AdaRadar: Rate Adaptive Spectral Compression for Radar-based Perception

Supplementary Material

Jinho Park[1] Se Young Chun[2] Mingoo Seok[1]
[1]Columbia University [2]Seoul National University
jp4327@columbia.edu, sychun@snu.ac.kr, ms4415@columbia.edu

jp4327.github.io/adaradar/

The Supplementary Material is organized as follows:

- Section S1 provides implementation details, including network architectures, training protocols, hardware resources, and asset licenses.
- Section S2 elaborates on notations for transform and alternative task objectives.
- Section S3 presents extended quantitative evaluations on the adaptive control, covering ablation studies on hyperparameters and surrogate objective, and stability analysis.
- Section S4 presents additional quantitative evaluations on the spectral compression, covering quantization and block length modulation.
- Section S5 presents additional baselines, including compression using autoencoders, H.264, Gumbel-sigmoid method, and CFAR algorithm.
- Section S6 illustrates the qualitative impact of AdaRadar through range-Doppler maps and visual performance comparisons.

S1. Experiment setup details

S1.1. Baseline specifications and architectures.

The baselines in Table 1–3, refer to the corresponding vanilla models without any compression. We demonstrate AdaRadar's compression capability across three datasets with distinct model architectures to showcase the *robustness* of our approach.

- The baseline, FFTRadNet [3] (Table 1), detects vehicles and free driving spaces by using a learned MIMO pre-encoder, a feature pyramidal network (FPN) encoder, and separate detection and segmentation heads.
- TMVA-Net [10] in Table 2 performs four-class radar semantic segmentation with pedestrian, cyclist, car, and background. It first reshapes the range-azimuth-Doppler (RAD) tensor into azimuth-Doppler (AD), range-Doppler (RD), and range-azimuth (RA) views, then applies 3D convolutions within view-specific encoders.
- Radatron [11] (Table 3) also runs parallel FPN branches on high- and low-resolution radar streams, which are fused later to output vehicle detection maps.

S1.2. Experimental details

For Table 1, we perform adaptation with a task objective same as Eq. 2 with $r_{\text{initial}} = 12$, $\eta = 1.0$, $\epsilon = 0.05$, $p_{\text{threshold}} = 0.8$, $\lambda = 1.0$, $\nabla_{\text{clip}} = 1.0$.

For Figure 6, we experiment with a task objective same as Eq. S4 with $r_{\text{initial}} = 20$, $\eta = 1.0$, $\epsilon = 0.05$, $p_{\text{threshold}} = 0.9$, $\lambda = 15.0$, $\nabla_{\text{clip}} = 1.0$.

S1.3. Additional training details and resources

Training details on TMVA-Net. All experiments are implemented in PyTorch and trained on a single Nvidia RTX A6000 GPU. We follow the setting as per the original paper with the Adam optimiser[44], using its default hyperparameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$). Training proceeds for 300 epochs with a mini-batch size of 6. The initial learning rate is set to 10^{-4} and decays by a factor of 0.1 every 20 epochs via a StepLR scheduler.

Training details on Radatron. We abide by the original training recipe and train the model using stochastic gradient descent (SGD) with a base learning rate of 0.01 on a single Nvidia RTX A6000 GPU. The learning rate decays by a factor of 0.2 at 15k and 20k iterations, following a step schedule. Training runs for 25k iterations in total, with a mini-batch size of 8 images. All other training hyperparameters follow the default settings in Detectron2.

Computation resources. The experiments on the RADIal dataset are run on a Red Hat Enterprise Linux 8 workstation equipped with a single 3.5 GHz Intel Core i9-9920X processor, four NVIDIA RTX 6000 GPUs (24 GB VRAM each), 128 GB of system RAM. The rest of our experiments are conducted on a Red Hat Enterprise Linux 8 server equipped with dual Nvidia RTX A6000 GPUs, each equipped with 48GB VRAM, an AMD Ryzen Threadripper PRO 3995WX CPU @ 4.2GHz, and 256GB of RAM.

S1.4. Assets

License for RADIal. The RADIal dataset[1] [3] is released without an explicit license; we cite the source and use it solely for non-commercial academic research, in accordance with standard scholarly practices.

[1] https://github.com/valeoai/RADIal

License for CARRADA. The CARRADA dataset [10] is published under the CC BY-NC-SA 4.0 License, and all CARRADA code[2] is released under the GNU General Public License v3.0 (GPL-3.0).

License for Radatron. The Radatron dataset[3] [11] is released under the Apache License 2.0.

S1.5. Discussions

Broader impacts. By reducing data transmission and memory access, our method could reduce energy usage, contributing to sustainability in high-throughput ML systems. Radar-based systems improve robustness under adverse weather or poor lighting conditions, potentially reducing accidents in autonomous and assisted driving.

S2. Additional notes on proposed method

S2.1. Discrete cosine transform

Type-II DCT computation is described by $\boldsymbol{z}_{c,b} = \mathbf{a} \odot \mathbf{G}\tilde{\mathbf{x}}_{c,b}$ where $\tilde{\mathbf{x}}_{c,b} \in \mathbb{R}^{M^2}$ is a flattened feature map, $\odot$ denotes Hadamaard (element-wise) product, $\mathbf{G} \in \mathbb{R}^{M^2 \times M^2}$ describes cosine coefficients, and $\mathbf{a} \in \mathbb{R}^{M^2}$ is a normalization vector. $\mathbf{G}$ is computed by $G_{Mu+v,Mi+j} = \cos\left(\frac{(2i+1)u\pi}{2M}\right)\cos\left(\frac{(2j+1)v\pi}{2M}\right)$ with indices $i, j, u, v \in \{0, \ldots, M-1\}$. $\mathbf{a}$ has entries

$$a_{Mu+v} = \frac{2}{M}\,\alpha(u)\,\alpha(v) \tag{S1}$$

$$\alpha(u) = \begin{cases} 1/\sqrt{2} & u = 0 \\ 1 & \text{otherwise} \end{cases} \tag{S2}$$

$$u, v \in \{0, \ldots, M-1\}. \tag{S3}$$

S2.2. Alternative task objective

Beyond the objective in Eq. 2, we define a constraint-aware gradient

$$\nabla_r J = (p - p_{\min}) + \hat{\nabla}_r h \cdot (r - r_{\min} + \lambda). \tag{S4}$$

In this formulation, $p_{\min}$ and $r_{\min}$ denote the lower bounds for confidence and pruning ratio, respectively, while $\hat{\nabla}_r h$ represents the zeroth-order gradient approximation. The hyperparameter λ regulates the Pareto optimal trade-off between task accuracy and transmission bandwidth.

[2] https://github.com/valeoai/carrada_dataset

[3] https://github.com/waleedillini/radatronDataset

We employ confidence thresholding and gradient clipping to facilitate more stable adaptation. Specifically, gradient clipping constrains the estimated gradient $\hat{\nabla}_r h(\mathbf{x}, r)$ within a predefined range $[\nabla_{\min}, \nabla_{\max}]$, while confidence thresholding skips the adaptation process for the given timestep if $p < p_{\text{threshold}}$.

S3. Extended quantitative analysis on adaptive control back end

S3.1. Compression/decompression latency

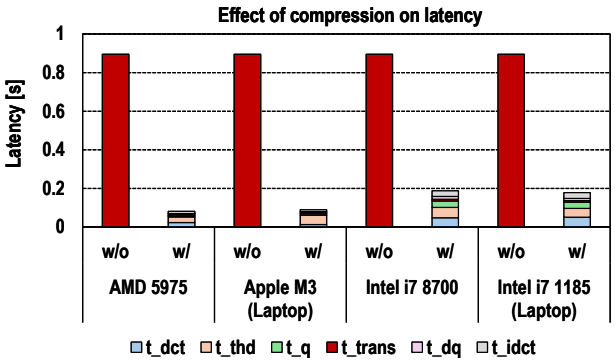

Figure S1. **Compression, decompression, communication cost analysis.** Our compression and decompression processes significantly reduce latency compared to the baseline system, which transmits the original radar tensor.

In Figure S1, we observe that compression and decompression alleviate the latency required to transfer radar data. This is because sensor-to-compute links, such as a CAN bus in automotive systems, have limited bandwidth for transferring high-fidelity data. Our method can easily re-utilize the embedded DSP block, which is used to compute the fast Fourier transform along range and Doppler dimensions in MIMO radars [5]. The radar side only requires simple operations (e.g., DCT, sorting, and rounding) to support AdaRadar.

The total latency arises from three components: (1) compression, (2) transmission, and (3) decompression. More concretely,

$$t_{\text{overall}} = \underbrace{t_{\text{dct}} + t_{\text{thd}} + t_{\text{q}}}_{\text{coder}} + t_{\text{trans}} + \underbrace{t_{\text{dq}} + t_{\text{idct}}}_{\text{decoder}}. \tag{S5}$$

In this experiment, compression and decompression latencies are measured on machines with different CPUs by compressing and decompressing 200 radar tensors of shape 512×256×32 from the RADIal test set. Transfer time is estimated by assuming a radar [5] followed by a serial link based on MIPI CSI2 with 150 Mbps bandwidth, sending the radar tensor in a burst mode.

This is analogous to loading a RAW image versus a JPEG-compressed image. Loading a RAW format takes longer because the loading time is bottlenecked by data *transmission* rather than by decompression. A typical JPEG decoder [47] takes about one millisecond to decompress a 512×512 image, while loading the uncompressed image from a CAN bus [6] with a bandwidth of a few MB/s would take dozens of milliseconds.

S3.2. Ablation study

Surrogate objective ablation. We use bounding box confidence as a proxy for task performance, assuming that regions with high detector confidence should be preserved more faithfully. To justify our choice, we show additional empirical analysis showing how confidence best correlates with the task performance under compression, and (2) add ablation studies where we replace confidence with alternative signals such as range and azimuth location variances among bounding box predictions per object.

Table S1. **Correlation between OD prediction output and performance metrics.** The performance metrics, such as AP and AR, are highly correlated with confidence but not with other objectives.

Surrogate objective	Corr. w/ AP	Corr. w/ AR
Confidence (Ours)	**0.841**	**0.857**
Range stdev.	-0.129	-0.045
Azimuth stdev.	-0.070	-0.016

In Table S1, we demonstrate that the bounding box confidence correlates most closely with the average precision (AP) and average recall (AR). On the other hand, the range and azimuth exhibit a weak (negative) correlation. We take the standard deviation of the range and azimuth values of predicted bounding boxes.

In Table S2, we ablate the surrogate objective with range and azimuth deviations and a random variable $p \sim U(0, 1)$, which are fed into our zeroth-order optimizer. We find that the compression ratio fluctuates widely, unlike the confidence metric, whose standard deviation across rate sequences remains less than one. This also results in severe performance degradation.

Hyperparameter ablation. We vary the regularization weight λ to explore how it governs the trade-off between compression efficiency and downstream performance in detection and segmentation. The experiments are conducted on the RADIal dataset using the FFTRadNet model (Table S3). We apply 4-bit quantization with a fixed block length of $M = 64$. The pruning ratio is initialized to 12, and the learning rate is set to $\eta = 1$. As λ increases, the system adjusts the pruning strength, balancing lower bit rates with the preservation of task-relevant features. Each configuration corresponds to a Pareto-optimal point that jointly optimizes compression and task performance.

Proxy gradient stability. Extreme fluctuation scenarios induced by high-speed motion or severe occlusion are unfortunately not present in the open-sourced datasets. We also believe that occlusion is an unrealistic scenario for radars: (1) a car would be almost crashing into another for a radar having a wide azimuth field of view (FoV) of 140°, and (2) a radar is unaffected by rain or fog, unlike a camera.

We therefore synthetically generate such extreme fluctuation scenarios by adding white noise to the radar tensor and present the results in Table S4. We mimic scenarios with mild, moderate, and severe conditions by modulating the noise power, which is scaled linearly with the noise scale factor. In this experiment, we confirm that AdaRadar *automatically* reduces the compression ratio under harsher conditions by monitoring the object-detection bounding-box confidence. We also compare the results with the vanilla model, which does not employ any compression. We find that our compression scheme achieves over 25× compression while maintaining the detection performance in those worse conditions.

S4. Extended quantitative analysis on spectral compression front end

S4.1. Effect of quantization on downstream tasks

Effect of quantization on RADIal. We evaluate how different quantization bit-widths affect both compression–decompression efficiency and network accuracy, and we summarize the resulting design choices. Figure S5 shows the average precision, average recall, and F_1 score roll-off for different bit-rates using the FFTRadNet network on the RADIal *test* dataset *without* any fine-tuning and block length $M = 64$. We observe that 4-bit quantization achieves equal or higher performance compared to bit widths of 8 and 16, and thus we adopt 4-bit quantization.

Effect of quantization on CARRADA. Figure S6 depicts how mIoU and mDice vary as the bit-rate decreases, which is the case for the CARRADA dataset with the TMVA-Net. Performance boosting until 5× pruning is pronounced. We select 8-bit quantization because it delivers performance similar to that of 16-bit quantization. We report that 4-bit symmetric quantization underperforms, and we use the block length $M = 16$ in the above quantization cases.

Effect of quantization on Radatron. Figure S7 reveals how detection metrics evolve as the bit-rate is lowered for the Radatron dataset using the baseline Radatron model. We also observe performance boosting until 7.5× pruning, especially for AP_{75}. We choose 8-bit quantization despite 4-bit quantization giving similar performance due to the consid-

Table S2. **Surrogate objective ablation with range standard deviation, azimuth standard deviation, and a random variable.** We observe that different choices of surrogate objective do not correlate well with the downstream task performance, leading to severe degradation in the perception performance. In those cases, such a significant drop in F_1 score bears similar performance to the random variable.

Surrogate objective	CR ↑	CR stdev. ↓	Precision ↑	Recall ↑	F_1 ↑	mIoU ↑
Confidence (Ours)	6.58	**0.91**	**0.969**	**0.948**	**0.958**	**0.796**
Range stdev.	60.9	22.1	0.941	0.785	0.856	0.721
Azimuth stdev.	**87.0**	9.8	0.930	0.762	0.837	0.708
Random	34.1	29.7	0.945	0.834	0.886	0.741

Table S3. **Hyperparameter ablation with λ.** We analyze the trade-off between bit rate, compression ratio, and downstream performance on detection and segmentation through modulation of λ. 'PR', 'BR', and 'CR' stand for prune ratio, bit rate, and compression ratio, respectively. We use 4-bit quantization with a block length of $M = 64$. The pruning ratio is initially set to 12, and the learning rate is configured as $\eta = 1$.

λ	PR ↑	BR [bpp] ↓	CR ↑	Precision (%) ↑	Recall (%) ↑	F_1 (%) ↑	mIoU (%) ↑
1	12.57	0.32	100.56	96.25	94.03	95.13	79.34
3.3	13.83	0.29	110.68	96.21	93.87	95.03	79.21
10	16.09	0.25	128.74	96.07	93.64	94.84	79.12
33	20.75	0.19	166.03	95.70	93.07	94.36	78.87
100	27.91	0.14	223.29	95.48	91.66	93.53	78.18

eration for the block length. We explain this in the following section.

S4.2. Effect of block length on downstream tasks

Effect of block length on RADIal. We examine the impact of varying block sizes on both compression–decompression performance and downstream network accuracy, and draw design choices. Figure S8 presents how average precision, recall, and F_1 score degrade across block sizes $M \in \{8, 16, 32, 64\}$ when evaluated on the RADIal *test* set using FFTRadNet, with 4-bit quantization and no fine-tuning. Among these, $M = 64$ yields the best trade-off between performance and overhead from the scaling factor.

We hypothesize that larger DCT blocks yield better performance because they aggregate more spatial information, leading to smoother and more compressible spectral representations. In contrast, smaller blocks may introduce higher variance across patches, making the compression less stable and more lossy. Also, smaller blocks may experience more fragments on the edges at a high pruning ratio.

Effect of block length on CARRADA. We evaluate how block length affects segmentation performance on the CARRADA dataset by varying $M \in \{8, 16, 32\}$ and monitoring mIoU and mDice scores. Fig S9 shows that $M = 32$ offers a favorable trade-off between accuracy and overhead. Additionally, we report that using $M = 64$ without fine-tuning leads to noticeably lower performance.

Effect of block length on Radatron. We assess how varying block sizes influence detection performance on the Radatron dataset by testing $M \in \{8, 16, 32, 64\}$. Figure S10 indicates that $M = 64$ results in relatively poor performance, while $M = 32$ provides a more favorable trade-off.

S5. Extended baseline comparisons

S5.1. Comparison with image domain compression methods, Gumbel-sigmoid, and CFAR

Autoencoder. Image-domain methods exhibit a performance gap due to the extreme sparsity of radar data, which contrasts with the dense spatial information of optical imagery. Table S5 compares the compression performance of the autoencoder (AE) architecture by Ballé *et al*. [41], implemented with the CompressAI library [48]. The results indicate a substantial $54\times$ reduction in radar data volume. However, the performance degrades by a larger amount than AdaRadar after compression by two orders of magnitude.

We trained the AE on the RADIal training set for epochs up to 300 with a learning rate of 10^{-4} and evaluated its impact by integrating it into the FFTRadNet pipeline. We treat each radar channel arising from Rx-Tx pairs independently. meaning the channel dimension is flattened with the batch dimension. The bit rate is measured directly from the bottleneck latent code. Since networks optimized for specific λ_{ae} hyperparameters are trained independently, they are fundamentally *static*. Thus, these AEs lack the capacity for dynamic bit-rate modulation during inference. The AE is comparable in size (3M parameters) to the perception network itself, undermining the objective of lightweight compression.

H.264. We evaluate FFTRadNet using raw radar data compressed via the H.264 codec from the ffmpeg library

Table S4. **Proxy gradient stability at varying additive Gaussian noise levels on the RADIal test set.** We test the stability of our feedback loop by adding Gaussian noise to the radar input (RD map). Even under the noisy scenario, our adaptive control-based compression demonstrates similar performance to the non-compressed baseline.

Compression	Condition	Noise Scale	Comp. Ratio ↑	Prun. Ratio ↑	Precision ↑	Recall ↑	F_1 ↑	mIoU ↑
✗ No	Mild	1	1.0	1.0	0.973	0.922	0.947	0.774
✓ Yes	Mild	1	36.8	4.6	0.969	0.910	0.939	0.777
✗ No	Moderate	2	1.0	1.0	0.957	0.884	0.919	0.694
✓ Yes	Moderate	2	33.7	4.2	0.967	0.860	0.910	0.692
✗ No	Severe	5	1.0	1.0	0.931	0.771	0.844	0.618
✓ Yes	Severe	5	25.6	3.2	0.939	0.720	0.815	0.602

Table S5. **Benchmark with extended baselines.** We compare radar compression with autoencoders, H.264, Gumbel-sigmoid-based pruning, and CFAR on RADIal.

Methods	Hyperparameters	BR [bpp] ↓	CR ↑	Precision (%) ↑	Recall (%) ↑	F_1 (%) ↑	mIoU (%) ↑
Baseline	-	32	1×	97.24	95.93	96.58	75.97
Ours	$s_{\text{FXP}} = 4$-bit	0.32	100.5×	96.25	94.04	95.13	79.34
AE	$\lambda_{ae} = 1.0 \cdot 10^{-3}$	1.66	19.2×	97.20	95.67	96.43	75.45
AE	$\lambda_{ae} = 3.3 \cdot 10^{-4}$	1.07	29.75×	97.22	94.85	96.02	74.42
AE	$\lambda_{ae} = 1.0 \cdot 10^{-4}$	0.59	54.1×	97.40	92.82	95.05	70.80
AE	$\lambda_{ae} = 3.3 \cdot 10^{-5}$	0.27	117.1×	97.49	77.05	86.08	61.04
H.264	crf $= 0$	7.8	4.08×	98.49	76.41	86.05	66.48
H.264	crf $= 10$	5.4	5.91×	98.44	76.28	85.95	66.43
H.264	crf $= 20$	3.6	8.87×	98.33	75.76	85.58	66.26
Gumbel	$\lambda_s = 0.03$	13.86	2.3×	96.37	96.89	96.63	75.17
Gumbel	$\lambda_s = 0.1$	8.98	3.6×	94.33	97.13	95.71	73.59
Gumbel	$\lambda_s = 0.3$	6.68	4.75×	91.24	96.23	93.67	70.40
Gumbel	$\lambda_s = 1.0$	2.83	11.3×	84.91	90.91	87.80	57.57
CFAR	$thd = 10^{0.05}$	7.68	4.17×	98.14	71.54	82.76	61.10
CFAR	$thd = 10^{0.2}$	5.56	5.76×	98.02	63.14	76.81	55.72
CFAR	$thd = 10^{0.5}$	2.29	14.0×	98.78	42.77	59.69	47.13

[49] (Table S5). To ensure robust scaling, we clip the radar distribution per channel at the 2nd and 98th percentiles, as standard min-max normalization yields inferior results. The compression is implemented by reformatting the radar channel dimension into a temporal sequence of 8-bit grayscale frames. We observe that H.264-based compression falls short in terms of downstream task performance because raw radar data is highly sparse and exhibits a wide dynamic range.

Soft gradient with Gumbel-sigmoid. We employ a Gumbel-sigmoid [50, 51] with $M^2 = 64^2$ learnable parameters to estimate soft gradients for pruning DCT coefficients. In Table S5, perception performance remains stable up to a compression factor of 2.3×. We attribute this robustness to the Gumbel-sigmoid acting as a static mask during inference. While dynamic pruning via an auxiliary network is possible, it would require transmitting gradients for the entire set of DCT coefficients, thereby posing a communication bottleneck unsuitable for our compression objectives.

We train the Gumbel-sigmoid layer for epochs up to 100 with a learning rate of 10^{-2}. During training, the detection backbone remains frozen while we optimize a composite loss $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_s \mathcal{L}_{\text{prune}}$ that combines perception task and pruning objectives.

CFAR. Although CFAR [27, 28] facilitates peak detection in radar images, its lossy nature prevents the reconstruction of the raw radar tensor. We assume that maintaining these peak values provides a viable basis for compression. We perform compression analogous to the index-value-based method, in which the non-zero elements are determined by the CFAR algorithm rather than by the magnitude alone. In Table S5, we use a Cell-Averaging CFAR detector with window and guard widths of $(9, 3)$, evaluating peak-based compression across varying SNR thresholds (thd).

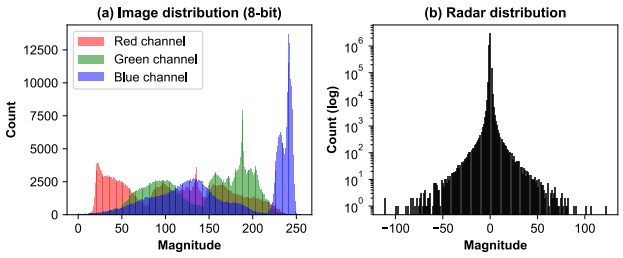

Figure S2. **Difference in distribution between image and radar tensors**. This illustrates that while natural (a) images span a broad dynamic range, (b) radar signals exhibit extreme sparsity and a heavy-tailed profile (1st/99th percentiles at ± 1.74), necessitating specialized compression beyond standard image-domain codecs.

S5.2. Distribution comparison between image and raw radar tensor

Figure S2 illustrates the stark contrast between the distributions of natural images and radar tensors. While the image data is distributed broadly across the 8-bit dynamic range, the radar data exhibits extreme sparsity. Here, the image is of size $3 \times 512 \times 768$ while the radar is $32 \times 512 \times 256$. Specifically, the radar signal is highly concentrated, with the 1st and 99th percentiles residing at -1.74 and 1.74, respectively. This statistical discrepancy highlights why standard image-domain compression codecs are ill-suited for radar data without domain-specific adaptation.

S5.3. Compression after learning-based FFT

We demonstrate the efficacy of AdaRadar on learning-based FFT in Table S6. T-FFTRadNet converts raw ADC inputs into spectral form via FourierNet, which performs *learned* FFT. Its Swin Transformer backbone uses this spectral representation to perform object detection. We integrate AdaRadar with 4-bit quantization and patch size $M = 8$ after the FourierNet, which performs learnable 2D-FFT from raw ADC outputs. It compresses the radar tensor by a factor of 32× with negligible loss in accuracy (˜1%), *without* retraining.

S6. Qualitative results and visualizations

S6.1. Raw radar data under compression

Range-doppler map visualization. Figure S3 compares the range-Doppler (RD) maps before compression with those reconstructed after the full compression–decompression cycle, and adds magnified patches for each map at several signal-to-noise ratio (SNR) levels. Figure S3(a) displays the original RD map $\mathbf{x}$, whereas Fig. S3(b) shows its reconstruction $\hat{\mathbf{x}}$ after the compression–decompression process with 8-bit quantization at a pruning ratio of 1. Each image visualizes the log-power spectrum $\log \|\mathcal{X}_c\| = \log \|\mathbf{x}_{2c} + j\mathbf{x}_{2c+1}\|$ where the radar's complex feature tensor $\mathcal{X} \in \mathbb{C}^{C \times H \times W}$ is decomposed into its real and imaginary parts and concatenated along the channel dimension to form the real-valued tensor $\mathbf{x} \in \mathbb{R}^{2C \times H \times W}$. Figure S3(c) zooms in on a 64×64 patch taken from the reconstructed RD map shown in Fig. S3(b). Figure S3(d)-(f) respectively correspond to the same patch with different pruning ratios of $\{5, 10, 20\}$.

Figure S11 provides additional qualitative results to showcase the effect of compression on the RD map and the corresponding radar perception outputs. Results on arbitrary samples consistently indicate that object detection or segmentation outputs perform comparably to the non-compressed baseline.

SNR against compression. Figure S3(g) plots the average SNR against bit-rate for 100 RD maps sampled from the RADIal *test* set. The SNR falls steeply as the pruning ratio rises from 1× to 5×. The curve flattens beyond 5×, indicating only marginal degradation.

RAE against compression. We computed the mean and maximum relative absolute errors (RAE) between the original and the reconstructed power spectrums (Table S7). Here, RAE $= \frac{|x-\hat{x}|}{|x|}$ where x and $\hat{x}$ respectively are the original and reconstructed spectrums, and both mean and max RAE are averaged across 100 radar tensor samples from the RADIal test set. While the maximum RAE saturates at approximately 0.38 beyond the compression ratio of 10, the mean RAE grows sub-linearly with a greater compression ratio. These results demonstrate the robustness and effectiveness of our compression method in preserving spectral fidelity even at high compression rates.

S6.2. Remarks on frequency components

Our method *adaptively* selects coefficients within each patch based on local energy, making it broadly applicable across different sensor types and downstream tasks. The frequency components are determined primarily by the radar sensor characteristics and the perceived scene. We provide further remarks on the distribution of frequency coefficients in the 2D map.

RADIal. In Figure 4(a), when we experiment with radar tensors from the RADIal dataset, the magnitude of the DCT coefficients is concentrated around the high-frequency components, with peak energy at $(x, y) = (63, 63)$ and decreasing radially. Taking a horizontal slice at $y = 63$ or a vertical slice at $x = 63$, we observe a linearly increasing ramp in frequency towards the end.

CARRADA. We observe that the energy is centered mostly at the low-frequency components, exhibiting high-intensity values at the left and top edges, i.e., $x = 0$ and $y = 0$. This suggests that the radar sensors capture smooth horizontal

Table S6. **Compression after learned FFT.** We demonstrate the effectiveness of AdaRadar even with *learned* FFT. We use T-FFTRadNet for detection and segmentation using the RADIal test set. Our compression method achieves 32× data reduction with negligible performance loss. (PR: prune rate; BR: bit rate; CR: compression rate.)

Methods	**Bit** ↓	**PR** ↑	**BR [bpp]** ↓	**CR** ↑	**Precision** (%) ↑	**Recall** (%) ↑	**F_1**(%) ↑	**mIoU**(%) ↑
Baseline [52]	32	1	32	1×	90.61	93.67	92.11	80.56
Ours	4	2	2	16×	90.94	93.19	92.05	80.34
Ours	4	4	1	32×	90.55	91.89	91.22	79.56

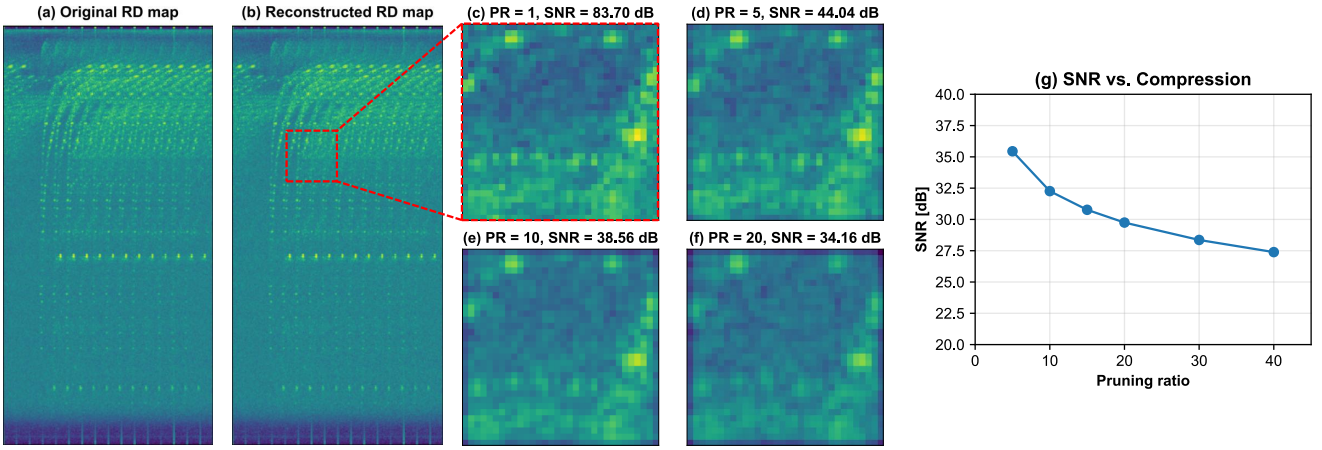

Figure S3. **Qualitative visualization on range-Doppler map.** (a) Original and (b) reconstructed RD map for a single channel with 8-bit quantization and a pruning ratio of 1. (c)-(f) Magnified 64×64 patches at pruning ratios of $\{1, 5, 10, 20\}$, respectively. (g) SNR vs. bit rate trade-off where numerical values are listed in Table S7.

Table S7. **SNR & RAE against compression.** Mean and maximum relative absolute errors (RAE) at different compression ratios.

Comp. Ratio ↑	**SNR [dB]** ↑	**RAE_{mean}** ↓	**RAE_{max}** ↓
1	144.17	0.000	0.000
5	35.45	0.010	0.329
10	32.26	0.017	0.378
15	30.77	0.022	0.389
20	29.75	0.025	0.390
30	28.36	0.031	0.383

and vertical patterns in the range Doppler map.

Radatron. The pattern is also similar for the Radatron dataset, wherein we discover the energy mainly being concentrated in the low-frequency bins, creating high-intensity bar shapes at the left and top edges.

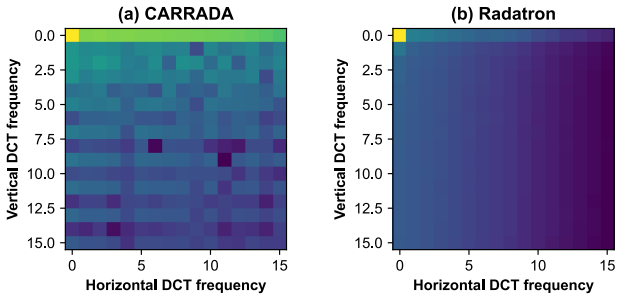

Figure S4. **DCT coefficient magnitudes.** (a) CARRADA and (b) Radatron datasets. Unlike the RADIal dataset, the spectral bins are more focused on low-frequency bins. Our adaptive pruning is agnostic to such spectral structures, since it ranks coefficients by energy.

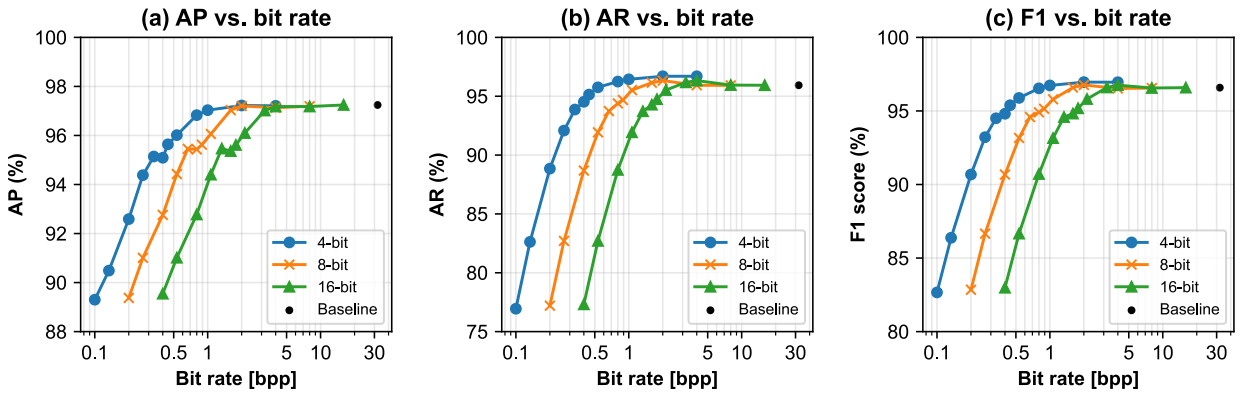

Figure S5. **Effect of quantization bit width on RADIal.** (a) Average precision, (b) average recall, and (c) F_1 score against the bit rate modulation from pruning. We observe that quantization up to 4 bits does not affect the performance compared to that of 8-bit and 16-bit. We use the FFTRadNet on the RADIal dataset *without* any fine-tuning with the block length $M = 64$ for all cases.

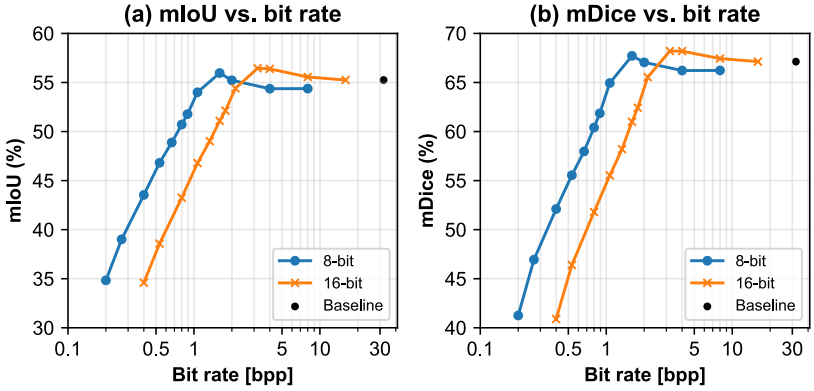

Figure S6. **Effect of quantization bit width on CARRADA.** (a) mIoU and (b) mDice reported versus the bit rate reduction. We observe that quantization up to 8 bits does not affect the performance compared to that of 16-bit. We use the TMVA-Net on the CARRADA dataset *without* any fine-tuning with the block length $M = 16$ for all cases.

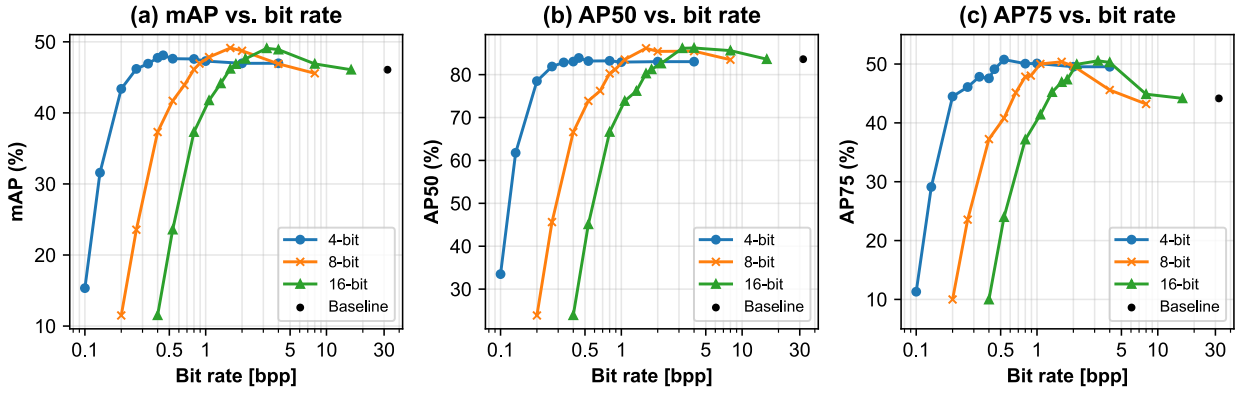

Figure S7. **Effect of quantization bit width on Radatron.** (a) mAP, (b) AP_{50}, and (c) AP_{75} vs. the bit rate. We observe that quantization up to 8 bits does not affect the performance compared to that of 4-bit and 16-bit. We use the baseline Radatron model on the Radatron dataset *without* any fine-tuning with the block length $M = 16$ for all cases.

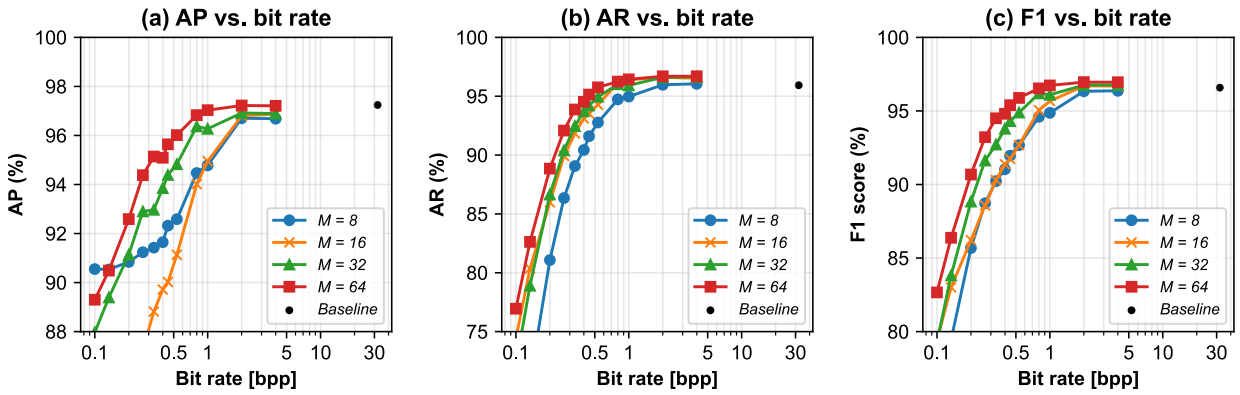

Figure S8. **Effect of block length on RADIal.** (a) Average precision, (b) average recall, and (c) F_1 score against the block length modulation. We observe that compression with a larger block length generally performs better. We use the FFTRadNet on the RADIal dataset *without* any fine-tuning with 4-bit quantization for all cases.

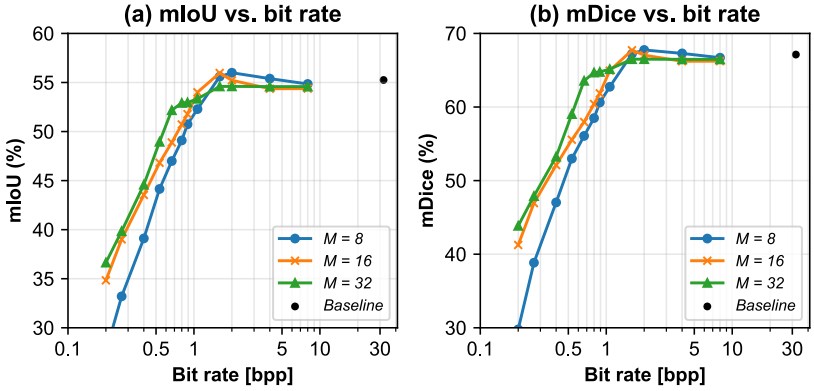

Figure S9. **Effect of block length on CARRADA.** (a) mIoU and (b) mDice reported versus the change in block length. We observe that a larger block is preferred for a higher pruning ratio and a smaller block size for a lower pruning ratio. We use the TMVA-Net on the CARRADA dataset *without* any fine-tuning with 8-bit quantization for all cases.

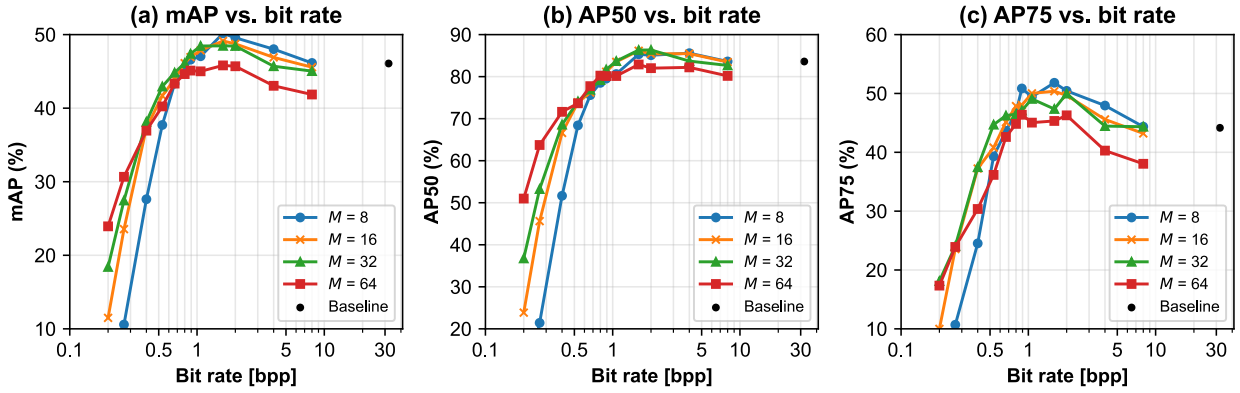

Figure S10. **Effect of block length on Radatron.** (a) mAP, (b) AP_{50}, and (c) AP_{75} vs. block size change. We observe that a larger patch performs better for higher pruning ratios. We use the baseline Radatron model on the Radatron dataset *without* any fine-tuning with 8-bit quantization for all cases.

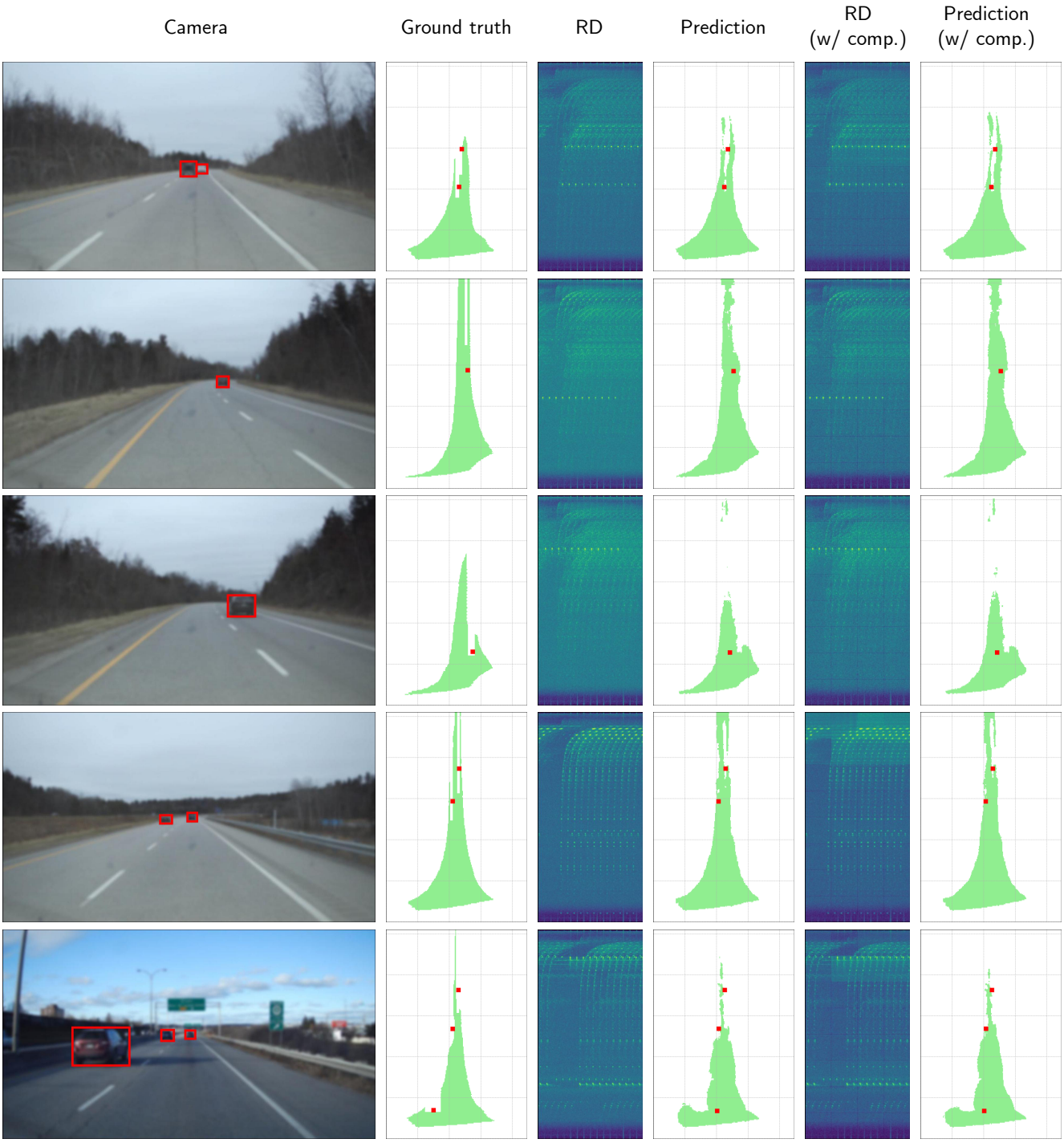

Figure S11. **Additional qualitative comparisons.** Camera images only provide contextual reference for the scene. RD images map Range to the y-axis and Doppler to the x-axis. We highlight the detected car in red and the segmented map in green in the bird's-eye-view visualization.