a drivable-area (freespace) mask obtained by projecting the LiDAR point cloud onto the ground. For our experiment, we use a 0.8-0.2 train-validation split for our experiments.

3.2. Implementation Details and Baselines

We summarize the implementation details on the two datasets in Table 2.

We used CNN-based ([15, 17, 19, 31, 33]), Attention-based ([5, 27]) & Chirp-based recurrent ([24]) state-of-the-art ML models in our experiments to establish a performance and runtime baseline for evaluation.

3.3. Evaluation Metrics

Following the **RADIal** benchmarking, for segmentation, we report mean intersection-over-union (**mIoU**) score between the predicted drivable-area mask and the annotated mask. In our ablation studies, we also report **dice coefficient**, mean average precision (**mAP**), mean average recall (**mAR**) and pixel-wise **accuracy** scores for testing model

Table 2. Training settings for the RADIal and RaDICaL datasets

Parameter	RADIal	RaDICaL
(C, S, N_{R_X})	(256, 512, 16)	(64, 192, 8)
Training epochs	200	300
Batch size	8	8
Optimizer	Adam	Adam
Learning rate	1×10^{-4}	1×10^{-4}
L2 regularization	5×10^{-6}	5×10^{-6}
Loss	Jaccard, Focal+Smooth L1	Pixel-wise BCE

Note: C- no. chirps/radar frame, S no. samples/chirp, N_{R_X} = no. receiver antennas. **BCE** - Binary cross entropy.

improvements. In the detection task, we report **mAP**, **mAR** and **F1** score.[17].

For the **RaDICaL** dataset, **Chamfer distance** was used to quantify the average bidirectional nearest-neighbor Euclidean distance between predicted and ground-truth BEV masks [3, 34], with lower values indicating tighter spatial alignment and thus better segmentation fidelity. Dice coefficient, was also reported to compare volumetric overlap between masks, with higher values indicating superior segmentation accuracy.

Runtime performance and efficiency is assessed by multiply–accumulate operations (MACs), parameter count (M) and latency (ms) for segmentation + detection in an NVIDIA RTX 4060 mobile GPU.

4. Results

4.1. Qualitative Results

We analyze SSMRadNet's features across stages to see how sequential radar data becomes spatial maps. Chirpwise features are pooled into vectors, generating map Figure 4 (a). After passing through the chirp-ssm, the feature-map (b) shows distinct feature peaks and troughs. After projecting this channel-chirp feature to spatial map

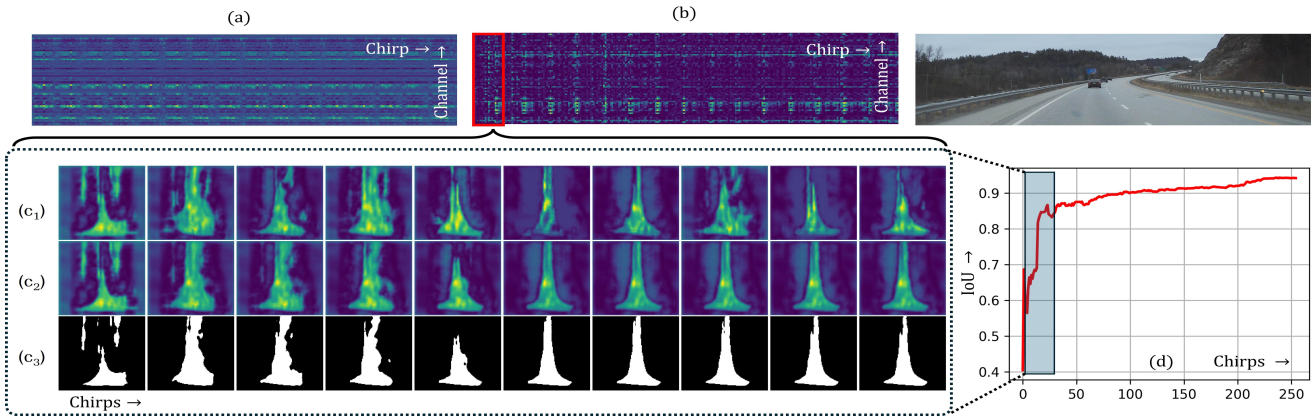

Figure 4. Feature evolution through model: (a) pooled sample-ssm output, (b) chirp-ssm output features. (c) chirp-wise feature evolution: (c_1) independent spatial information per chirp states, (c_2) accumulated over time, chirp state saturates, and (c_3) a projection is formed within first 25 chirps of random initialization. (d) IoU increases rapidly at first and reaches saturation.

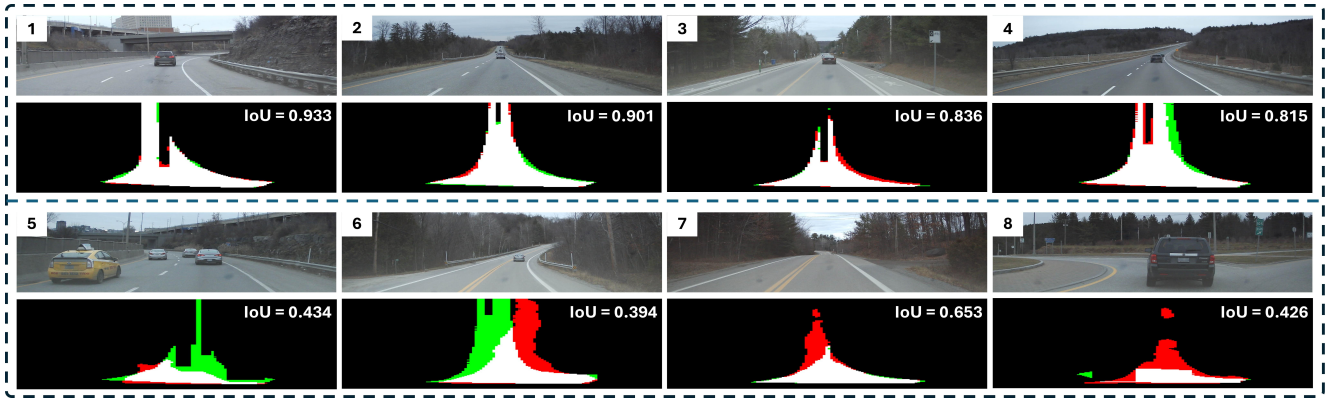

Figure 5. Ground truth vs. predicted drivable-space maps: white = true positives, green = false negatives (drivable, flagged as blocked), red = false positives (blocked, flagged as drivable). Row 1: high-IoU examples. Row 2: challenging cases (dense traffic, elevation changes).

followed by upsampling, Figure 4 (c) occupancy maps are formed. In this step, we compare the features extracted from each individual chirp to those obtained after aggregating chirp states across the entire frame. From (c_2), the spatial representation saturates early in the frame, and additional chirps contribute less to the final representation. This suggests that the radar frames can be downsized in along the chirp axis without significant loss in performance. We show some additional experiments on retaining internal state across frames in Section 4.2.

We compare ground truth and predicted freespace maps to reveal both successes and failure modes (Fig. 5). Samples 1–4 demonstrate accurate detection, with only slight drops in IoU when ground elevation fluctuates. The bottom row illustrates more challenging cases, where errors often stem from labeling inconsistencies; since drivable-space masks were auto-generated by image/LiDAR models and not calibrated to the radar sensor (refer to [17])—causing mismatches across elevations and obstacle distances. In sample 5, dense traffic causes the model to underestimate the free space (in green). In sample-6, 7, the ground truth and prediction maps show two different road curvatures, suggesting that the labeling might have been over constrained. Finally, in sample 8, the model fails to find the freespace when the view is obstructed by another vehicle at an exit, which can be attributed to edge-case samples in the dataset.

4.2. Ablation Studies

We begin our experiments with chirp-wise sequence modeling with SSM to reduce computation without sacrificing accuracy. In the ChirpNet[24] model family, flattened sample–channel linear projection scales poorly with large $S \times N_{R_x}$. So we introduced sample-wise SSM that processes each sample sequentially and uses its final state as the chirp token. This change alone boosts mIoU by 2.4% and dramatically cuts GMACs (row 2 of Table 3). Now the question comes, how to reduce the

Table 3. Ablation study on **RADIal** segmentation task.

Model	C-SSM	S-SSM	Proj	D-dim	Aggr	Expn	mIoU %	Dice %	mAP %	mAR %	Acc %	Comp. GMACs	Param (M)
SSM based chirp model	✓	–	✓	16	–	–	65.59	77.82	81.67	77.99	95.22	15.50	45.85
SSMRadNet-Proj	✓	✓	✓	16	Last	–	67.99	79.88	82.37	79.99	95.57	0.73	13.40
SSMRadNet-NoProj-16	✓	✓	–	16	Last	–	68.27	80.25	82.44	80.49	95.70	0.94	0.119
SSMRadNet-NoProj-32	✓	✓	–	32	Last	–	70.59	81.85	83.62	82.23	96.07	1.21	0.126
SSMRadNet-FeatConv	✓	✓	–	32	Conv1D	–	74.44	84.58	85.84	85.00	96.61	1.22	0.126
SSMRadNet-FeatPool	✓	✓	–	32	AvgPool	–	74.75	84.86	87.15	84.26	96.72	1.20	0.123
SSMRadNet (Ours)	✓	✓	–	32	AvgPool	✓	**78.60**	**86.77**	**88.02**	**86.81**	**97.10**	**1.67**	**0.314**

C-SSM - Chirpwise SSM, **S-SSM** - Samplewise SSM; **Proj** - Input projection; **D-dim** - Internal vector size of Mamba-SSM D-state; **Aggr** - Feature aggregation strategy of samplewise-SSM. ***Last*** - takes last state output, ***Conv1D*** - 1D conv over samplewise features, ***AvgPool*** - adaptive average pooling; ***Expn.*** - S-SSM output channels linearly expanded to $2N_{R_X}$ before C-SSM.

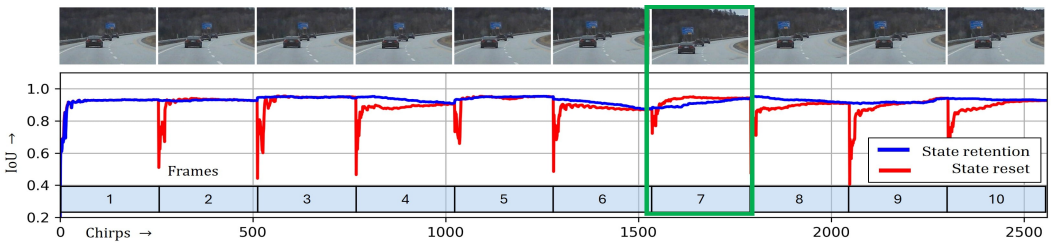

Figure 6. Decoding with state retention over frames (blue) vs state resetting per frame (red). State retention stabilizes segmentation over smooth sequences, while during physical perturbations, frame resetting performs better (green).

parameter count as well. We observed that projecting the entire time–channel feature map bloated parameters and destabilized training. For this, we applied conv1D based channel expansion from encoder output, followed by a temporal pooling to convert the chirp-channel features to a 2D featuremap. This yields a slight higher segmentation gain, far fewer parameters, and more stable training.

To test whether the SSM block D-state (output projection) has any effect in learning performance, we increased D-dim to 32 instead of 16. This gives us another 2.32% increase in segmentation performance, with slightly increased computation. Now, we ask whether choosing only the final state output from the Chirp-SSM was a good design choice, as it restricts the model to rely more on later samples. So we experimented with feature aggregation for generating chirp tokens. Performing a feature aggregation over the sample-dimension using Conv1D (row-5)or AvgPool (row-6) after the sample-ssm gives a huge performance increase, bringing SSMRadNet mIoU to 0.75. A last experiment on slow-time channel expansion (projecting N_{R_X} to $2N_{R_X}$) before the chirp-ssm layer offered another improvement, bringing our model mIoU to 0.79, achieving SoTA performance.

Figure 6 shows what happens if internal states are carried over to the next frame instead of resetting. In the particular scene of 10 frames, frame-wise state reset (red) yields mIoU=0.9253/frame, while state retention (blue) gives a higher score of mIoU=0.9258/frame and makes the state updates smoother. Observing each frame, we see that state retention generates early results in steady scenes, whereas a sudden change in scene (perturbations in frame 7) causes state retention to perform slightly worse than frame reset method. These observations suggest the possibility of inter-frame inferencing for speed and adaptive chirp reduction for additional computation savings.

4.3. Results on RaDICaL

SSMRadNet achieves a dice score of **0.996** on the RADICAL segmentation task (Table 5), matching the top-performing FFT-RadNet [17], U-Net[19] models, outperforming T-FFT-RadNet[5], while requiring **148**× less compute and **16**×–**30**× fewer parameters. Despite the drastic reduction in GMACs and model complexity, SSMRadNet maintains state-of-the-art segmentation accuracy, achieving an average Chamfer Distance of **0.086**, comparable with FFT-RadNet and U-Net, while surpassing transformer-based T-FFT-RadNet baselines and other ChirpNet variants.

4.4. Results on RADIal

SSMRadNet attains **0.79** mIoU on the segmentation task, showing competitive performance with state-of-the-art models like T-FFTRADNet[5] with 60× less compute (1.67 GMACs) and 33× fewer parameters (0.33 M)

Table 4. Overall segmentation and detection performance on **RADIal [17]**.

Class	Model	Segmentation	Detection					Computational Metrics*		
		mIoU	F1	mAP	mAR	RE (m)	AE (°)	GMACs	Params (M)	Latency (ms)
Convolution	Pixor (PC) [31]	—	—	0.96	0.32	0.17	0.25	—	—	—
	Pixor (RA) [31]	—	—	0.96	0.82	0.10	0.20	—	—	—
	PolarNet [15]	0.61	—	—	—	—	—	—	—	—
	Conv3D + FFT-RadNet [29]	0.75	0.47	0.58	0.39	0.19	0.33	—	—	—
	FFT-RadNet [17]	0.74	—	0.97	0.82	0.11	0.17	146.58	3.79	53.59
	FFT-RadUNet[a]	0.75	0.80	0.83	0.77	0.16	0.09	134.40	18.48	44.92
	ADCNet [33]	0.79	0.89	0.93	0.86	0.13	0.11	—	2.50	18.13**
	ADC UNet [33]	0.77	0.85	0.88	0.82	0.18	0.11	—	17.50	8.18**
	ADC UNet (NPT) [33]	0.73	0.80	0.83	0.77	0.19	0.10	—	—	—
	FourierNet-FFT-RadUNet[b]	0.78	0.86	0.84	0.87	0.16	0.11	134.41	19.13	48.73
	FourierNet-FFT-RadNet[c]	0.79	0.88	0.87	0.89	0.14	0.12	146.59	4.45	57.44
Attention	Self Attention-based chirp model[d]	0.65	—	—	—	—	—	33.00	50.95	20.37
	TFFT-RADNet [5]	0.79	0.87	0.88	0.87	0.16	0.13	99.38	10.29	52.90
Recurrent	ChirpNet (GRU) [24]	0.64	—	—	—	—	—	24.70	57.72	27.33
SSM	SSM-based chirp model[e]	0.66	—	—	—	—	—	15.50	45.85	8.32
	SSMRadNet (Ours)	**0.79**	0.77	0.83	0.71	0.14	0.15	**1.67**	**0.31**	14.20

RE = range error; AE = azimuth error; RA = range azimuth; PC = pointcloud.
[a, b, c, d, e] [a] FFT-RadNet[17] + UNet[19]; FourierNet[35] FFT fed to [b] FFT-RadUNet, [c] FFT-RadNet; ChirpNet[24] with [d] transformer [27], [e] SSM [6].
*Runtime characterization for **segmentation + detection** on an NVIDIA RTX 4060 mobile GPU. **Reported by [33] on an NVIDIA RTX 3090 GPU.

Table 5. Comparison with prior works on **RADICal[8]**.

Model	GMACs	Params (M)	Dice Coefficient (↑)	Chamfer (↓)
ChirpNet [24]	1.480	3.780	0.986	0.097
ChirpNetLite [24]	0.320	3.761	0.989	0.095
ChirpNet SSM[a]	0.340	3.761	0.990	0.088
ChirpNet-SelfAttn[b]	0.350	3.761	0.991	0.091
T-FFT-RadNet [5]	15.990	9.000	0.995	0.108
FFT-RadNet [17]	41.740	4.250	0.996	0.076
UNet [19]	15.140	17.270	0.996	0.078
SSMRadNet (Ours)	**0.108**	**0.566**	**0.996**	0.086

[a,b] [a] ChirpNet[24] with SSM; [b] with self-attention [27].
* Dice coefficient higher is better; Chamfer Distance lower is better.

than [5] (Table 4), while beating benchmark models like FFTRadNet[17] and ChirpNet[24]. Detailed results on the segemntation task is provided in Table 4, listing all the SoTA models grouped according to architecture type discussed in Table 1. In the detection task, SSMRadNet has yet to achieve SoTA performance as shown in Table 4, achieving F1 score of 0.77, mAP score of 0.83 and mAR score of 0.71. Simplified decoder structure used in our experiments is one of the key factors behind the performance loss in detection, and is a possible direction for future work.

5. Conclusion & Future Work

We presented SSMRadNet, a compact neural architecture that processes raw radar ADC signals via sample-wise and chirp-wise state-space models to produce BEV occupancy maps. Across two diverse datasets, SSMRadNet matches or exceeds prior state-of-the-art radar-only methods in segmentation benchmarks while using less than one million parameters and only a few GFLOPs per frame. Despite a minor decrease in detection performance, it remains competitive with other SOTA radar-only approaches. By processing sample-wise raw ADC sequences with hierarchical state space modeling architecture, SSMRadNet opens a new direction for efficient multi-task radar perception.

Future work could focus on training with adverse weather data to stress test model robustness. Moreover, SSMRadNet naturally lends itself to multi-modal fusion: additional camera or LiDAR embeddings could be incorporated into the SSMs or fused in the BEV decoder, combining radar's all-weather sensing with rich visual context. Finally, the linear-scaling compute of SSMs makes our approach suitable for next-generation radars with higher chirp count and larger antenna arrays. We hope this work inspires further exploration of lightweight, radar-specific deep learning architectures for autonomous systems.

6. Acknowledgement

This material is based upon work supported in part by SRC JUMP 2.0 (CogniSense, #2023-JU-3133) and in part by DARPA through the OPTIMA program. Any opinions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of SRC or DARPA.

References

[1] Shih-Yun Chu and Ming-Sui Lee. Mt-detr: Robust end-to-end multimodal detection with confidence fusion. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5241–5250, 2023. 1

[2] Yahia Dalbah, Jean Lahoud, and Hisham Cholakkal. Radarformer: Lightweight and nbsp;accurate real-time radar object detection model. In *Image Analysis: 22nd Scandinavian Conference, SCIA 2023, Sirkka, Finland, April 18–21, 2023, Proceedings, Part I.*, page 341–358, Berlin, Heidelberg, 2023. Springer-Verlag. 2

[3] Haoqiang Fan, Hao Su, and Leonidas Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image . In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, Los Alamitos, CA, USA, 2017. IEEE Computer Society. 5

[4] Yanwen Fang, Qinshuo Liu, Weiqin Zhao, Wei Huang, Lequan Yu, and Guodong Li. Selective state space models. *CoRR*, abs/2306.11817, 2023. 1

[5] James Giroux, Martin Bouchard, and Robert Laganière. T-fftradnet: Object detection with swin vision transformers from raw adc radar signals. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 4032–4041, 2023. 1, 2, 5, 7, 8

[6] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. 1, 8

[7] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. 1

[8] Teck Yian Lim, Spencer A. Markowitz, and Minh N. Do. Radical: A synchronized fmcw radar, depth, imu and rgb camera dataset with low-level fmcw radar signals. https://doi.org/10.13012/B2IDB-3289560_V1, 2021. 4, 5, 8

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 4

[10] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: visual state space model. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 1

[11] 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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. 1, 2

[12] Bence Major, Daniel Fontijne, Amin Ansari, Ravi Teja Sukhavasi, Radhika Gowaikar, Michael Hamilton, Sean Lee, Slawomir Grzechnik, and Sundar Subramanian. Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 924–932, 2019. 1

[13] Michael Meyer and Georg Kuschk. Deep learning based 3d object detection for automotive radar and camera. In *2019 16th European Radar Conference (EuRAD)*, pages 133–136, 2019. 1

[14] Eric Nguyen, Karan Goel, Albert Gu, Gordon W. Downs, Preey Shah, Tri Dao, Stephen A. Baccus, and Christopher Ré. S4nd: modeling images and videos as multidimensional signals using state spaces. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates Inc. 1

[15] Farzan Erlik Nowruzi, Dhanvin Kolhatkar, Prince Kapoor, Fahed Al Hassanat, Elnaz Jahani Heravi, Robert Laganiere, Julien Rebut, and Waqas Malik. Deep open space segmentation using automotive radar. In *2020 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4, 2020. 5, 8

[16] Sujeet Milind Patole, Murat Torlak, Dan Wang, and Murtaza Ali. Automotive radars: A review of signal processing techniques. *IEEE Signal Processing Magazine*, 34(2):22–35, 2017. 1

[17] Julien Rebut, Arthur Ouaknine, Waqas Malik, and Patrick Pérez. Raw high-definition radar for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17021–17030, 2022. 1, 2, 4, 5, 6, 7, 8

[18] Mark A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill, New York, 2nd edition, 2014. 1

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 5, 7, 8

[20] R. Roy and T. Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):984–995, 1989. 1

[21] Nicolas Scheiner, Florian Kraus, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. Object detection for automotive radar point clouds—a comparison. *AI Perspectives*, 3(6), 2021.

[22] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, 1986. 1

[23] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187–210, 2018. 5

[24] Sudarshan Sharma, Hemant Kumawat, and Saibal Mukhopadhyay. Chirpnet: Noise-resilient sequential chirp based radar processing for object detection. In *2024 IEEE/MTT-S International Microwave Symposium - IMS 2024*, pages 102–105, 2024. 1, 2, 5, 6, 8

[25] Merrill I. Skolnik. *Radar Handbook*. McGraw-Hill, New York, 3 edition, 2008. 1

[26] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9626–9635, 2019. 4

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 5, 8

[28] 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. 1

[29] Jialong Wu, Mirko Meuter, Markus Schoeler, and Matthias Rottmann. Sparseradnet: Sparse perception neural network on subsampled radar data. In *Computer Vision – ECCV 2024*, pages 52–69, Cham, 2025. Springer Nature Switzerland. 8

[30] Zizhang Wu, Yunzhe Wu, Xiaoquan Wang, Yuanzhu Gan, and Jian Pu. A robust diffusion modeling framework for radar camera 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3282–3292, 2024. 1

[31] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 5, 8

[32] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. Radarnet: Exploiting radar for robust perception of dynamic objects. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII*, page 496–512, Berlin, Heidelberg, 2020. Springer-Verlag. 1

[33] Bo Yang, Ishan Khatri, Michael Happold, and Chulong Chen. Adcnet: Learning from raw radar data via distillation, 2023. 1, 2, 5, 8

[34] 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. 1, 5

[35] Peijun Zhao, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, and Andrew Markham. Cubelearn: End-to-end learning for human motion recognition from raw mmwave radar signals. *IEEE Internet of Things Journal*, 10(12):10236–10249, 2023. 1, 8