RAVEN: Radar Adaptive Vision Encoders for Efficient Chirp-wise Object Detection and Segmentation

Supplementary Material

1. Experimental Details

1.1. Datasets

1.1.1. RaDICaL dataset and annotation

We use the RaDICaL dataset [2], which provides synchronized measurements from a 4-Rx, 3-Tx 77 GHz FMCW radar, an RGB camera, a depth camera, and an inertial measurement unit (IMU). The depth camera produces reliable depth estimates only up to approximately 10 m, making it less effective for distant objects, whereas the radar remains sensitive to far-range targets. Scenes are recorded from a vehicle-mounted sensor rig across urban streets, country roads, and highways.

Unlike many prior radar datasets [8], RaDICaL releases raw ADC samples in addition to preprocessed range–Doppler or range–angle maps. This preserves the full semantic content of the radar data and enables efficient raw chirp-wise processing. While the radar hardware supports a 3-Tx×4-Rx MIMO configuration, the dataset was collected using a 2-Tx×4-Rx TDM MIMO setup, yielding 8 virtual channels per chirp. Our pipeline uses all available virtual channels.

RaDICaL Annotation pipeline. Supervised radar learning is limited by the difficulty of generating high-quality labels directly in the radar domain (e.g., range–azimuth–Doppler tensors or sparse point clouds). Instead of annotating radar data manually or relying on CFAR-based heuristics, we derive supervision from synchronized RGB images. We use a RetinaNet [3] detector with a ResNet-50 backbone pre-trained on COCO on the camera images. To improve detection of small and distant objects, we adopt a tiling strategy [1]: each image is split into overlapping tiles, inference is run independently on each tile, and detections are stitched back into the original resolution. This improves recall for far objects compared to a single-pass detector. We restrict COCO classes to *person*, *bicycle*, *car*, *motorcycle*, *bus*, and *truck*.

From the stitched detections, we generate a binary mask in image space. This mask serves as the ground-truth signal during training. Importantly, we do *not* use radar-to-camera calibration matrices to project annotations across modalities; instead, the model learns cross-modal alignment implicitly through its architecture. This avoids dependence on calibration, eliminates the need to store radar-domain labels, and minimizes the alignment noise and sparsity issues seen in RODNet-style labels [7]. The tiling strategy can produce duplicate detections when a single object spans multiple tiles; we mitigate this with non-maximum suppression (NMS) after stitching. An overview of the RaDICaL annotation pipeline is shown in Fig. 1(a).

1.1.2. RADIal dataset overview

For comparison and context, we briefly summarize the RADIal dataset [4]. RADIal uses a high-definition imaging radar with $N_{\text{Rx}} = 16$ receiver antennas and $N_{\text{Tx}} = 12$ transmitter antennas, giving $N_{\text{Rx}}N_{\text{Tx}} = 192$ virtual channels. This dense virtual array provides fine azimuth resolution and supports elevation estimation.

The radar is accompanied by a 16-layer automotive-grade LiDAR, a 5 Mpix RGB camera mounted behind the windshield, and synchronized GPS and CAN traces for vehicle pose and kinematics. The three sensors have parallel horizontal lines of sight in the driving direction, and their extrinsic calibration is provided. RADIal contains 91 sequences of 1–4 minutes each (city, highway, and countryside driving), for a total of roughly 25k synchronized frames, of which 8,252 frames are labeled with about 9,550 vehicles. The RADIal signal-processing and labeling pipeline is summarized in Fig. 1(b).

Nevertheless, RADIal is the only large-scale dataset that provides raw analog-to-digital converter (ADC) radar signals rather than only preprocessed FFT cubes. This makes it possible to train foundation models directly on raw radar data streams, yielding competitive performance and architectures like RAVEN that can exploit raw signals efficiently for on-edge deployment.

2. RAVEN Block-Wise Analysis

RAVEN's encoder–decoder pipeline consists of four logical components: (i) per-RX channel SSMs that operate along fast time, (ii) an antenna attention mixer that reconstructs virtual-MIMO features, (iii) a chirp-wise SSM backbone along slow time, and (iv) lightweight decoders for detection and segmentation. We profile them individually. Figure 2 summarizes the per-block parameter count, GMACs, and latency contributions, normalized to the full model. These plots show a consistent picture: most parameters reside in the 2D decoders, most MACs in the combination of chirp-wise SSM and decoders, and most latency in the channel SSM. The antenna mixer, despite encoding detailed virtual-MIMO structure, contributes only a small fraction

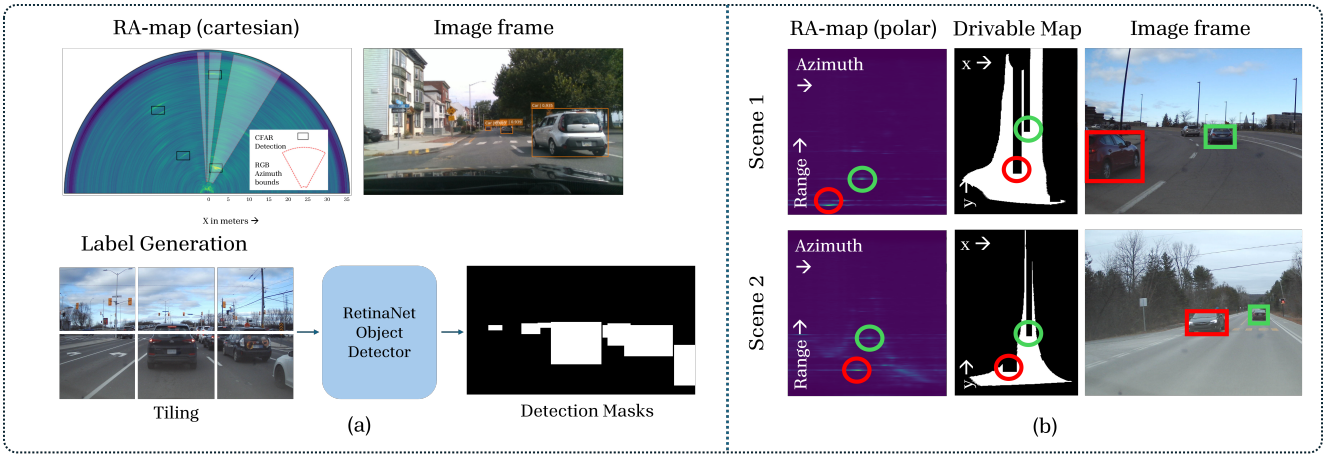


Figure 1. (a) **RaDICaL [2]**: label generation from RGB frames using a tiled RetinaNet detector (adapted from [6]). (b) **RADIal [4]**: FFT of raw ADC data produces range–azimuth maps; CFAR yields radar point clouds; segmentation maps mark drivable (white) vs. non-drivable (black) areas; nearest and second-nearest vehicles are highlighted in red and green, respectively.

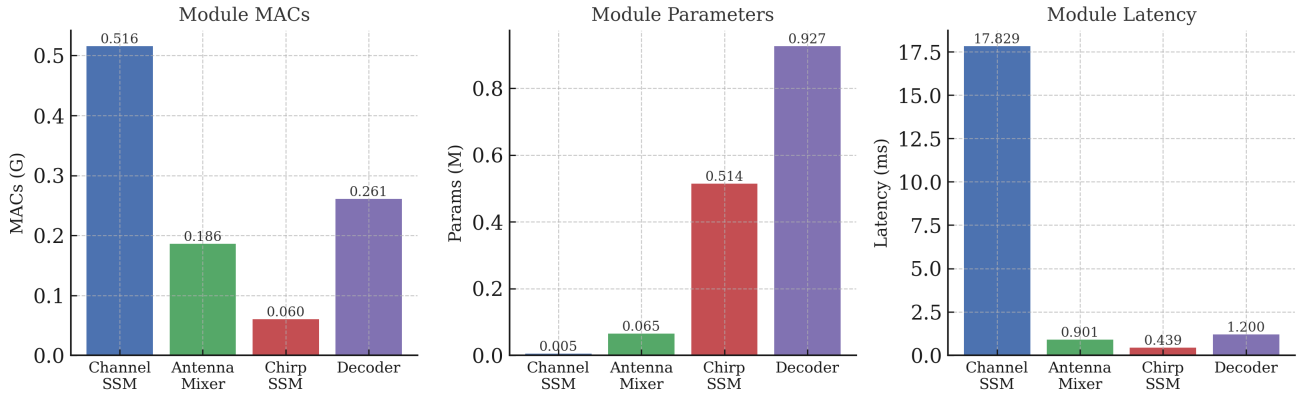


Figure 2. Per-block latency (ms) on a single GPU. The channel SSM is the main sequential bottleneck because it processes long fast-time sequences; the mixer and decoders are highly parallelizable.

of total compute and parameter count. This means we can afford spatial reasoning without compromising efficiency, provided that the fast-time block remains narrow and the slow-time backbone operates on sufficiently compressed tokens.

3. Physics-guided Encoder Design

The design of RAVEN's encoder is guided directly by the signal and array physics of FMCW MIMO radar. In this section, we move from the basic chirp model to the virtual-array view and then to architectural choices: (i) how fast-time structure suggests 1D state space models, (ii) how MIMO geometry encodes angle, (iii) why naive channel mixing destroys that information, and (iv) how our channel SSMs and antenna mixer modules implement a physics-aligned, end-to-end encoder for object detection. Section 5 then validates these choices empirically.

3.1. FMCW chirp and beat signal

A single FMCW chirp of duration T_c and bandwidth B has instantaneous transmit frequency

$$f_{\text{tx}}(t) = f_0 + St, \qquad S = \frac{B}{T_c}, \quad 0 \le t \le T_c, \tag{1}$$

and complex baseband signal

$$s_{\text{tx}}(t) = \exp\Big(j2\pi\Big(f_0 t + \tfrac{1}{2}St^2\Big)\Big). \tag{2}$$

A target at range R and radial velocity v yields an echo delayed by $\tau = \frac{2R}{c}$ and Doppler-shifted by $f_D = \frac{2v}{\lambda}$, where c is the speed of light and λ is the carrier wavelength. After mixing with the transmit signal and low-pass filtering, the resulting beat signal can be approximated as

$$s_b(t) \approx \exp\big(j2\pi\big(f_b t + \text{const}\big)\big), \qquad f_b \approx f_r + f_D, \tag{3}$$

with range-dependent frequency $f_r = \frac{2SR}{c}$ and Doppler frequency $f_D = \frac{2v}{\lambda}$. Thus, *range* manifests as a linear frequency along fast time, while *velocity* appears as phase evolution across chirps.

Let T_s denote the ADC sampling period and $n \in \{0, \ldots, N_s-1\}$ the fast-time index. For chirp index k with repetition interval T_R, a single-target beat sample at one receiver is approximately

$$x_k[n] \propto \exp\big(j2\pi(f_r n T_s + f_D k T_R)\big), \tag{4}$$

which is the starting point for our fast-time state space encoders: the fast-time dimension is a 1D sequence whose frequency encodes range, motivating SSMs along fast time (ADC samples).

3.2. MIMO virtual array and angle encoding

For an N_{Rx}-element receive array with inter-element spacing d, a plane wave from azimuth θ induces a spatial steering vector

$$\mathbf{a}(\theta) = \big[1, e^{j\phi}, \ldots, e^{j(N_{\text{Rx}}-1)\phi}\big]^\top, \quad \phi = 2\pi\frac{d}{\lambda}\sin\theta. \tag{5}$$

Stacking the beat samples across antennas for chirp k and fast-time index n yields

$$\mathbf{x}_k[n] = \sum_\ell A_\ell e^{j2\pi(f_{r,\ell} n T_s + f_{D,\ell} k T_R)}\mathbf{a}(\theta_\ell) + \mathbf{w}_k[n], \tag{6}$$

where A_ℓ and θ_ℓ denote the complex amplitude and angle of the ℓ-th target and $\mathbf{w}_k[n]$ represents noise and clutter.

In TDM/DDM MIMO, each receiver additionally sees echoes from multiple transmitters, so the virtual array combines TX and RX patterns. The virtual steering vector becomes a Kronecker product of TX and RX steering vectors, and different transmitters are separated in either time (TDM) or Doppler (DDM). Crucially, *angle information is encoded in relative phase differences across antennas and transmitters*; any operation that averages these channels too early risks collapsing the array response to a single beam. Architecturally this means we should preserve per-antenna channels until we have a mechanism that can explicitly reason over them.

3.3. Why naive channel mixing loses angle

To see how early mixing harms angular resolution, consider a simplistic encoder that first maps each receiver's fast-time sequence into a scalar summary and then averages across receivers. Let $x_{r,k}[\cdot]$ denote the fast-time samples for receiver r and chirp k, and let $g(\cdot)$ be a (near-linear) temporal encoder. We define

$$u_{r,k} = g\big(x_{r,k}[\cdot]\big), \qquad \mathbf{u}_k = \big[u_{1,k}, \ldots, u_{N_{\text{Rx}},k}\big]^\top, \tag{7}$$

and obtain a per-chirp token via uniform averaging

$$z_k = \tfrac{1}{N_{\text{Rx}}}\sum_{r=1}^{N_{\text{Rx}}} u_{r,k} = \mathbf{w}^{\text{H}}\mathbf{u}_k, \qquad \mathbf{w} = \tfrac{1}{N_{\text{Rx}}}\mathbf{1}. \tag{8}$$

If the scene is dominated by a single far-field target, then $\mathbf{u}_k$ is approximately proportional to the steering vector $\mathbf{a}(\theta)$, so the token becomes

$$z_k \propto \mathbf{w}^{\text{H}}\mathbf{a}(\theta) = \tfrac{1}{N_{\text{Rx}}}\mathbf{1}^{\text{H}}\mathbf{a}(\theta). \tag{9}$$

This is precisely the output of a fixed beamformer with weights $\mathbf{w}$: all spatial information is compressed into one scalar, and only that one beam pattern is available to the downstream network. Relative phase shifts $e^{jr\phi}$ across antennas, which distinguish different angles θ, no longer appear explicitly in the representation.

In DDM/TDM MIMO, where TX waveforms are interleaved in Doppler or time, this problem becomes more severe: the virtual array structure is already entangled across chirps and frequencies, and early channel mixing further entangles it, making it difficult for later layers to recover angle-of-arrival (AoA) cues without reconstructing RAD tensors. This motivates an encoder that *first* models each channel's fast-time dynamics and *then* performs explicit, learned spatial mixing across antennas.

3.4. Per-RX Channel Fast Time SSMs and Antenna mixer as a radar physics-friendly alternative

RAVEN avoids this pitfall by inserting two carefully structured stages before the slow-time backbone.

Per-RX channel fast time SSMs: Instead of aggregating channels immediately, we maintain a separate fast-time encoder for each receiver. For receiver r and chirp k, we collect the I/Q sequence

$$\mathbf{x}_{r,k} \in \mathbb{R}^{N_s \times 2}, \tag{10}$$

and feed it to a Mamba-style state space model SSM_r:

$$\tilde{\mathbf{z}}_{r,k} = \text{SSM}_r(\mathbf{x}_{r,k}) \in \mathbb{R}^{N_s \times 2}, \tag{11}$$

$$\mathbf{f}_{r,k} = \text{Pool}_1\big(\tilde{\mathbf{z}}_{r,k}^\top\big) \in \mathbb{R}^2, \tag{12}$$

where Pool_1 adaptively averages the fast-time dimension to length 1. Stacking across receivers yields

$$\mathbf{F}_k = [\mathbf{f}_{1,k}, \ldots, \mathbf{f}_{N_{\text{Rx}},k}] \in \mathbb{R}^{N_{\text{Rx}} \times 2}, \tag{13}$$

so each antenna contributes a compact per-chirp descriptor that retains its relative phase and amplitude structure. This implements the "first compress fast time per channel" step suggested by the physics above.

Model Variant	Channel SSM	Antenna Mixer	mIoU	F1	mAP	GMACs
(A) Shared Fast-time SSM	✗	✗	0.79	0.77	0.81	1.67
(B) Cross-Antenna Attention + Shared Fast-time SSM	✗	✗	0.80	0.79	0.83	38.89
(C) Shared Fast-time SSM + Cross-Antenna Attention	✗	✓	0.79	0.80	0.83	1.62
(D) Cross-Antenna Attention + Channel SSM	✓	✓	0.84	0.88	0.88	34.56
(E) **Channel SSM + Cross-Antenna Attention (RAVEN, full-frame)**	✓	✓	**0.90**	**0.93**	**0.95**	**1.02**
(F) **Channel SSM + Cross-Antenna Attention (RAVEN, sub-frame)**	✓	✓	**0.85**	**0.89**	**0.88**	**0.27**

Table 1. **Ablation of channel SSM and antenna mixer on RADIal.** All variants share the same chirp-wise SSM backbone and decoders. Our physics-guided RAVEN encoders (blue rows), which apply per-RX channel SSMs before the antenna mixer, achieve the best trade-off between accuracy and compute; the sub-frame variant further improves efficiency for early-exit decisions.

Attention-based antenna mixer: The antenna mixer then interprets $\mathbf{F}_k$ as a set of tokens and learns how to combine them in a way analogous to a set of learnable beams. After projecting from $\mathbb{R}^2$ to $\mathbb{R}^d$ and adding RX embeddings, we obtain

$$\mathbf{H}_k^{\text{rx}} = W_{\text{in}}\mathbf{F}_k + \mathbf{E}^{\text{rx}} \in \mathbb{R}^{N_{\text{Rx}}\times d}, \tag{14}$$

and introduce N_{Tx} TX queries $\mathbf{Q} \in \mathbb{R}^{N_{\text{Tx}}\times d}$. Multi-head attention produces a set of TX-aligned features

$$\mathbf{T}_k = \text{Attn}(\mathbf{Q}, \mathbf{H}_k^{\text{rx}}, \mathbf{H}_k^{\text{rx}}) \in \mathbb{R}^{N_{\text{Tx}}\times d}, \tag{15}$$

which can be interpreted as learnable steering patterns over the RX tokens.

To expose joint TX–RX information to the downstream SSM, we form small pairwise features for every (r, t) combination (e.g., by concatenation and a linear layer) and compress each pair to a two-dimensional vector:

$$\mathbf{p}_{r,t,k} = W_{\text{pair}}\left[\mathbf{h}_{r,k}^{\text{rx}}; \mathbf{t}_{t,k}\right] \in \mathbb{R}^2, \tag{16}$$

$$\mathbf{y}_k = \text{LN}\big(\text{vec}\big(\{\mathbf{p}_{r,t,k}\}_{r,t}\big)\big) \in \mathbb{R}^{2N_{\text{Rx}}N_{\text{Tx}}}. \tag{17}$$

Thus, each chirp is represented by a $2N_{\text{Rx}}N_{\text{Tx}}$-dimensional *learned virtual-antenna feature vector*. Crucially, this representation is obtained directly from time-domain ADC signals through learned projections and attention; we never perform explicit 2D/3D FFTs across range or angle, and we never construct dense range–azimuth–Doppler (RAD) tensors.

Stacking over chirps gives

$$\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{N_c}] \in \mathbb{R}^{N_c\times(2N_{\text{Rx}}N_{\text{Tx}})}, \tag{18}$$

which preserves the structure of the MIMO array in a compact latent space and feeds directly into the chirp-wise SSM. This gives a physics-inspired end-to-end encoder: fast-time SSMs for range, attention for angle, and chirp-wise SSMs for temporal evolution.

4. Ablation: Role and Ordering of Per RX Channel Fast Time SSM and Antenna Mixer

The radar physics discussion suggests that both the per-RX channel SSMs and the cross-antenna attention mixer are important, and that their ordering should follow the natural flow of information. Our hypothesis is to first compress ADC samples across each receiver channel along fast time, then isolate angle information from the channels. To validate this, we compare the model variants in Table 1, which all share the same chirp-wise SSM and decoders but differ in how they model fast time and cross-antenna interactions.

Model Variants: We briefly restate what each row does and why some variants are much heavier:

- **(A) Shared fast-time SSM.** A single fast-time SSM operates on all $2N_{\text{Rx}}$ input channels jointly. There is no per-RX channel SSM and no dedicated antenna mixer; the model treats the ADC samples as a generic multichannel sequence. This gives a reasonable baseline in both accuracy and compute (1.67 GMACs).
- **(B) Cross-antenna attention + shared fast-time SSM.** Here we augment the shared fast-time SSM with global cross-antenna interactions inside the same block, but still without a separate channel SSM module or a structured antenna mixer head. In our implementation, this attention is applied at *full fast-time resolution*: it sees roughly $512 \times N_{\text{Rx}}$ tokens per chirp instead of a compressed set of per-antenna summaries. Because attention scales at least quadratically with the sequence length, this makes the block extremely expensive (38.89 GMACs) even though the accuracy gain over (A) is small.
- **(C) Fast-time SSM + cross-antenna attention.** A shared fast-time SSM is followed by a dedicated cross-antenna attention mixer. This introduces an explicit mixer module, but because the fast-time SSM is still shared across all channels, it does not produce clean per-antenna summaries. The mixer therefore operates on features that

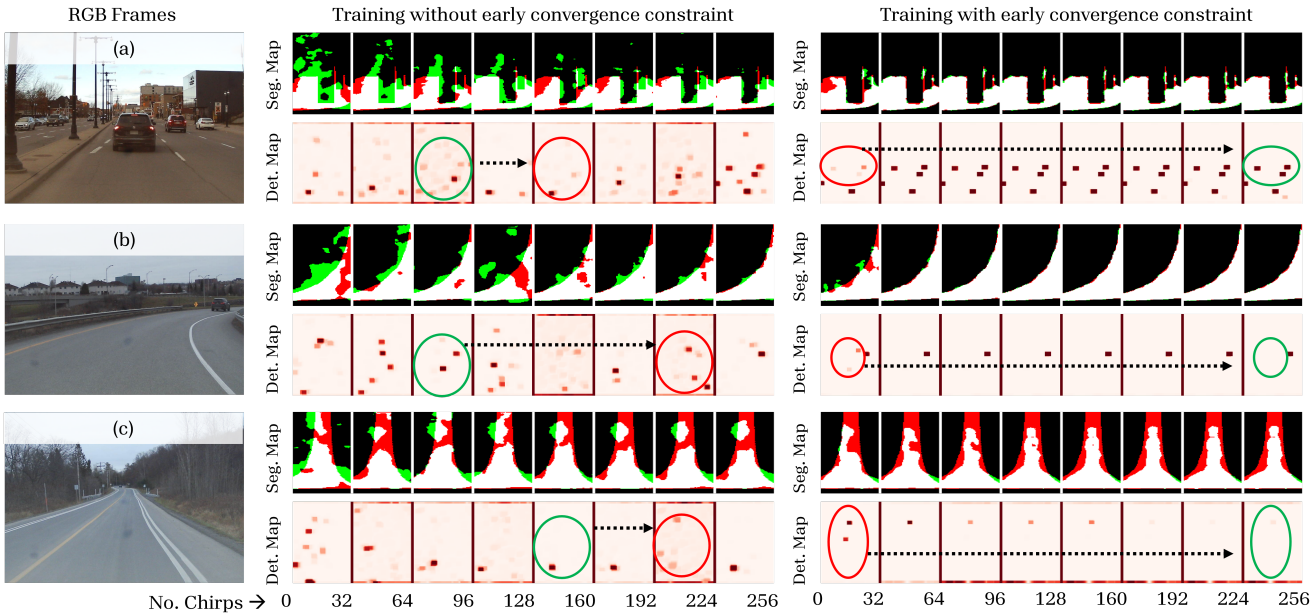


Figure 3. Segmentation and detection maps across driving scenes with and without multi-chirp supervision. Without supervision across chirp levels, segmentation gradually approaches the ground truth, but detection remains unstable throughout the sequence, consistent with Figure 4. In (a), the model forgets real objects mid-frame that only reappear at the end. In (b), it initially identifies one set of objects but later predicts an entirely different set. In (c), it begins to hallucinate obstacles near the final chirps. With multi-chirp supervision, these issues disappear: detection becomes consistent across the sequence, and both segmentation and detection remain accurate through the final frame.

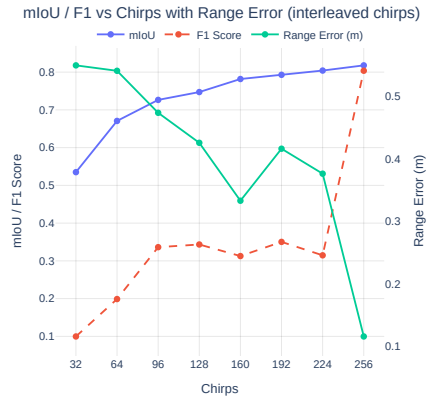
partially blur channel structure, and accuracy improves only marginally over (A)/(B), while compute (1.62 GMACs) remains comparable to (A).

- **(D) Cross-antenna attention + channel SSM.** Both channel SSMs and the mixer are present, but in the *reverse* order: cross-antenna attention is applied first on lightly projected I/Q samples, and the resulting mixed features are then processed by per-RX SSMs. As in (B), the attention still runs on long fast-time sequences (≈ 512 samples per antenna), so it sees a large number of tokens and dominates the compute. This explains why (D) achieves good accuracy (mIoU 0.84, F1 0.88) but remains very heavy at 34.56 GMACs.
- **(E) Channel SSM + cross-antenna attention (RAVEN, full-frame).** Our proposed hybrid encoder: per-RX channel SSMs first compress each fast-time sequence into a low-dimensional token, so each chirp is represented by only N_{Rx} channel tokens instead of $512 \times N_{\text{Rx}}$ time samples. The cross-antenna attention mixer then operates on this compressed set of tokens, reconstructing virtual MIMO structure at a much smaller sequence length. This ordering is motivated by the physics analysis in Section 3.
- **(F) Channel SSM + cross-antenna attention (RAVEN, sub-frame).** This variant uses the hybrid encoder as (E) but trains with our early-chirp criterion and decodes from a sub-frame subset of chirps. It maintains strong accuracy while reducing compute to 0.27 GMACs, providing an efficient early-exit option for on-edge deployment.

Optimal placement of the hybrid design for efficiency.

The trends in Table 1 support the physics-guided design. Variants (B) and (D) show that simply adding cross-antenna attention on top of raw fast-time sequences is not a good trade-off: attention over $\sim 512 \times N_{\text{Rx}}$ tokens per chirp is powerful but incurs tens of GMACs. Variants (A) and (C), which avoid that extreme cost, either lack structured cross-antenna reasoning or do not preserve clean per-channel summaries and therefore underperform in accuracy. Our RAVEN encoders in (E) and (F) implement a hybrid placement: channel SSMs first compress each fast-time stream into a single token per antenna, and the cross-antenna attention mixer then operates on this compact set of tokens. This reduces the attention sequence length by roughly a factor of 512 while preserving the virtual-array structure, and is exactly why (E) and (F) achieve the best balance between accuracy and compute, turning the SSM-first mixer placement into a core physics-inspired architectural contribution rather than just another variant.

5. Early Chirp State Saturation Experiment



(a)

(b)

Figure 4. Design motivation for adaptive chirp selection. **(a)** Validation curves illustrate how detection and segmentation performance evolve with increasing chirp count. Without explicit early-convergence constraints, detection remains suboptimal until the full chirp frame is processed. **(b)** Introducing multi-chirp supervision during training encourages the model to saturate earlier and learn temporal continuity across chirps, yielding smoother convergence and higher overall detection–segmentation performance.

We evaluate the impact of enforcing early state convergence by decoding from partial chirp sets, compared to training without this constraint. Consistent with the observations in [5], we find that mIoU improves rapidly in the early chirp regime. However, this behavior does not naturally extend to detection, where performance depends on information accumulated across the full chirp sequence (Figure 4a). To encourage the latent states to saturate earlier—enabling reliable early-exit decisions—we decode intermediate outputs from multiple chirp subsets and supervise each with detection targets (Section 3.3):

$$\mathcal{L}_{\text{task}} = \sum_{L \in \mathcal{L}} \Big[\ell_{\text{det}}\big(\widehat{\text{Det}}^{(L)}, \text{Det}^{\star}\big) + \ell_{\text{seg}}\big(\widehat{\text{Seg}}^{(L)}, \text{Seg}^{\star}\big) \Big]$$

Supervising detection at multiple chirp depths forces the model to extract complete spatial cues from early temporal observations. Learning this temporal–spatial continuity within radar frames improves overall performance, as shown in Figure 4b.

We further visualize the effect of this training strategy on sub-frame decisions (Figure 3). Without early-decision supervision, the temporal progression of the radar frame is poorly preserved: detection heatmaps fluctuate significantly across chirps, with the model sometimes forgetting strong reflectors or hallucinating obstacles mid-frame. With the early-chirp constraint, these inconsistencies largely disappear. Both detection and segmentation exhibit smoother evolution across chirps, and the latent states stabilize much earlier. This leads not only to improved detection performance but also to earlier state saturation, which directly contributes to computational savings under our early-exit framework.

6. Additional Results

6.1. Architecture Hyperparameters

Table 2 lists the key architectural hyperparameters of RAVEN. The antenna mixer is deliberately narrow (64 dims, 8 heads) so that it adds negligible GMACs on top of the channel SSMs; the Mamba state dimension of 16 keeps per-RX encoders lightweight; and the 1×1 Conv1D projection maps chirp features to a 32×56 BEV grid before the detection and segmentation decoders.

Component	Configuration
Antenna Mixer	Dim 64, 8 heads, expansion $4\times$, init $\sim\mathcal{N}(0,1)$
SSM (Mamba)	State dim 16, conv kernel 4, expansion 2
Spatial Proj.	1×1 Conv1D $\rightarrow$ 1792 ch. (grid 32×56)

Table 2. **RAVEN architectural hyperparameters.**

To quantify the impact of compressing per-RX ADC samples to a single token, we test RAVEN variants where the fast-time SSM condenses each RX channel into $K \in \{1, 4, 8, 16\}$ tokens before the cross-antenna mixer. Table 3 shows that the marginal gain from $K{=}1$ to $K{=}16$ is only 0.3% F1, confirming that fast-time samples are highly compressible and that a single token captures the essential range/phase information needed downstream, validating our design choice.

Tokens per RX (K)	1 (RAVEN)	4	8	16
F1 (Detection)	0.934	0.936	0.936	0.937

Table 3. **Fast-time token compression ablation on RADIal.** Marginal gains from $K{=}1$ to $K{=}16$ confirm that a single token per RX channel sufficiently captures range and phase information.

6.2. Early-Exit Decision Rule: Cosine Similarity vs. Entropy

We compare two chirp-stopping criteria: (i) minimum cosine similarity between the new chirp latent state and all prior states (our default), and (ii) entropy of the chirp-state distribution. Although entropy produces a smoother signal, cosine similarity yields better validation performance (+0.85% mAP, +0.67% mIoU) at similar compute, as shown in Table 4 and Figure 5. The cosine rule directly measures the novelty of each new chirp in the latent space, providing a more reliable and interpretable stopping condition.

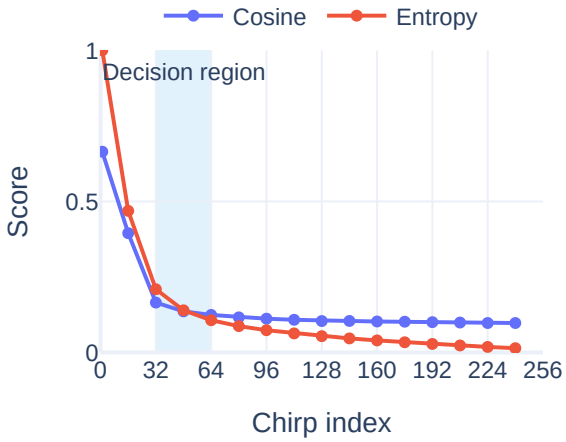


Figure 5. **Cosine distance vs. entropy as chirp-stopping signals.** Cosine similarity (blue) produces a cleaner knee-point, enabling more consistent early-exit decisions than entropy (orange).

Stopping Rule	mAP	mAR	F1	mIoU
Cosine (Ours)	**94.5**	**95.1**	**94.8**	**89.5**
Entropy	93.6	94.0	93.8	88.8

Table 4. **Early-exit decision rule comparison on RADIal** (all metrics in %). Cosine similarity outperforms entropy on every metric.

6.3. Adaptive Chirp Selection vs. Scene Velocity

Although static scenes nominally require less Doppler resolution, multiple chirps are still needed to form the virtual MIMO aperture for angular cues; fewer chirps shrink the virtual array and degrade spatial localization. Figure 6 shows no correlation between selected chirp count and object velocity, confirming that our adaptive stopping rule is driven by *prediction stability* in the latent space rather than scene motion.

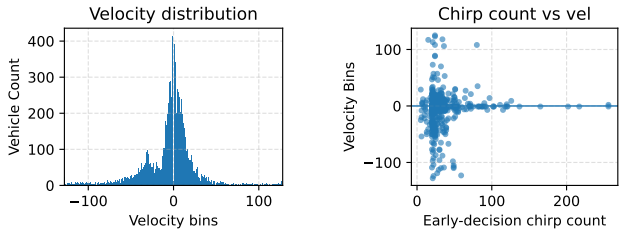


Figure 6. **Velocity distribution and adaptive chirp count.** (Left) Velocity histogram of annotated objects in RADIal. (Right) Scatter plot of per-frame selected chirp count vs. object velocity. The absence of correlation confirms that adaptive stopping is stability-driven, not velocity-driven.

Class	AP	AR	F1	Chamfer↓
Person (Pedestrian)	96.3	94.2	95.2	0.085
Vehicle (Car)	98.4	97.2	97.8	0.082

Table 5. **Class-wise detection and localization on RaDICaL.** AP and AR are reported in %; Chamfer distance (lower is better) is in metres.

6.4. Multi-Task vs. Task-Specific Performance

Joint training does not introduce gradient interference. RAVEN trained jointly outperforms task-specific single-head baselines on both objectives: detection (0.95 vs. 0.93 mAP) and segmentation (90.2% vs. 90.1% mIoU). We attribute this to the shared chirp-SSM backbone learning complementary spatial features that benefit both heads simultaneously.

References

[1] Hemant Kumawat and Saibal Mukhopadhyay. Radar guided dynamic visual attention for resource-efficient rgb object detection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. 1

[2] 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. 1, 2

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. 1

[4] 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 17000–17009, 2022. Paper: https://doi.org/10.1109/CVPR52688.2022.01651. Dataset: https://github.com/valeoai/RADIal. 1, 2

[5] Anuvab Sen, Mir Sayeed Mohammad, and Saibal Mukhopadhyay. Ssmradnet : A sample-wise state-space framework for efficient and ultra-light radar segmentation and object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4365–4374, 2026. 6

[6] Sudarshan Sharma, Hemant Kumawat, and Saibal Mukhopadhyay. Chirpnet: Noise-resilient sequential chirp-based radar processing for object detection. In *IEEE International Microwave Symposium*, 2024. 2

[7] 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

[8] 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