Rethinking Radar 3D Perception: Representation Learning from Raw Spectra

Miao Zhang[1,2]* **Mohamed Abdelsamad**[1,3]* **Bin Yang**[1] **Sherif Abdulatif**[1]
Benedikt Loesch[1] **Marco Altmann**[1] **Abhinav Valada**[3] **Bin Yang**[2]

[1]Bosch Research [2]University of Stuttgart [3]University of Freiburg
{miao.zhang5,mohamed.abdelsamad,bin.yang3}@de.bosch.com
valada@cs.uni-freiburg.de bin.yang@iss.uni-stuttgart.de

Abstract

Raw radar spectra are widely viewed as too cluttered and noisy to support direct representation learning. This has led to handcrafted reductions, sparse point clouds, or compact Doppler descriptors as the default preprocessing step in radar perception. In this work, we challenge this assumption and argue that the main limitation is not the raw spectral representation itself, but rather the absence of a learning objective capable of separating the structured signal from noise. Handcrafted reductions partially address this problem, but they impose rigid assumptions about what counts as signal, discarding weak but informative structure while retaining artifacts that match the reduction model. Once such information is removed, supervised learning cannot recover it or question what was preserved. We propose a principled alternative: structure is the component of the radar spectrum that remains reconstructable under localized corruption, while noise is the component that does not. This perspective makes self-supervised masked modeling a natural learning paradigm for dense radar spectra, without imposing a predefined signal–noise separation. We instantiate it with Leakage-aware Spectral MAE, a masked autoencoding framework for raw radar cubes that incorporates a corruption strategy inspired by the cross-bin energy spreading caused by spectral leakage. We also show that Doppler organization is crucial: treating Doppler as a primary spatial dimension, using an x-y-D representation with height encoded as features, substantially outperforms naive channel stacking and is essential for exploiting dense 4D spectra. Without any handcrafted reductions, our approach consistently outperforms state-of-the-art radar-based detectors and point-cloud-based self-supervised learning (SSL) counterparts. These results establish raw radar spectra as a highly effective representation for 3D perception when paired with an appropriate pretraining objective.

1 Introduction

Radar sensors are becoming increasingly important to autonomous driving as they remain reliable under adverse weather and lighting conditions, where cameras and LiDARs often degrade [1]. However, the rich frequency-domain spectra produced by radar hardware are rarely used directly in perception pipelines. Instead, they are first compressed through handcrafted reductions such as CFAR detection [2], which collapses them into sparse point clouds, or compact statistics that summarize their Doppler content [3, 4]. Modern radar perception systems are then developed around these reduced representations [5–8], and substantial effort has gone into refining this paradigm [9–11].

This pipeline has an important but largely unexamined cost. Handcrafted reductions do not merely suppress noise but they *define* what will be treated as noise before learning begins. Weak reflections

*These authors contributed equally.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

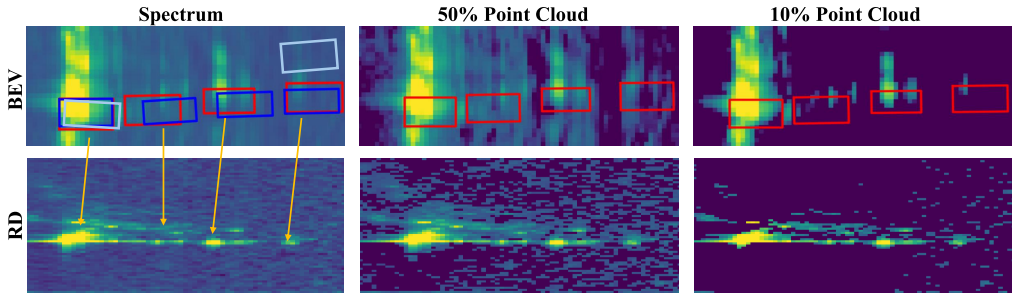


Figure 1: Reducing dense spectra into sparse point representations suppresses both structured signals and noise simultaneously, which may remove informative spectral structures and degrade perception performance. In contrast, dense spectra preserve the full spectral information, whereas SSL enables the model to better exploit useful structures in noisy observations. Red: ground truth, blue: prediction by spectrum-based SSL, light blue: prediction by point-cloud-based SSL. RD: Range-Doppler view. BEV: Bird's-eye view.

that fall below CFAR thresholds are discarded. Doppler structure not captured by summary statistics is discarded. The resulting representation is therefore not simply a cleaner version of the radar measurement. It is the subset of the measurement that survives a fixed set of assumptions about what signal should look like. Any weak, distributed, or noncanonical structure that violates these assumptions is removed, even if it is informative for downstream perception.

A natural alternative is to learn directly from unreduced post-FFT radar spectra, which we refer to as raw spectra because they preserve the spectral information otherwise discarded by handcrafted reductions. However, prior attempts have not established dense spectra as a clearly superior representation. Supervised learning on fully dense spectra under realistic driving conditions does not reliably outperform reduced inputs [12], a finding we also reproduce. This outcome is often interpreted as evidence that raw spectra are too noisy to support direct learning. We argue that the failure lies in the objective, not the input. Supervised perception losses, such as those based on bounding boxes or semantic labels, are defined over the task's output space. They provide only indirect supervision for how the model should interpret the input. When trained on dense spectra, a supervised model is exposed to all components that handcrafted pipelines would have removed, but receives no explicit signal about which spectral structures are coherent and which are incidental.

This suggests that dense radar perception requires an objective that reasons about structure in the input space itself, without presupposing the morphology of signal or noise. We propose a novel definition: *signal is what remains reconstructable under localized corruption, and noise is what does not*. An objective built on this definition requires no prior on signal or noise morphology. It discovers signal boundaries from the data itself. This is related to the $\mathcal{J}$-invariant framework of [13], but the problems are fundamentally different. Those methods assume an underlying clean signal and use self-supervision to recover it from noisy observations. Raw radar spectra have no clean counterpart. Our goal is not to recover a hidden denoised spectrum, but to learn which structures in the observed spectrum are coherent enough to support prediction. This view makes self-supervised masked modeling a natural learning paradigm for dense radar spectra.

We instantiate this principle by introducing **Spectral MAE (SPMAE)**, the first masked autoencoding framework for raw radar spectra, and **Leakage-aware Spectral MAE (L-SPMAE)**, a physically grounded extension that corrupts the visible context by simulating the cross-bin energy spreading induced by missing pre-FFT measurements. We find that leakage corruption alone achieves performance comparable to random masking, and that the two yield the strongest representations. Our results also reveal that the organization of Doppler is critical. Simply stacking Doppler bins as channels brings no benefit over mean Doppler aggregation, while treating Doppler as a primary spatial dimension in an x-y-D representation substantially improves downstream perception after self-supervised pretraining.

Figure 1 illustrates the central empirical finding: supervised learning degrades as spectral density increases, whereas self-supervised pretraining benefits from the additional spectral information. On K-Radar [14], our approach surpasses recent radar-specific detectors such as RTNH+ [10] and RADENet [6], despite using a generic 3D detector [15] rather than a radar-specific architecture. Consistent improvements are also observed on RADIal [16]. These performance improvements are not explained by model capacity or additional supervision. They come from giving the model

access to signals that handcrafted pipelines discard, together with an objective capable of discovering coherent structure within dense spectra.

Our contributions are as follows:

- We introduce a principle for learning from unprocessed radar spectra: signal is the component that remains reconstructable under localized corruption. This view motivates self-supervised masked modeling as a natural learning paradigm for dense radar spectra.
- Based on this insight, we introduce **SPMAE** for radar spectrum pretraining and further propose **L-SPMAE**, a physically grounded extension that simulates cross-bin spectral leakage induced by missing pre-FFT measurements, yielding stronger representations than random masking alone.
- An (x, y, D) Doppler encoding that treats Doppler as a primary spatial dimension rather than a channel feature, which we show is necessary for SSL pretraining to exploit the full 4D radar cube.
- State-of-the-art 3D detection on K-Radar and RADIal using a generic detector pretrained on raw spectra, surpassing radar-specific detectors and SSL baselines trained on reduced representations, without any handcrafted reduction.

2 Related Work

Radar Signal Representation. Radar systems acquire time-domain analog-to-digital converter (ADC) signals and apply DFT-based processing to obtain a 4D spectrum spanning range, azimuth, elevation, and Doppler [17], which is then followed by a signal processing reduction pipeline to obtain a sparse, clean representation, such as a point cloud. CFAR algorithms [2] serve as a standard approach for detecting peak reflections above an adaptive threshold. In addition, sparse radar point clouds in K-Radar [14] are generated by selecting the top $N\%$ reflections ranked by power. Furthermore, RTNH+ [10] introduces a combined CFAR-based two-level preprocessing method to capture different radar characteristics. Recently, learnable radar representations have been explored [18, 19], but these approaches typically rely on supervision from other modalities such as LiDAR or camera.

Radar 3D Perception. Building upon these representations, radar sensors have been increasingly adopted for 3D perception tasks [3, 9, 20], especially 3D object detection. Most existing methods extract statistical features from CFAR-filtered radar point clouds. For example, SMURF [4] uses kernel density estimation, while MUFASA [11] designs local geometric descriptors based on Lalonde features. A few methods also operate directly on radar spectra, but still rely on compressed representations. For instance, RADENet [6] projects the 4D spectrum into 2D by compressing the Doppler and elevation channels. Overall, existing radar perception methods rely heavily on handcrafted reductions and sparse supervision, limiting their ability to exploit dense radar spectra. We argue that self-supervised learning directly on radar spectra offers a more suitable alternative.

3D Self-Supervised Learning. Reconstruction-based SSL has proven effective for learning geometric representations in 3D perception. GeoMAE [21] learns structure-aware features by reconstructing geometric properties such as centroids, OccupancyMAE [22] proposes voxel occupancy as a pretraining target, and NOMAE [23] extends this by predicting multi-scale occupancy in neighborhoods of visible tokens. Self-distillation approaches have also shown promising results [24–27], although they typically rely on carefully designed augmentations or regularizations to avoid collapse, whereas reconstruction-based methods directly exploit the intrinsic structure of the input. In radar perception, existing SSL works either operate on point clouds [28, 29], depend on cross-modal supervision [30, 31], or apply MAE-style pretraining to RF signals with sparsity-aware masking, as in PRISM [32]. However, SSL tailored to the intrinsic structure of unreduced radar spectra remains largely underexplored. Our work is further inspired by Noise2Self [13], which shows that $\mathcal{J}$-invariant masking enables self-supervised prediction without clean supervision, a perspective particularly relevant to radar, where observations are inherently noisy.

3 Methods

In this section, we develop a framework for learning directly from raw radar spectra without any handcrafted reduction. Section 3.1 establishes the theoretical basis for signal discovery in dense

spectra using corruption and reconstruction. We instantiate this through two complementary corruption strategies: Spectral MAE (Section 3.2), which applies standard masked autoencoding to raw radar spectra, and Leakage-aware Spectral MAE (Section 3.3), which extends it with a physically grounded corruption strategy. Both are developed under the RAE-cube formulation, with Doppler encoded as features. Section 3.4 later shows that how Doppler enters this representation is itself a critical design choice, and introduces a RAD-cube reformulation that the framework accommodates without modification.

3.1 Signal Discovery via Corrupted-Context Prediction

The radar spectrum is a 4D tensor, but for the purpose of this treatment, we flatten the radar spectrum to a vector $X \in \mathbb{R}^m$. The partition structure we define is over indices and does not depend on any particular ordering, so the treatment applies to any indexing of the spectrum. Let $J \subset \{1, \ldots, m\}$ be a subset of indices and $J^c = \{1, \ldots, m\} \setminus J$ its complement. We define the family of $\mathcal{J}$-invariant functions as all functions f whose output on J does not depend on X_J. A network that receives only X_{J^c} as input is by definition $\mathcal{J}$-invariant.

For a fixed partition $\{J, J^c\}$, the self-supervised reconstruction loss over the $\mathcal{J}$-invariant family is

$$\mathcal{L}_J(f) = \mathbb{E}_X \left[\| f(X_{J^c})_J - X_J \|^2 \right], \tag{1}$$

whose unique minimizer is $f^*_J(X_{J^c})_J = \mathbb{E}[X_J \mid X_{J^c}]$. When J and J^c are statistically coherent, X_{J^c} is informative about X_J and $\mathcal{L}_J$ will be low at the optimum. When they are independent, the optimum reduces to the marginal mean $\mathbb{E}[X_J]$ and $\mathcal{L}_J$ will be high. The per-partition loss is therefore a measure of statistical coherence between J and J^c.

The global loss takes an expectation over both X and randomly drawn partitions J:

$$\mathcal{L}(f) = \mathbb{E}_{X,J} \left[\| f(X_{J^c})_J - X_J \|^2 \right]. \tag{2}$$

Since J is sampled randomly, the model cannot rely on any fixed set of coherent indices always being visible. Across many training steps, coherences at all strengths and scales are alternately available in context and masked as targets, forcing the model to discover and represent the full coherence structure of the spectrum.

We define *signal* as the coherence structure of X, operationalized as $f^*(X)$: the output of the model trained under the global loss when presented with the full unmasked spectrum at inference time. Indices statistically independent of all others do not reduce $\mathcal{L}_J$ under any partition and are not represented in f^*. Indices that participate in coherent structure are represented in proportion to that coherence.

Radar physics provides intuition for this notion of coherence. Target extent, the sensor's point spread function, and spectral leakage spread reflected energy across neighboring bins, making target responses locally predictable, whereas thermal noise is less predictable from context. This predictability can also extend to multipath and stable clutter, which may carry useful information about objects and scene geometry. Learning these coherent components preserves that information for downstream fine-tuning to exploit. We stress that this interpretation is a motivating argument rather than a formal proof. We validate it empirically in Section 5.

This framework is related to the $\mathcal{J}$-invariant framework of Noise2Self [13], from which we borrow the loss formulation. The problems are, however, fundamentally different: Noise2Self assumes a clean signal exists and uses $\mathcal{J}$-invariance to recover it under a noise independence assumption, whereas we make no such assumption and define signal as the coherence structure of the spectrum itself. The corruption strategy need not be limited to binary masking: any transformation that renders X_J uninformative about itself defines the same objective, and physically motivated corruptions that distort rather than remove can define harder and more structured prediction tasks.

3.2 Spectral MAE

Following the corrupted-context formulation in Section 3.1, we instantiate a self-supervised learning method termed Spectral MAE (SPMAE), which applies masked autoencoding directly to raw radar spectra. Given a 4D radar spectrum $\mathbb{R}^{R \times A \times E \times D}$ indexed by range, azimuth, elevation, and Doppler, we convert the center of each (r, a, e) bin from range–azimuth–elevation coordinates to Cartesian

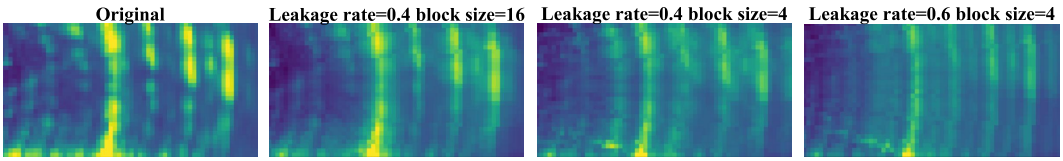


Figure 2: BEV visualization of the leakage effect. A higher leakage rate l_{rate} leads to stronger energy spreading. Larger block sizes b produce more structured and spatially correlated low-frequency distortions, while smaller blocks generate less correlated and more diffuse high-frequency perturbations.

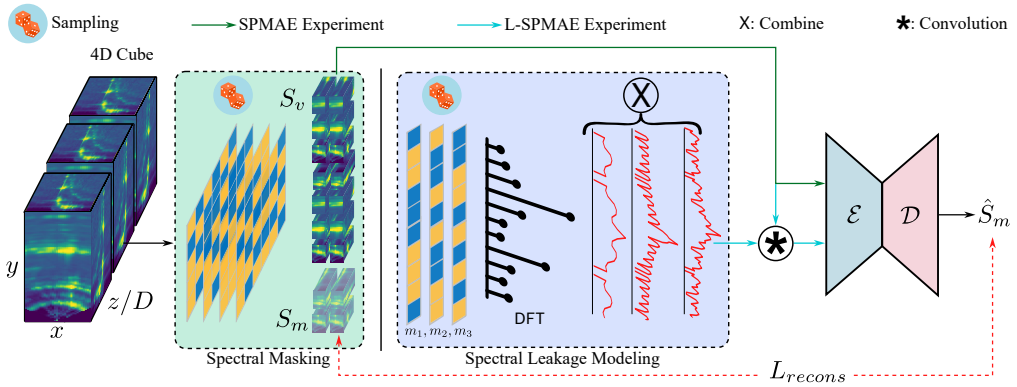


Figure 3: Overall pipeline. For **L-SPMAE**, random spatial masking is applied to the radar spectrum S, producing the visible spectrum S_v and masked spectrum S_m. Spectral leakage modeling is then applied to S_v (Section 3.3) to generate the corrupted $\tilde{S}_v$, which is fed into the encoder $\mathcal{E}$, followed by the decoder $\mathcal{D}$ with mask tokens for reconstruction. For downstream tasks, only $\mathcal{E}$ is retained as the feature extractor.

coordinates using the corresponding bin resolutions. This yields $P = R \times A \times E$ spectrum points with

$$\mathbf{s}_i = (\mathbf{x}_i, \mathbf{f}_i), \mathbf{x}_i = (x_i, y_i, z_i), \quad S = \{\mathbf{s}_i\}_{i=1}^{P}. \tag{3}$$

The feature $\mathbf{f}_i$ is obtained by aggregating along the Doppler dimension. In the absence of ego-motion information, we follow K-Radar [14] and use the mean Doppler power as the point feature, which we refer to as RAE.

During pretraining, we randomly mask a subset of spectrum points S according to a mask ratio m_{rate}. Let S_v and S_m denote the visible and masked spectrum points, respectively. The visible points S_v are fed into the encoder $\mathcal{E}$ to obtain latent representations. For the masked positions indexed by S_m, we introduce a set of mask tokens $\{t_i\}_{i \in S_m}$ and pass them to the decoder $\mathcal{D}$ together with the encoded visible tokens. The training objective is to recover the original feature vectors $\mathbf{f}_i$ of the masked spectrum points:

$$\mathcal{L}_{\text{SPMAE}} = \frac{1}{|S_m|} \sum_{i \in S_m} \left\| \mathcal{D}(\mathcal{E}(S_v), t_i) - \mathbf{f}_i \right\|_1, \tag{4}$$

where we use the ℓ_1 loss and average it over masked locations.

3.3 Leakage-aware Spectral MAE

Although SPMAE provides a simple masked reconstruction baseline, random spatial masking ignores a key property of radar spectra: spectral bins are coupled through the DFT. In digital signal processing, finite-length sampling is equivalent to applying a window in the signal domain, whose frequency response induces spectral leakage [33]. For a signal z and a binary sampling window m, this can be written as

$$\text{DFT}(z \odot m) = \text{DFT}(z) \circledast \text{DFT}(m), \tag{5}$$

where $\odot$ denotes element-wise multiplication and $\circledast$ denotes circular convolution. Thus, windowed measurements do not merely remove isolated spectral bins but also cause cross-bin energy spreading in the spectral domain.

Motivated by this observation, we propose Leakage-aware Spectral MAE (L-SPMAE), a radar-specific extension of SPMAE that directly models leakage effects on radar spectra. Rather than corrupting spectrum points independently, we generate random binary windows and transform their DFT responses into controllable convolution kernels that simulate cross-bin energy spreading in the spectral domain. The resulting corrupted views better capture radar-specific spectral distortions and provide a physically inspired pretext task distinct from standard point masking. The randomized corruption is designed to make reconstruction challenging, rather than to match the statistics of actual hardware leakage.

To construct a leakage-aware corrupted view, we generate axis-wise binary windows

$$m_1 \in \{0,1\}^R,\ m_2 \in \{0,1\}^A,\ m_3 \in \{0,1\}^E, \tag{6}$$

where each window keeps a fraction $(1 - l_{\text{rate}})$ of samples along the corresponding axis. The leakage rate $l_{\text{rate}} \in [0,1]$ controls the strength of the induced corruption: a larger l_{rate} preserves fewer effective samples and therefore produces stronger energy spreading. To generate more structured leakage patterns, we sample the binary windows in contiguous blocks with a block size b. Larger blocks introduce more structured and spatially correlated perturbations, resulting in more localized and low-frequency distortions in the spectrum domain. In contrast, smaller blocks produce less correlated, high-frequency perturbations, which lead to more diffuse and globally distributed cross-bin energy spreading. We then compute a separable leakage kernel

$$K = \left|\mathrm{DFT}(m_1)\right|^2 \otimes \left|\mathrm{DFT}(m_2)\right|^2 \otimes \left|\mathrm{DFT}(m_3)\right|^2, \tag{7}$$

where $\otimes$ denotes the outer product. The squared magnitude is used to model energy spreading in the spectrum domain. The corrupted spectrum is obtained by circular convolution $\circledast$ along the three spatial axes independently for each channel. An example of the corrupted spectrum is visualized in Figure 2.

As shown in Figure 3, L-SPMAE combines leakage-aware corruption with masked reconstruction in a decoupled manner. Random masking is first applied to S to obtain the visible spectrum S_v, after which the leakage kernel K is used to construct the corrupted spectrum:

$$\tilde{S}_v = S_v \circledast K. \tag{8}$$

The reconstruction target remains the original $\mathbf{f}_i$ from the uncorrupted masked spectrum S_m. The only difference from Equation 4 is that $\tilde{S}_v$ is used as input instead of the original S_v. This design makes the reconstruction task more radar-consistent than SPMAE, since the model must infer missing targets not only from incomplete observations, but also from contexts distorted by cross-bin leakage. Further details of the leakage modeling are provided in Appendix E.

3.4 Doppler-aware Spectrum Encoding

The framework developed in Sections 3.2 and 3.3 uses the RAE-cube formulation, in which Doppler is aggregated into point features via mean Doppler power. A natural question is whether SSL can extract additional value from the full 4D spectrum by retaining all Doppler bins. The obvious approach is channel stacking (which we refer to as RAE-d), but we find it brings no benefit over mean Doppler aggregation: without a spatial axis for Doppler, the encoder cannot model Doppler structure, and SSL pretraining cannot leverage Doppler coherence as a predictive signal.

To address this, we introduce a Doppler-aware reformulation that treats Doppler as a primary spatial dimension. We swap the elevation and Doppler axes and construct spectrum points from (r, a, d) bins, yielding $P = R \times A \times D$ points with

$$\mathbf{x}_i = (x_i, y_i, d_i), \tag{9}$$

where (x_i, y_i) are obtained from range and azimuth bins and d_i denotes the Doppler coordinate (denoted as RAD). The associated feature vector is taken from the elevation dimension:

$$\mathbf{f}_i \in \mathbb{R}^E. \tag{10}$$

Height is encoded as channels while Doppler becomes an explicit axis for attention, allowing the encoder to directly model dependencies along the Doppler dimension. SPMAE and L-SPMAE apply without modification: $\mathbf{x}_i$ changes from (x, y, z) to (x, y, d), $\mathbf{f}_i$ changes from mean Doppler power to elevation bins, and for L-SPMAE the dense grid becomes $\mathbb{R}^{R \times A \times D \times C}$ with $C = E$.

4 Experiments

4.1 Dataset

K-Radar [14] is the largest open-source 4D radar dataset providing raw 4D radar spectrum data. The dataset contains over 35k frames with object detection annotations. As noted in previous works [34–36], the official split divides individual sequences into subsequences that appear in both the training and testing sets. Therefore, in addition to the official split, we construct an alternative sequence-level split. Details about the split construction and the results are in Appendix A.1 and B.1. We further evaluate our method on the occupancy prediction task following the experimental setting of RadarOcc [5].

RADIal [16] contains 91 driving sequences (about 2 hours in total) with around 8k annotated frames, with radar signals released in raw ADC form. Since no precomputed 4D radar cubes are provided, we reconstruct the 4D spectrum from the raw ADC data using standard signal processing.

4.2 Evaluation Protocol

We adopt 3D object detection as the main downstream task, as it is the most widely studied setting in radar perception. Following the official K-Radar protocol [14], we report $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$ with an IoU threshold of 0.3. For RADIal, we report mAP with an IoU threshold of 0.5 due to known issues with the default evaluation metric[2]. For occupancy prediction, we evaluate foreground-background segmentation and report the mean Intersection over Union (mIoU), averaged over the two classes, at three evaluation ranges: 12.8 m, 25.6 m, and 51.2 m.

To evaluate self-supervised learning methods, we consider two evaluation protocols. For fine-tuning, the encoder backbone is unfrozen and jointly optimized with the detection head to evaluate the transferability of the pretrained representations. The main results are discussed under the fine-tuning setting. Additionally, we conduct decoder probing, where the encoder is frozen while a lightweight two-layer decoder and the detection head are trained on top of it. Following recent work [23, 24], this protocol provides a more direct evaluation of the quality of the learned representation. We use it for the hyperparameter analysis (Figure 4) and the additional comparisons in Appendix B.

4.3 Implementation Details

Following prior work [23–25], we adopt the state-of-the-art 3D detector PTv3 [15] for all experiments. The voxel size is set to $[0.4, 0.4, 0.4]$ m. The detection range of $[x, y, z]$ is $[0, 72]\times[-6.4, 6.4]\times[-2, 6]$ m for K-Radar and $[0, 103] \times [-40, 40] \times [-2.5, 2.0]$ m for RADIal. The encoder and decoder depths are set to $[2, 6]$ and $[4]$, respectively. During pretraining, models are trained for 60k iterations with a learning rate of 1×10^{-3}. For fine-tuning, we follow [23] and apply layer-wise learning rate decay (LLRD) [37], using a base learning rate of 1×10^{-4} for 20 epochs with a batch size of 8. For SPMAE, the spatial masking ratio is $m_{\mathrm{rate}} = 0.7$. For L-SPMAE, $m_{\mathrm{rate}} = 0.4$, and spectral leakage is applied with $l_{\mathrm{rate}} = 0.2$ along the 3D axes using a block size of $b = 4$. We use CenterHead [38] with a score threshold of 0.1. Additional details are provided in Appendix A.2.

5 Results and Analysis

5.1 From Handcrafted Reductions to Signal Discovery

Table 1 reports results on the K-Radar benchmark. Under fully supervised training, richer representations counterintuitively perform worse: compared with point clouds, RAE and RAE-d decrease performance by 1.6/1.5 and 3.2/2.8 mAP, respectively. This is consistent with the argument in Section 3.1: the supervised objective alone provides insufficient guidance to interpret dense spectral structures, and the coherence structure latent in the spectrum remains undiscovered.

Signal discovery reverses this trend. After fine-tuning, SPMAE and L-SPMAE consistently outperform all point-cloud-based counterparts across all spectral representations, with L-SPMAE on RAE reaching $53.7/48.4$ mAP ($+2.9/+3.0$ over the best baseline). This confirms that the pretraining

[2] https://github.com/valeoai/RADIal/issues/46

Table 1: Comparison of $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$ on the K-Radar official split. Sup. denotes supervised learning. RAE is defined in Section 3.2, and RAE-d and RAD in Section 3.4. The best results are highlighted in **bold**.

	Methods	Representation	Sedan	Bus/Truck	Pedestrian	mAP
Sup.	PTv3 [15]	Point Cloud	56.5/50.9	51.3/39.2	39.1/36.9	49.0/42.3
		RAE	56.5/49.8	50.1/39.2	35.5/33.3	47.4/40.8
		RAE-d	53.4/47.3	46.2/36.8	37.7/34.5	45.8/39.5
Fine-tune	+OccupancyMAE [22]	Point Cloud	58.6/53.0	54.0/45.5	38.5/37.6	50.4/45.4
	+SONATA [24]	Point Cloud	58.2/52.6	51.4/41.0	42.7/41.6	50.8/45.0
	+NOMAE [23]	Point Cloud	59.3/53.7	53.8/43.5	39.2/37.3	50.8/44.9
	+SPMAE (Ours)	RAE	61.0/55.0	55.7/48.1	42.5/41.2	53.1/48.1
	+L-SPMAE (Ours)	RAE	61.3/55.3	56.9/47.8	42.9/42.0	53.7/48.4
	+SPMAE (Ours)	RAE-d	59.3/53.4	54.2/44.8	42.9/40.8	52.1/46.3
	+L-SPMAE (Ours)	RAE-d	59.3/53.4	54.9/45.7	42.4/41.4	52.2/46.8
Sup.	PTv3 [15]	RAD	58.1/51.6	51.4/43.5	37.1/34.3	48.9/43.1
Fine-tune	**+SPMAE (Ours)**	RAD	61.2/55.0	58.2/47.2	43.5/41.5	54.3/47.9
	+L-SPMAE (Ours)	RAD	**62.7/55.3**	**59.5/51.7**	**45.6/43.4**	**55.9/50.1**

Table 2: Comparison of $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$ on the K-Radar benchmark: single-class "Sedan", label version v1.0, score_threshold $= 0.3$. Best results in **bold**.

Methods	Representation	Sedan
RTNH [14]	Point Cloud	58.4/47.4
DPFT [34]	Image+Reduced Spectrum	-/56.1
RTNH+ [10]	Reduced Spectrum	65.7/57.6
PTv3 [15]	Spectrum	61.9/58.4
+L-SPMAE(Ours)	Spectrum	**67.7/64.0**

Table 3: Comparison of $\mathrm{AP}_{\mathrm{BEV}}$ on the RADIal test set. We follow the same evaluation protocol as K-Radar, reporting mAP based on the PR curve with IoU=0.5.

Methods	Representation	mAP
FFTRadNet [16]	RD map	89.3
PTv3 [15]	Spectrum	89.0
+SPMAE (Ours)	Spectrum	89.6
+L-SPMAE (Ours)	Spectrum	**90.0**

objective forces the encoder to discover and represent coherent structures in the spectrum that handcrafted reductions discard.

Comparing L-SPMAE and SPMAE, spectral leakage modeling consistently improves performance across all representations ($+0.6/+0.3$ mAP on RAE), demonstrating that physics-inspired corruption defines a harder and more structured prediction task, yielding more informative representations.

Despite containing more information than RAE, RAE-d underperforms after fine-tuning, suggesting that stacking Doppler as channels prevents the model from exploiting Doppler coherence effectively. The RAD representation addresses this by exposing Doppler as a dedicated axis, preserving the same information as RAE-d but in a layout that makes Doppler structure accessible to the model. L-SPMAE on RAD achieves $\mathbf{55.9/50.1}$ mAP, surpassing all point-cloud baselines by +5.1/+4.7 points in $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$ and recovering the expected ordering where richer representations yield better performance after signal discovery. With the PTv3 backbone and RAD representation fixed, SPMAE improves over supervised training by +5.4/+4.8 $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$, while L-SPMAE provides a further gain of +1.6/+2.2 over SPMAE. These comparisons distinguish the benefit of masked pretraining from the additional contribution of leakage corruption.

5.2 Comparison with State-of-the-Art Radar Detectors

Existing radar detectors typically rely on handcrafted reduced representations with radar-specific architectural designs, as discussed in Section 2. We investigate whether signal discovery on raw radar spectra using a generic 3D backbone can learn more representative features. Following the official K-Radar benchmarking protocol, Table 2 shows that without pretraining, PTv3 already achieves higher $\mathrm{AP}_{\mathrm{3D}}$ than RTNH+ [10], and with fine-tuning further obtains gains of $+2.0/+6.4$ in $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$ over RTNH+.

Table 5 reports results under the 4-class setting of RADENet [6]. PTv3 with RAD already surpasses RADENet across all classes without pretraining, with particularly strong gains on Pedestrian and Bicycle. Fine-tuning further amplifies these gains, achieving $+6.6/+7.0$ on the Sedan class, and L-SPMAE consistently outperforms SPMAE by $+1.5/+1.4$. Performance degradation on Bicycle after fine-tuning is attributed to the extremely limited number of training samples for that class.

Table 4: Occupancy prediction results on K-Radar following RadarOcc [5]. Reproduced RadarOcc results slightly outperform the reported values.

Methods	mIoU (%)			BG IoU (%)			FG IoU (%)		
	12.8m	25.6m	51.2m	12.8m	25.6m	51.2m	12.8m	25.6m	51.2m
RadarOcc [5]	42.1	33.7	25.7	**49.7**	**41.0**	**30.6**	34.5	26.4	20.8
PTv3 [15]	45.4	34.2	26.8	48.2	36.6	28.4	42.5	31.8	25.2
+SPMAE (Ours)	47.2	35.1	27.4	49.0	37.6	28.8	45.3	32.6	25.9
+L-SPMAE (Ours)	**47.5**	**35.5**	**27.8**	49.3	37.7	29.1	**45.8**	**33.3**	**26.5**

Table 5: Comparison of $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$ on the K-Radar official split with RADENet 4-class setting, label version v2.1.

Methods	Representation	Sedan	Bus/Truck	Pedestrian	Bicycle	mAP
RADENet [6]	Reduced Spectrum	63.2/56.8	48.3/41.0	6.8/5.4	1.5/1.5	30.0/26.2
PTv3 [15]	Spectrum	64.7/58.8	61.6/50.8	35.6/34.4	**49.0/42.2**	52.7/46.5
+SPMAE (Ours)	Spectrum	67.8/61.8	**65.1/53.2**	48.7/47.0	29.7/29.7	52.8/47.9
+L-SPMAE (Ours)	Spectrum	**69.8/63.8**	63.9/52.1	**49.9/47.6**	33.6/33.6	**54.3/49.3**

5.3 Generalization to Other Datasets and Tasks

Table 3 reports results on RADIal, where spectrum inputs are constructed from raw radar signals. L-SPMAE achieves the best performance ($+0.7$ over FFTRadNet), though gains are smaller than on K-Radar, which we attribute to the lower scene diversity and complexity of RADIal.

We further evaluate our methods on radar-based occupancy prediction, with the results summarized in Table 4. Both SPMAE and L-SPMAE consistently improve over the supervised PTv3 baseline across all evaluation ranges. Compared with RadarOcc, our models achieve substantially higher foreground IoU, while RadarOcc performs better on background IoU, indicating that our pretrained representations are particularly effective at capturing foreground objects. The gain of L-SPMAE over SPMAE is relatively modest, likely because the binary foreground-background formulation provides only coarse supervision. We additionally evaluate the learned representations on radar-based multi-object tracking, with results provided in Appendix B.3.

5.4 Ablation Study

Ablation studies are conducted on the K-Radar official split for three classes with label version v2.0. Additional results and visualizations are provided in Appendix B.

Variants of L-SPMAE. Table 6 compares different masking and corruption strategies. The FOCUS masking [32] improves over the supervised baseline, but remains below global random masking (SPMAE). This highlights a key distinction: FOCUS relies on predefined power-based selection of informative regions, whereas SPMAE/L-SPMAE operate on the complete unreduced spectrum without such preselection. Adding leakage corruption to FOCUS yields only a modest gain, while applying leakage globally performs substantially better. Combining global random masking with leakage modeling achieves the best performance, suggesting that full-spectrum learning and leakage-aware corruption provide complementary benefits.

Backbone Choices. To verify that our method is not backbone-specific, we additionally report results with a Sparse Encoder [38] in Table 8. SPMAE clearly improves over the supervised baseline, and L-SPMAE brings further gains, showing the same trend as with PTv3. These results confirm that our leakage-inspired pretraining generalizes across backbone architectures.

Comparison with LiDAR-Guided Pretraining. Recent methods [39, 40] use LiDAR as auxiliary supervision to provide cleaner geometry for radar perception. We compare our radar-only pretraining with two LiDAR-guided approaches in Table 7. L-SPMAE clearly outperforms LiDAR feature distillation and LiDAR occupancy pretraining. This suggests that constraining radar representations toward clean LiDAR geometry may discard useful radar-specific context, while L-SPMAE better preserves and exploits such task-relevant structures.

Empirical Validation of Spectral Reconstruction. To empirically validate the analysis in Section 3.1, we reconstruct each radar spectrum in a single unmasked forward pass through a frozen

Table 6: Comparison of variants. FOCUS Masking is reimplemented following the original paper.

Method	AP_{BEV}	AP_{3D}
From Scratch	48.9	43.1
FOCUS Mask [32]	52.1	45.4
FOCUS Mask + Leakage	52.4	45.7
Leakage Only	53.9	47.8
Random Mask	54.3	47.9
Leakage + Random Mask	**55.9**	**50.1**

Table 7: Comparison of radar-only L-SPMAE with LiDAR-guided pretraining approaches using the same PTv3 backbone on the K-Radar official split.

Methods	mAP
PTv3 [15]	48.9 / 43.1
+ LiDAR Occ Pretrain [39]	52.2 / 45.9
+ LiDAR Feat Distill [40]	53.1 / 46.8
+ L-SPMAE (Ours)	**55.9 / 50.1**

Table 8: Results with the Sparse Encoder backbone.

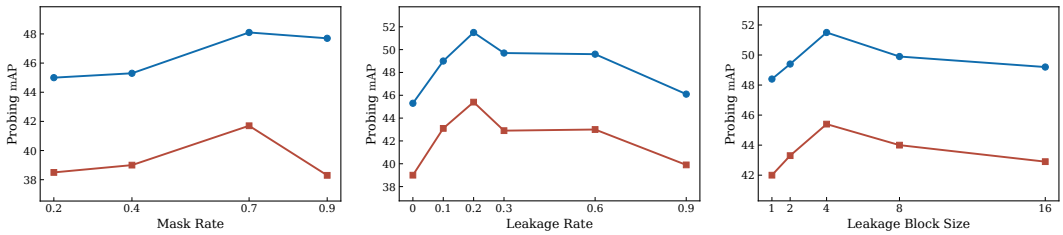
Backbone	AP_{BEV}	AP_{3D}
Sparse Encoder [38]	47.2	40.5
+ SPMAE (Ours)	52.4	44.9
+ L-SPMAE (Ours)	**53.1**	**47.0**

Table 9: Effect of L-SPMAE reconstruction. A PTv3 is trained from scratch using the reconstructed spectra.

Representation	AP_{BEV}	AP_{3D}
PTv3+Original Spectra	48.9	43.1
PTv3+Reconstructed Spectra	**52.0**	**45.7**

L-SPMAE, then train a new PTv3 from scratch using only the reconstructed spectra. As shown in Table 9, the reconstructed spectra improve AP_{BEV} from 48.9 to 52.0 and AP_{3D} from 43.1 to 45.7. Since the downstream network is trained from scratch, this gain is independent of weight initialization, suggesting that L-SPMAE suppresses task-irrelevant variations while preserving coherent structures useful for perception.

Hyperparameter Analysis. Figure 4 studies the impact of the masking rate m_{rate}, leakage rate l_{rate}, and leakage block size b. The masking rate controls the difficulty of the pretext task, where $m_{\text{rate}} = 0.7$ performs best. The leakage rate determines the amount of energy spread: values in 0.1–0.6 are robust with $l_{\text{rate}} = 0.2$ performing best, while overly aggressive leakage (e.g., 0.9) degrades performance. The block size b controls the locality of leakage, with $b = 4$ yielding the best overall performance.



(a) (b) (c)

Figure 4: Probing results of hyperparameter settings: (a) mask rate m_{rate} (SPMAE, no leakage), (b) leakage rate l_{rate}, and (c) leakage block size b. (AP_{BEV}/AP_{3D}).

6 Conclusion

We have argued and empirically demonstrated that handcrafted reductions of the radar spectrum such as point clouds discard coherent structure that supervised training cannot recover, but that corrupted-context prediction can discover. Under fully supervised training, richer spectral representations perform worse than point clouds; after signal discovery pretraining, this ordering reverses, with L-SPMAE on the RAD representation surpassing all point-cloud-based counterparts by a substantial margin. This reversal is the central finding of our work: the information is present in the spectrum, but requires the right pretraining objective to be unlocked. We further show that physics-inspired corruption via spectral leakage modeling defines a harder and more structured prediction task than random masking alone, and that the choice of representation layout, not merely the quantity of retained information, determines how effectively the coherence structure of the spectrum can be exploited.

Limitations and Future Work. Processing dense radar spectra incurs substantially higher computational cost than point-cloud-based perception. Future work will explore token pruning [41] to reduce this overhead, as well as alternative SSL methods beyond MAE for spectrum representations.

Acknowledgements and Disclosures

This work was supported in part by the IPCEI ME/CT project. The IPCEI ME/CT project is supported by the Federal Ministry for Economic Affairs and Climate Action on the basis of a decision by the German Parliament, by the Ministry for Economic Affairs, Labor and Tourism of Baden-Württemberg based on a decision of the State Parliament of Baden-Württemberg, the Free State of Saxony on the basis of the budget adopted by the Saxon State Parliament, the Bavarian State Ministry for Economic Affairs, Regional Development and Energy and financed by the European Union – NextGenerationEU.

References

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

[2] Hermann Rohling. Radar cfar thresholding in clutter and multiple target situations. *IEEE Transactions on Aerospace and Electronic Systems*, AES-19(4):608–621, 1983.

[3] Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian FP Kooij, and Dariu M Gavrila. Multi-class road user detection with 3+ 1d radar in the view-of-delft dataset. *IEEE Robotics and Automation Letters*, 7(2):4961–4968, 2022.

[4] Jianan Liu, Qiuchi Zhao, Weiyi Xiong, Tao Huang, Qing-Long Han, and Bing Zhu. Smurf: Spatial multi-representation fusion for 3d object detection with 4d imaging radar. *IEEE Transactions on Intelligent Vehicles*, 9(1):799–812, 2023.

[5] Fangqiang Ding, Xiangyu Wen, Yunzhou Zhu, Yiming Li, and Chris Xiaoxuan Lu. Radarocc: Robust 3d occupancy prediction with 4d imaging radar. *Advances in Neural Information Processing Systems*, 37:101589–101617, 2024.

[6] Christof Leitgeb, Thomas Puchleitner, Max Peter Ronecker, and Daniel Watzenig. Rade-net: Robust attention network for radar-only object detection in adverse weather. *arXiv preprint arXiv:2602.19994*, 2026.

[7] Santosh Parit, Julia Hindel, Rohit Mohan, Georgi Pramatarov, Paul Newman, Daniele De Martini, and Abhinav Valada. Real-time radar semantic and panoptic scene understanding. *arXiv preprint arXiv:2610.26816*, 2026.

[8] Abhijeet Nayak, Daniele Cattaneo, and Abhinav Valada. Ralf: Flow-based global and metric radar localization in lidar maps. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5097–5103. IEEE, 2024.

[9] Jen-Hao Cheng, Sheng-Yao Kuan, Hou-I Liu, Hugo Latapie, Gaowen Liu, and Jenq-Neng Hwang. Centerradarnet: Joint 3d object detection and tracking framework using 4d fmcw radar. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 998–1004. IEEE, 2024.

[10] Seung-Hyun Kong, Dong-Hee Paek, and Sangyeong Lee. Rtnh+: Enhanced 4d radar object detection network using two-level preprocessing and vertical encoding. *IEEE Transactions on Intelligent Vehicles*, 10:1427–1440, 2025.

[11] Xiangyuan Peng, Miao Tang, Huawei Sun, Kay Bierzynski, Lorenzo Servadei, and Robert Wille. Mufasa: Multi-view fusion and adaptation network with spatial awareness for radar object detection. In *International Conference on Artificial Neural Networks*, pages 168–184. Springer, 2024.

[12] Dong-Hee Paek, Seung-Hyun Kong, and Kevin Tirta Wijaya. Enhanced k-radar: Optimal density reduction to improve detection performance and accessibility of 4d radar tensor-based object detection. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2023.

[13] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference On Machine Learning*, pages 524–533. PMLR, 2019.

[14] Dong-Hee Paek, Seung-Hyun Kong, and Kevin Tirta Wijaya. K-radar: 4d radar object detection for autonomous driving in various weather conditions. *Advances in Neural Information Processing Systems*, 35:3819–3829, 2022.

[15] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024.

[16] 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, June 2022.

[17] Zeyu Han, Jiahao Wang, Zikun Xu, Shuocheng Yang, Lei He, Shaobing Xu, Jianqiang Wang, and Keqiang Li. 4d millimeter-wave radar in autonomous driving: A survey. *arXiv preprint arXiv:2306.04242*, 2023.

[18] Shengpeng Wang, Xin Luo, Yulong Xie, and Wei Wang. Sddiff: Boosting radar perception via spatial-doppler diffusion. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 8805–8813, 2025.

[19] Ignacio Roldan, Andras Palffy, Julian FP Kooij, Dariu M Gavrila, Francesco Fioranelli, and Alexander Yarovoy. See further than cfar: a data-driven radar detector trained by lidar. In *2024 IEEE Radar Conference (RadarConf24)*, pages 1–6. IEEE, 2024.

[20] Matthias Zeller, Vardeep S Sandhu, Benedikt Mersch, Jens Behley, Michael Heidingsfeld, and Cyrill Stachniss. Radar instance transformer: Reliable moving instance segmentation in sparse radar point clouds. *IEEE Transactions on Robotics*, 40:2357–2372, 2023.

[21] Xiaoyu Tian, Haoxi Ran, Yue Wang, and Hang Zhao. Geomae: Masked geometric target prediction for self-supervised point cloud pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13570–13580, 2023.

[22] Chen Min, Liang Xiao, Dawei Zhao, Yiming Nie, and Bin Dai. Occupancy-mae: Self-supervised pre-training large-scale lidar point clouds with masked occupancy autoencoders. *IEEE Transactions on Intelligent Vehicles*, 9(7):5150–5162, 2023.

[23] Mohamed Abdelsamad, Michael Ulrich, Claudius Gläser, and Abhinav Valada. Multi-scale neighborhood occupancy masked autoencoder for self-supervised learning in lidar point clouds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22234–22243, 2025.

[24] Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22193–22204, 2025.

[25] Mohamed Abdelsamad, Michael Ulrich, Bin Yang, Miao Zhang, Yakov Miron, and Abhinav Valada. Dos: Distilling observable softmaps of zipfian prototypes for self-supervised point representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 19514–19523, 2026.

[26] Bin Yang, Mohamed Abdelsamad, Miao Zhang, and Alexandru Paul Condurache. Towards foundation models for 3d scene understanding: Instance-aware self-supervised learning for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2938–2947, June 2026.

[27] Mohamed Abdelsamad, Bin Yang, Michael Ulrich, Miao Zhang, Yakov Miron, Alexandru Paul Condurache, and Abhinav Valada. Ghostpoint: Self-supervised representation learning by hallucinating occluded lidar structure. In *European Conference on Computer Vision*, pages 116–137. Springer, 2026.

[28] Fangqiang Ding, Zhijun Pan, Yimin Deng, Jianning Deng, and Chris Xiaoxuan Lu. Self-supervised scene flow estimation with 4-d automotive radar. *IEEE Robotics and Automation Letters*, 7(3):8233–8240, 2022.

[29] Zhiheng Li, Weihua Wang, Qiang Shen, Yichen Zhao, and Zheng Fang. Super4dr: 4d radar-centric self-supervised odometry and gaussian-based map optimization. *arXiv preprint arXiv:2512.09608*, 2025.

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

[31] Yuan Luo, Rudolf Hoffmann, Yan Xia, Olaf Wysocki, Benedikt Schwab, Thomas H Kolbe, and Daniel Cremers. Radler: Radar object detection leveraging semantic 3d city models and self-supervised radar-image learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4452–4461, 2025.

[32] Liang Fang, Ruiyuan Song, Zhi Lu, Dongheng Zhang, Yang Hu, Qibin Sun, and Yan Chen. Prism: Pre-training rf signals in sparsity-aware masked autoencoders. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, pages 2109–2118. IEEE, 2024.

[33] Douglas A Lyon. The discrete fourier transform, part 4: spectral leakage. *Journal of object technology*, 8(7), 2009.

[34] Felix Fent, Andras Palffy, and Holger Caesar. Dpft: Dual perspective fusion transformer for camera-radar-based object detection. *IEEE Transactions on Intelligent Vehicles*, 2024.

[35] Miao Zhang, Sherif Abdulatif, Benedikt Loesch, Marco Altmann, Marius Schwarz, and Bin Yang. Exploring domain shift on radar-based 3d object detection amidst diverse environmental conditions. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3513–3518. IEEE, 2024.

[36] Miao Zhang, Sherif Abdulatif, Benedikt Loesch, Marco Altmann, and Bin Yang. Class-aware pillarmix: Can mixed sample data augmentation enhance 3d object detection with radar point clouds? *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13349–13356, 2025.

[37] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[38] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[39] Tianshu Huang, Akarsh Prabhakara, Chuhan Chen, Jay Karhade, Deva Ramanan, Matthew O'toole, and Anthony Rowe. Towards foundational models for single-chip radar. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 24655–24665. IEEE, 2025.

[40] Geonho Bang, Kwangjin Choi, Jisong Kim, Dongsuk Kum, and Jun Won Choi. Radardistill: Boosting radar-based object detection performance via knowledge distillation from lidar features. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15491–15500. IEEE, 2024.

[41] Tuan Anh Tran, Duy Minh Ho Nguyen, Hoai-Chau Tran, Michael Barz, Khoa D. Doan, Roger Wattenhofer, Vien Anh Ngo, Mathias Niepert, Daniel Sonntag, and Paul Swoboda. How many tokens do 3d point cloud transformer architectures really need? In *Advances in Neural Information Processing Systems*, 2025.

[42] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. In *European conference on computer vision*, pages 680–696. Springer, 2022.

[43] Bin Yang and Alexandru Paul Condurache. Collaborative learning for semi-supervised lidar semantic segmentation. In *International Conference on Machine Learning*, 2026.

[44] Pointcept Contributors. Pointcept: A codebase for point cloud perception research. `https://github.com/Pointcept/Pointcept`, 2023.

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

[46] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. `https://github.com/open-mmlab/OpenPCDet`, 2020.

Appendix

A Experimental Settings

A.1 Sequence Split Details of K-Radar

To enforce a stricter separation between training and testing data, we propose a new sequence split, which is more challenging due to the limited number of sequences and the resulting domain shift. Table 10 summarizes the proposed sequence-level split. We partition the dataset by sequence ID with a 75/25 train/test ratio, ensuring that each weather condition is represented in both splits. The reported numbers of frames and the annotation distribution are computed after applying our predefined ROI filtering.

Table 10: Split statistics of the sequence-level split.

Split Info	**Train**	**Test**
Sequence IDs	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 19, 21, 22, 24, 25, 27, 28, 30, 31, 32, 33, 34, 35, 37, 38, 39, 40, 42, 43, 45, 46, 47, 48, 50, 51, 52, 54, 55, 56, 57	4, 13, 15, 17, 20, 23, 26, 29, 36, 41, 44, 49, 53, 58
Filtered frames	19,843	5,903
Filtered class distribution	Sedan: 45,251 Bus/Truck: 8,481 Pedestrian: 3,513	Sedan: 10,409 Bus/Truck: 2,483 Pedestrian: 1,408

A.2 Hyperparameter Settings

Table 11 summarizes the PTv3 [15] configuration in our work. Compared to the original setting for LiDAR and RGB-D camera, we adopt fewer stages and a shallower network.

We give the hyperparameter setting of point-cloud-based SSL in Table 12. For OccupancyMAE [22], we keep the range-wise masking but reduce the mask rate m_{rate} by 0.2, since the default setting is too aggressive for radar's lower-resolution signals. For NOMAE [23], all the default settings are maintained. For SONATA [24], we limit rotation augmentation to a small range due to our front-view setup. Accordingly, BEV flipping is applied only horizontally.

B Additional Experiments

B.1 Sequence Split Results

Beyond the official split, we evaluate on a more challenging sequence split, as shown in Table 13. Overall performance decreases due to stricter sequence-level separation, which requires stronger generalization. Both point clouds and 3D cubes struggle on the Pedestrian class, with or without SSL.

Table 11: PTv3 model setting.

Config	Value
serialization pattern	Z+TZ+H+TH
patch interaction	Shift Order+Shuffle Order
encoder depth	[2,6]
encoder channels	[64,128]
encoder num heads	[4,8]
encoder patch size	[128,128]
decoder depth	[4]
decoder channels	[64]
decoder num heads	[8]
decoder patch size	[128]
mlp ratio	[4]

Table 12: Settings of point-cloud-based SSL.

Methods	Settings
OccupancyMAE [22]	m_{rate} in 0–30 m: 0.7 m_{rate} in 30–50 m: 0.5 m_{rate} in $>$ 50 m: 0.3
NOMAE [23]	neighborhood size $n = 9$ $m_{\text{rate}} = 0.7$
SONATA [24]	mask view:2 local view:4 $m_{\text{rate}} : 0.4 \rightarrow 0.7$ Augmentations: GlobalRotScaleTrans, Flip3D, JitterPoints

Table 13: Comparison of $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$ on the K-Radar sequence split. Sup. denotes supervised learning. RAE is introduced in Section 3.2, and RAD in Section 3.4. The best results are highlighted in **bold**.

	Methods	Representation	Sedan	Bus/Truck	Pedestrian	mAP
Sup.	PTv3[15]	Point Cloud	46.9/41.5	26.7/20.6	0.5/0.1	24.7/20.8
		RAE	51.2/42.9	28.4/23.8	1.5/0.8	27.0/22.5
		RAD	49.2/41.5	23.8/17.4	18.7/14.9	30.6/24.6
Fine-tune	+OccupancyMAE[22]	Point Cloud	47.7/42.1	25.1/19.0	2.6/2.4	25.2/21.2
	+SONATA[24]	Point Cloud	49.2/42.5	29.2/23.2	1.8/1.2	26.8/22.3
	+NOMAE [23]	Point Cloud	48.7/42.6	28.5/23.4	1.6/1.1	26.3/22.4
	+SPMAE(Ours)	RAE	53.1/45.5	28.5/22.7	6.9/6.2	29.5/24.8
	+L-SPMAE(Ours)	RAE	53.7/46.4	29.3/24.1	5.0/4.2	29.3/24.9
	+SPMAE(Ours)	RAD	53.5/45.8	31.3/**25.9**	15.2/13.1	33.4/28.2
	+L-SPMAE(Ours)	RAD	**54.0/47.3**	**33.3**/21.8	**21.9/18.5**	**36.4/29.2**

In contrast, the 4D cube with Doppler-aware encoding achieves substantially better performance under both supervised and fine-tuning settings, highlighting the importance of effectively modeling Doppler information.

B.2 Probing Comparison

We provide probing comparisons on both the official K-Radar split (Table 14) and the sequence split (Table 15). On the official split, spectrum-based SSL consistently outperforms its point-cloud-based counterparts. In particular, L-SPMAE achieves gains of $(+3.4/+3.7)$ over SPMAE, demonstrating the effectiveness of the proposed leakage modeling during pretraining. On the sequence split, although NOMAE achieves competitive probing performance on Sedan and Bus/Truck categories, spectrum-based SSL shows clear superiority on Pedestrian and achieves the best overall mAP across all three classes. L-SPMAE also consistently outperforms SPMAE under this setting. Overall, these results demonstrate the advantage of spectrum-based SSL and validate the effectiveness of the proposed spectral leakage modeling strategy.

Table 14: Probing comparison of $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$ on the K-Radar official split. The best performance is highlighted in **bold**.

	PTv3	Representation	Sedan	Bus/Truck	Pedestrian	mAP
Probing	+OccupancyMAE[22]	Point Cloud	55.2/49.0	43.1/34.9	38.8/35.8	45.7/39.8
	+SONATA[24]	Point Cloud	55.8/49.7	41.3/31.9	38.0/36.5	45.0/39.4
	+NOMAE[23]	Point Cloud	56.0/49.9	43.6/33.4	38.2/35.8	46.0/39.7
	+SPMAE(Ours)	RAD	56.6/49.9	46.2/36.7	41.5/38.5	48.1/41.7
	+L-SPMAE(Ours)	RAD	**59.2/52.6**	**51.2/41.7**	**44.1/41.9**	**51.5/45.4**

B.3 Multi-object Tracking

We further evaluate the transferability of the learned representations on radar-based multi-object tracking, following the official K-Radar tracking protocol[3]. As shown in Table 16, both SPMAE and L-SPMAE outperform training from scratch, demonstrating effective transfer beyond object detection. L-SPMAE achieves the best AMOTA, MOTA, and recall, while SPMAE obtains a slightly better AMOTP.

B.4 Label-Efficient Fine-tuning

Table 17 reports L-SPMAE fine-tuned with varying fractions of labeled data. Pretraining consistently improves performance across all data scales, and fine-tuning with only 25% of labels already surpasses the fully supervised baseline trained on 100% of the data, demonstrating that signal discovery substantially reduces annotation requirements, which is particularly valuable for radar, where labeling

[3] https://github.com/kaist-avelab/K-Radar/blob/main/docs/tracking.md

Table 15: Probing comparison of $\mathrm{AP}_{\mathrm{BEV}}/\mathrm{AP}_{\mathrm{3D}}$ on the K-Radar sequence split. The best performance is highlighted in **bold**.

	PTv3	Representation	Sedan	Bus/Truck	Pedestrian	mAP
Probing	+OccupancyMAE[22]	Point Cloud	45.5/40.1	17.8/13.6	4.4/2.8	22.6/18.9
	+SONATA[24]	Point Cloud	45.2/39.8	14.3/10.2	3.8/3.5	21.1/17.8
	+NOMAE[23]	Point Cloud	48.0/**42.5**	**21.7/12.3**	4.3/3.9	24.7/19.6
	+SPMAE(Ours)	RAD	46.4/39.6	15.9/10.8	20.3/17.4	27.5/22.6
	+L-SPMAE(Ours)	RAD	**49.3**/42.3	16.9/11.4	**22.3/19.0**	**29.5/24.2**

Table 16: Radar-based multi-object tracking results on K-Radar. We use SimpleTrack [42] as the tracking framework.

Methods	AMOTA ↑	MOTA ↑	Recall ↑	AMOTP ↓
PTv3	0.529	0.486	0.567	1.038
+SPMAE(Ours)	0.562	0.517	0.607	**0.973**
+L-SPMAE(Ours)	**0.577**	**0.530**	**0.640**	0.975

is costly. Our pretraining is complementary to semi-supervised approaches that exploit unlabeled data during task training [43], and combining the two is a promising direction for future work.

B.5 Qualitative Results

Figure 5 gives qualitative results of how our L-SPMAE pretraining benefits the downstream task. Generally, the pretraining helps the networks better exploit spectral features, providing a more accurate prediction, particularly at long range.

C Running Environment

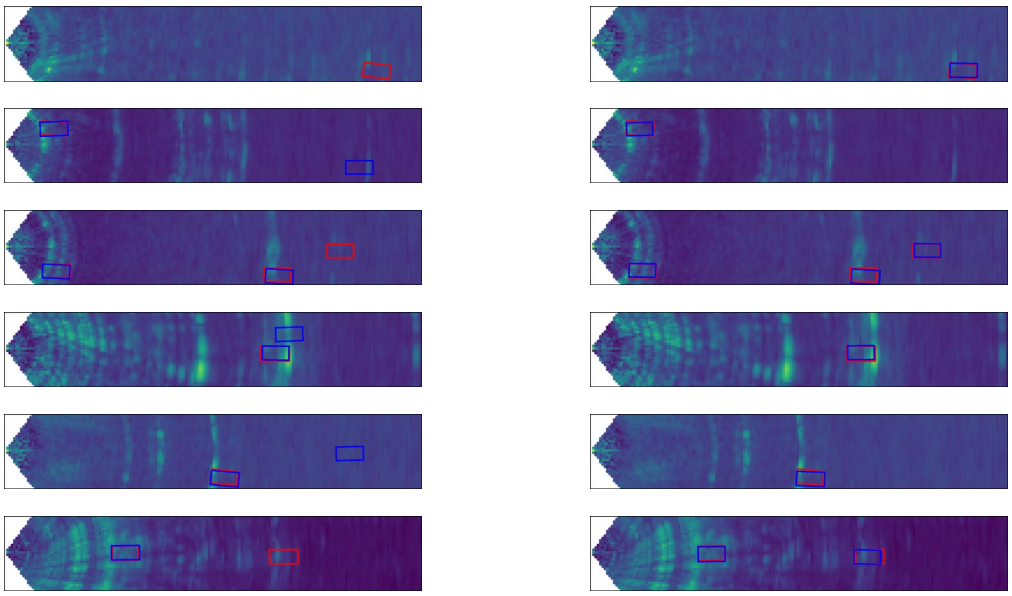
All experiments are conducted on a single NVIDIA B200 GPU. Our implementation is based on Pointcept [44], MMDetection3D [45], and OpenPCDet [46], using PyTorch 2.8, spconv 2.x, and FlashAttention2. Since FlashAttention2 requires GPUs with SM80 or newer architectures, Ampere or later generations are needed for reproduction. With a batch size of 8, our method requires at least 50 GB GPU memory for training and 13 GB for inference. The code will be publicly released after internal approval.

D Running Efficiency

Although the radar spectrum preserves substantially richer information than sparse point clouds and can benefit perception performance, it also introduces significantly higher computational cost. In particular, the raw RAE cube is approximately $10\times$ larger in data volume than the corresponding sparse point cloud representation. Table 18 compares the running speed on the same GPU device. Compared with point-cloud-based processing, 3D spectrum processing is $3.9\times$ slower during training and $4.8\times$ slower during inference, while 4D spectrum processing is $9.5\times$ slower during training and $11\times$ slower during inference. In future work, we plan to explore token pruning techniques [41] for spectrum representations to further reduce the computational cost.

Table 17: Fine-tuning mAP on the K-Radar official split with different annotated subsets.

Ratio (%)	10	25	50	75	100
From Scratch	12.7/8.9	42.2/35.7	43.5/37.9	45.4/38.9	48.9/43.1
+L-SPMAE	34.5/28.1	50.4/44.4	52.4/47.0	54.5/48.5	55.9/50.1



Supervised Learning Results *Fine-tuned Results*

Figure 5: Prediction results on Bird's-eye view. (ground truth; prediction).

Table 18: Comparison of running speed with PTv3 for object detection on a single NVIDIA B200 GPU, where each iteration uses a batch size of 8. Note that the reported speed is not the upper performance bound, since the implementation is not fully hardware-optimized.

Representation	Training iterations/s	Inference iterations/s
Point Cloud	6.7	14.3
3D Spectrum	1.7	3.0
4D Spectrum	0.7	1.3

E Algorithm Details

The details of spectral leakage modeling are illustrated in Algorithm 1. We first map the spectrum points into a generalized 3D spectral grid for modeling the leakage effect directly in the spectral domain. The three spectral axes are not restricted to a specific representation and can correspond to different coordinate parameterizations, such as RAE or RAD. To formulate the 3D leakage kernel, we construct three normalized 1D leakage kernels from the block-wise masks along each spectral axis. In practice, the convolution is efficiently implemented as $\mathrm{IFFT}\big(\mathrm{FFT}(S) \odot \mathrm{FFT}(K)\big)$ according to the convolution theorem. For training stability, we further rescale the corrupted spectrum to preserve the total spectral energy of the original spectrum.

Algorithm 1 Spectral Leakage Modeling

Require: Points $\{(\mathbf{x}_i, f_i)\}_{i=1}^{P}$; grid mapping $\pi(\cdot)$; grid size (D_1, D_2, D_3); leakage rate λ; block size b
Ensure: Corrupted point powers $\{f_i'\}_{i=1}^{P}$

Step 1: Map points to a generalized 3D spectral grid.

Compute spectral-grid indices:

$$\mathbf{u}_i \leftarrow \pi(\mathbf{x}_i), \qquad \mathbf{u}_i = (u_i^{(1)}, u_i^{(2)}, u_i^{(3)}).$$

Scatter point powers to a dense spectral grid:

$$S \in \mathbb{R}^{D_1 \times D_2 \times D_3}, \qquad S[\mathbf{u}_i] \leftarrow S[\mathbf{u}_i] + f_i.$$

Step 2: Generate block-wise leakage masks.

for $j = 1, 2, 3$ **do**
  Let D_j denote the length of the j-th spectral axis.
  Sample a 1D block-wise binary mask:

$$m_j \leftarrow \mathrm{BlockMask}(D_j, \lambda, b),$$

  where λ controls the leakage strength and b controls the spatial extent of corrupted regions.
end for

Step 3: Convert block masks to leakage kernels.

for $j = 1, 2, 3$ **do**
  Compute the spectral power response:

$$\bar{K}_j \leftarrow |\mathrm{FFT}(m_j)|^2.$$

  Normalize into a 1D leakage kernel:

$$K_j \leftarrow \frac{\bar{K}_j}{\sum_{q=1}^{D_j} \bar{K}_j[q]}.$$

end for

Step 4: Construct a separable 3D leakage kernel.

Form the separable kernel:

$$K[\mathbf{u}] \leftarrow K_1[u^{(1)}] \cdot K_2[u^{(2)}] \cdot K_3[u^{(3)}].$$

Step 5: Apply leakage by FFT-based convolution.

Apply circular convolution:

$$\tilde{S} \leftarrow \Re\left\{\mathrm{IFFT}\left(\mathrm{FFT}(S) \odot \mathrm{FFT}(K)\right)\right\}.$$

Step 6: Preserve total spectral energy.

Compute spectral energies:

$$E_{\mathrm{in}} \leftarrow \sum_{\mathbf{u}} S[\mathbf{u}], \qquad E_{\mathrm{out}} \leftarrow \sum_{\mathbf{u}} \tilde{S}[\mathbf{u}].$$

Calibrate the leaked spectrum:

$$\tilde{S} \leftarrow \frac{E_{\mathrm{in}}}{E_{\mathrm{out}} + \epsilon} \tilde{S}.$$

Step 7: Gather corrupted values back to points.

Assign corrupted powers:

$$f_i' \leftarrow \tilde{S}[\mathbf{u}_i].$$

return $\{f_i'\}_{i=1}^{P}$

F Broader Impact

This work investigates self-supervised learning on dense radar spectrum representations for 3D perception. The proposed method may improve the robustness of radar-based perception systems for safety-critical applications such as autonomous driving, especially under adverse weather conditions. Potential risks include incorrect detections in downstream autonomous systems and possible misuse in surveillance scenarios. These risks can be mitigated through careful validation and evaluation before deployment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state the paper's contributions and scope in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We present the technical details of our method in Section 3.3 and Appendix E. Implementation details are provided in Section 4.3 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code will be publicly released after internal approval. All data used in this work are from publicly available datasets.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide those details in Section 4.3 and Appendix A.1, A.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are omitted due to the substantial computational cost. Nevertheless, for most results reported in Table 1 and Table 13, which contain our main findings, we mitigate the effect of random seed variability by averaging over three independent runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we have specified it in Appendix C, D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses the broader impacts in Appendix F.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all related works.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.