

Building 6G Radio Foundation Models with Transformer Architectures

Ahmed Aboulfotouh[‡], Ashkan Eshaghbeigi^{*}, and Hatem Abou-Zeid[‡]
[‡]Department of Electrical and Software Engineering, University of Calgary, Canada
^{*}Qoherent Inc., Toronto, Ontario, Canada

Abstract—Foundation deep learning (DL) models are general models, designed to learn general, robust and adaptable representations of their target modality, enabling finetuning across a range of downstream tasks. These models are pretrained on large, unlabeled datasets using self-supervised learning (SSL). Foundation models have demonstrated better generalization than traditional supervised approaches, a critical requirement for wireless communications where the dynamic environment demands model adaptability. In this work, we propose and demonstrate the effectiveness of a Vision Transformer (ViT) as a *radio foundation model* for spectrogram learning. We introduce a Masked Spectrogram Modeling (MSM) approach to pretrain the ViT in a self-supervised fashion. We evaluate the ViT-based foundation model on two downstream tasks: Channel State Information (CSI)-based Human Activity Sensing and Spectrogram Segmentation. Experimental results demonstrate competitive performance to supervised training while generalizing across diverse domains. Notably, the pretrained ViT model outperforms a four-times larger model that is trained from scratch on the spectrogram segmentation task, while requiring significantly less training time, and achieves competitive performance on the CSI-based human activity sensing task. This work demonstrates the effectiveness of ViT with MSM for pretraining as a promising technique for scalable foundation model development in future 6G networks.

Index Terms—Self-Supervised Learning, Foundation Models, Deep Learning, Human Activity Sensing, Spectrogram Segmentation

I. INTRODUCTION

Foundation models (FMs) are first trained on a large, often unlabeled dataset, allowing them to build broad, adaptable representations that can be finetuned for various downstream tasks. This initial pretraining stage is done using self-supervised learning (SSL), where the model learns underlying patterns and relationships within the data without relying on labeled examples [1]–[3]. The model ideally develops a robust understanding of its target modality, which, in our case, is radio spectrograms.

In fields like computer vision and natural language processing, FMs have set new benchmarks [4]–[7], often surpassing supervised learning models, specifically designed for individual tasks. This is largely due to their ability to generalize: FMs learn flexible and transferable representations that make them better suited to handle variations in data, perform across diverse tasks, and adapt to new contexts. Generalization is especially valuable when labeled data is scarce, as foundation models can perform well with minimal additional labeled samples.

Deep learning (DL) has demonstrated strong potential when applied to individual wireless tasks, including automatic modulation classification [8], channel estimation [9], constellation

and waveform design [10], among others. However, these models are highly specialized, and there are concerns about their ability to generalize effectively in real-world scenarios. Wireless signals are subject to time-varying impairments, and the communication environment is constantly changing, which can degrade a DL model’s performance if it fails to adapt. Introducing the concept of FMs for wireless can potentially overcome these limitations [11], [12].

We propose FMs for wireless signals as a solution to address these challenges. By capturing over-the-air radio signals and pretraining FMs through SSL, there is no need for labeled data. Additionally, these pretrained models can then serve as backbones for multiple tasks, reducing computational costs. Most importantly, FMs are expected to achieve better generalization by leveraging their broad, transferable representations, making them well-suited to handle diverse and dynamic wireless environments. The primary contributions of our paper are:

- We propose and demonstrate the effectiveness of a Vision Transformer (ViT) as a radio foundation model for spectrogram learning. Adopting ViT as the FM offers enhanced flexibility, particularly in handling variable input sequences, and increased scalability, as training and evaluation can be parallelized. ViT also captures long-term dependencies through its attention mechanisms.
- We introduce a Masked Spectrogram Modeling (MSM) approach to pretrain the ViT in a self-supervised fashion, and thoroughly evaluate key design considerations of the masking procedure and transformer size on performance.
- By finetuning across two downstream tasks, we demonstrate that the ViT radio FM effectively learns features that generalize across diverse domains, achieving competitive—or even superior—performance with 4x smaller model sizes compared to baselines.
- We demonstrate the effectiveness of the proposed foundation model by utilizing a real-world dataset that is captured over-the-air in a software-defined radio testbed. Upon acceptance, the datasets and code will be publicly available to encourage further research within the community on FM for wireless.

The remainder of the paper is structured as follows: Section II presents the datasets utilized for pretraining the foundation model, and for the CSI-based human activity sensing and spectrogram segmentation tasks. Section III outlines the ViT architecture and algorithm of the self-supervised foundation

model. Section IV presents numerical experiments conducted to evaluate the proposed methodology. Finally, section V concludes the paper.

II. TESTBED AND DATASETS

We use three datasets in this paper. The first, the Real-time Radio Dataset (RRD), consists of over-the-air radio recordings captured in real-time with a software-defined radio (SDR) test bed built using PlutoSDRs. The second, the Human Sensing Dataset (HSD), utilizes Wi-Fi channel state information (CSI) to detect human activity in an indoor environment. The third dataset, the Segmentation Dataset (SD), simulates 5G New Radio (NR) and LTE transmissions in neighboring frequency bands.

A. Real-time Radio Dataset (RRD)

The RRD dataset consists of recordings of IQ samples, representing both in-phase (I) and quadrature (Q) components of the RF signal. Each recording is captured with a center frequency (ranging from 2.4 to 2.65 GHz), sampling frequency (between 10 MHz and 60 MHz), and duration, typically averaging around 100 ms. Data collection took place in downtown Toronto, Canada, resulting in 240 recordings, which cover approximately 24 seconds of RF activity. This dataset is used for initial model pretraining.

Spectrogram Computation. We create spectrograms from IQ recordings through the following steps: 1) Divide each recording into non-overlapping 16 ms segments; 2) Compute the spectrogram for each segment using the short-time Fourier transform (STFT); 3) Resize each spectrogram to a 224×224 shape; 4) Convert the spectrogram to log scale; 5) Normalize and standardize using dataset-wide statistics.

The dataset parameters are summarized in Table I. Learning performance is generally robust to these specific parameter choices, which are selected to balance computational efficiency and preserve information-rich content.

B. Human Activity CSI-Based Sensing Dataset (HSD)

The HSD dataset contains CSI measurements for six human activities: running, walking, falling, boxing, arm circling, and floor cleaning [13]. Each subject performs these activities between a pair of Wi-Fi access points, each equipped with three antennas. CSI is measured for each activity, across 114 subcarriers and 3 channels (one per antenna) over 2000 samples at a 500 Hz rate. Each recording is thus a 3D tensor of shape $3 \times 114 \times 2000$, paired with its activity label. The CSI is processed by resizing each recording to a shape of $3 \times 224 \times 224$. Then, each channel is normalized and standardized using dataset-wide statistics. A sample from each class of the dataset is illustrated in Figure 1 where the horizontal axis is time and the vertical axis is frequency.

C. NR-LTE Segmentation Dataset (SD)

The SD dataset is created by generating NR and LTE signals, each transmitted through its respective wireless channel in

TABLE I: RRD Dataset Generation Parameters

Parameters		Value
STFT Parameters	FFT Size	1024
	Window Function	Hanning
	Window Size	512
	Hop Size	512
Slicing Parameters	Duration	16 ms
	Resizing Shape	(224, 224)

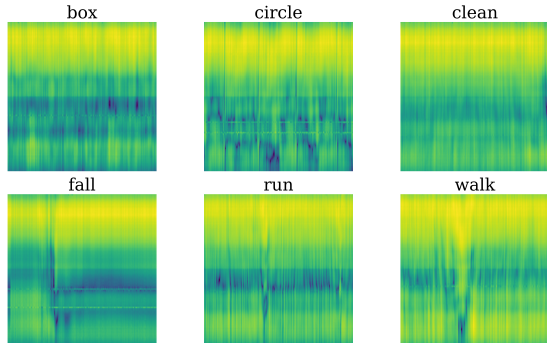


Fig. 1: A sample from each class of the HSD dataset. Only the CSI of the first antenna is plotted.

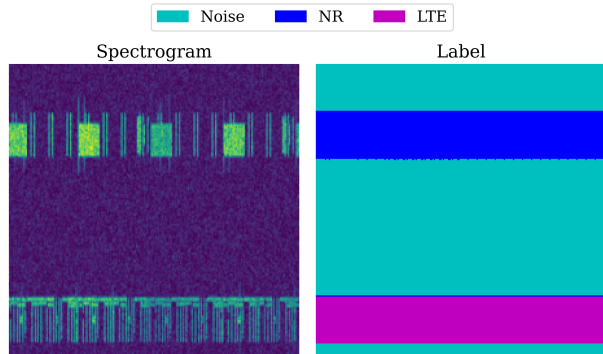


Fig. 2: A spectrogram and its segmentation from the SD dataset.

adjacent, non-overlapping bands. We use the Matlab Communication Toolbox for signal generation, following the guidelines in [14].

A spectrogram of the NR-LTE signal mixture is computed and resized to 224×224 . A corresponding label image is also created, marking NR signals as 1, LTE signals as 2, and noise as 0. For more details about data generation, refer to [15]. A sample is illustrated in Figure 2 where the horizontal axis represents time and the vertical axis represents frequency.

III. VISION TRANSFORMER FOUNDATION MODEL FOR SPECTROGRAM LEARNING

A. Masked Spectrogram Modeling (MSM)

We introduce the Masked Spectrogram Modeling (MSM) approach using Vision Transformers (ViT). In this method, we divide each spectrogram image into $p \times p$ patches and randomly sample a subset of these patches using a uniform distribution.

The goal is to reconstruct the missing patches from only the sampled subset, while the remaining patches—effectively the masked patches—are excluded. This approach is illustrated in Figure 3. While this approach resembles a traditional auto-encoder, a key difference is that the model is trained to reconstruct the masked patches only, rather than the full set.

We employ high masking ratios (e.g., 80%) as in [7] to reduce redundancy and make reconstruction more challenging. This forces the model to rely less on extrapolation from visible patches, effectively avoiding learning features that are more local, and instead emphasizing general characteristics that contribute to the overall representation of the spectrogram, its underlying structure and statistical patterns.

This approach offers several advantages: masking a large portion of the spectrogram and only processing the visible patches makes pretraining more efficient. This method requires no labeled data, recordings can be captured over-the-air using software-defined radios (as we have done with the RRD dataset) and fed into the model directly, making large-scale pretraining more feasible.

B. Spectrogram Masked ViT Autoencoder

As shown in Figure 3, we use an encoder-decoder architecture based on a ViT masked autoencoder [7], [16]. This design is asymmetric in several respects. The encoder processes the visible patches outputting feature tokens, and the decoder handles the feature and mask tokens. The decoder reconstructs the original spectrogram by attending to the feature tokens provided by the encoder.

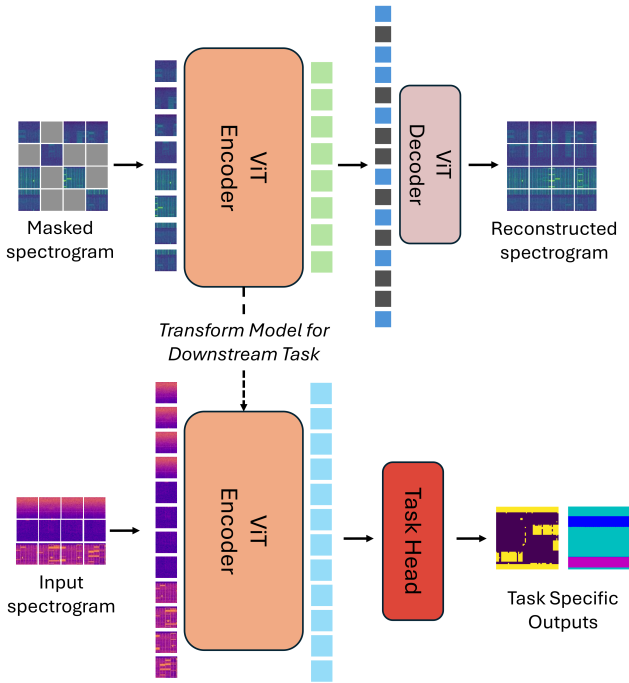


Fig. 3: Proposed ViT Foundation Model for Radio Spectrograms.

Algorithm 1: Masked Spectrogram Modeling with ViT

Input : spectrogram dataset \mathcal{D} , initial model \mathcal{M} , patch size p , mask ratio γ

Output: foundation model \mathcal{B}

$\mathcal{B} \leftarrow$ encoder of the ViT model \mathcal{M}

repeat

foreach $spect_sample$ in \mathcal{D} **do**

$patches \leftarrow$ PATCHIFY($spect_sample, p$)

$visible_patches \leftarrow$ SAMPLE($patches, \gamma$)

$encoder_in \leftarrow \mathcal{M} \cdot$ EMBED($visible_patches$)

$encoder_in \leftarrow$ POS_EMBED($encoder_in$)

$encoder_out \leftarrow \mathcal{M} \cdot$ ENCODE($encoder_in$)

$decoder_in \leftarrow \mathcal{M} \cdot$ DECODER_EMBED($encoder_out$)

$decoder_in \leftarrow$ APPEND_REORDER($decoder_in$)

$decoder_in \leftarrow$ POS_EMBED($decoder_in$)

$decoder_out \leftarrow \mathcal{M} \cdot$ DECODE($decoder_in$)

$recon_patches \leftarrow$ UNPATCHIFY($decoder_out$)

$loss \leftarrow \mathcal{L}_{MSM}(recon_patches, visible_patches)$ as per equation (1)

BACKWARD($\mathcal{M}, loss$)

until convergence is reached or another stopping condition is met;

Function	Description
PATCHIFY	Splits the input spectrogram into smaller patches of a specified size $p \times p$.
SAMPLE	Selects a subset of patches based on the mask ratio γ .
EMBED*	Prepare the visible patches for the encoder by mapping them to its embedding space.
POS_EMBED	Adds sinusoidal positional embeddings to its input.
ENCODE*	Processes the embedded patches through the encoder transformer blocks.
DECODER_EMBED*	Prepares the encoder output for the decoder by mapping it to the decoder embedding space.
APPEND_REORDER	Reorders tokens and inserts mask tokens to restore the original time-frequency order.
DECODE*	Processes the reordered tokens through the decoder transformer blocks, producing the reconstructed patches.
UNPATCHIFY	Combines reconstructed patches back into a complete spectrogram.
BACKWARD	Computes gradients and updates the model using Backpropagation

TABLE II: Explanation of Functions in Algorithm 1 (* denotes functions called through the model)

Masked tokens are learnable embeddings which are positioned in the original locations of the masked patches (i.e., not inputted to the encoder).

The encoder is larger than the decoder in terms of capacity, it performs the majority of the computation. As a result, the encoder can function independently as a feature extractor, while the decoder can be discarded. The approach is detailed in Algorithm 1. In the following, we provide the high-level details

of the ViT architecture.

Encoder. Each input patch is embedded using a linear projection, and sinusoidal positional embeddings are added to create a token. The purpose of the positional embeddings is to indicate the order, as transformers lack a built-in ordering mechanism. The tokens are then processed through a series of transformer blocks, producing the output feature tokens.

Decoder. At the decoder, a linear projection is applied to match the feature token dimension to the decoder embedding dimension. The original time-frequency ordering of the resulting tokens is restored with mask tokens inserted in place of the masked patches. Sinusoidal positional embeddings are added as well. The sequence of tokens is then processed by a series of transformer blocks and the output is the reconstructed spectrogram.

By processing only a subset of the tokens using the larger encoder and handling the full set with the smaller decoder, this design enables the training of much larger models without extensive computational resources.

Objective. We train the model in a self-supervised way to reconstruct the masked patches. The loss function \mathcal{L}_{MSM} of MSM task can be written as:

$$\mathcal{L}_{\text{MSM}} = \frac{1}{NM} \sum_{n=1}^N \sum_{ij} \left\| \text{vec}(\mathbf{X}_{ij}^{(n)}) - \text{vec}(\hat{\mathbf{X}}_{ij}^{(n)}) \right\|_2^2 \mathbb{I}_{\text{mask}}(n, i, j) \quad (1)$$

where N is the batch size, M is the total number of patches, $\mathbf{X}_{ij}^{(n)} \in \mathbb{R}^{P \times P}$ is the input patch at position (i, j) in sample n , and $\hat{\mathbf{X}}_{ij}^{(n)} \in \mathbb{R}^{P \times P}$ denotes the reconstructed patch at position (i, j) for sample n . The vectorization operation vec flattens each patch into a vector, $\|\cdot\|_2$ is the L_2 norm and $\mathbb{I}_{\text{masked}}(n, i, j)$ is an indicator function that outputs 1 if patch (i, j) in sample n was masked and 0 otherwise.

The encoder of the self-supervised pretrained ViT masked autoencoder serves as our *radio foundation model* which can be finetuned for downstream tasks. We finetune for two downstream tasks: CSI-based human activity sensing and spectrogram segmentation, introduced next.

C. CSI-based Human Activity Sensing

The task is to classify CSI measurements into one of six distinct human activity classes. We utilize the ViT encoder from the pretrained model as a feature extractor, adding a linear layer as a classification head on top. The ViT encoder is entirely frozen, only the linear classifier is finetuned on the dataset. The pretrained model was originally trained on single-channel spectrograms, whereas here the input is a three-channel tensor representing the CSI. To accommodate this difference, the positional embeddings are modified to align with the new input, while the remainder of the encoder remains unchanged. The CSI data is divided into patches in the same manner as the spectrograms. The model outputs a softmax probability vector, and the loss function is the label smoothing cross-entropy,

defined as:

$$\mathcal{L}_{\text{HSD}} = -\frac{1}{N} \sum_{n=1}^N \sum_{i=1}^C \left(y_i^{(n)} \cdot (1 - \alpha) + \frac{\alpha}{C} \right) \cdot \log \left(\hat{y}_i^{(n)} \right) \quad (2)$$

where N is the batch size, $C = 6$ is the number of classes, $y_i^{(n)} \in [0, 1]$ is the true label for class i (either 0 or 1 for sample n), $\hat{y}_i^{(n)} \in [0, 1]$ is the model's predicted probability for class i , and $\alpha \in (0, 1)$ is the smoothing factor. Unlike traditional cross-entropy, label smoothing distributes a small probability to incorrect labels, preventing the model from becoming overly confident which enhances generalization. The degree of smoothing is determined by α .

D. Spectrogram Segmentation

The task is to segment the input spectrogram into three classes: noise, NR signal, and LTE signal. We use the pretrained ViT encoder as a feature extractor, adding two standard transformer decoder blocks on top as a segmentation head. The ViT encoder is kept frozen, and only the decoder is finetuned. Since the input is a spectrogram, no modifications are made to the positional embeddings. The model's output is a 3D tensor providing a probability distribution for each pixel in the segmented spectrogram. We use label smoothing cross-entropy as the loss function, which is defined as follows:

$$\mathcal{L}_{\text{SG}} = -\frac{1}{NM} \sum_{n=1}^N \sum_{k=1}^C \sum_{ij} \left(y_{ijk}^{(n)} \cdot (1 - \alpha) + \frac{\alpha}{C} \right) \cdot \log \left(\hat{y}_{ijk}^{(n)} \right) \quad (3)$$

Here, M is the total number of pixels in the segmented image, $C = 3$ is the number of classes, $y_{ijk}^{(n)} \in [0, 1]$ is the correct label at pixel (i, j) for class k in sample n , $\hat{y}_{ijk}^{(n)} \in [0, 1]$ is the predicted probability at pixel (i, j) for class k , and α is the smoothing factor.

IV. RESULTS AND DISCUSSION

We perform self-supervised pretraining with masking on the RRD dataset, then evaluate the learned representations by finetuning. For finetuning, the decoder is discarded, and the frozen ViT encoder serves as a feature extractor, with only the task-specific head updated. No masking is done during finetuning. Three models are pretrained: ViT-S (small), ViT-M (medium), and ViT-L (large), with details provided in Table III. Here, different sizes refer to the encoder, while the decoder remains largely unchanged. First, we evaluate the models' reconstruction performance across various masking ratios, followed by assessing generalization capabilities on the CSI Sensing and Segmentation datasets.

A. Reconstruction Performance

First, we showcase reconstruction examples for ViT-M from which it is clear that the model exhibits strong performance. This is illustrated in Figure 4. Each row shows the original spectrogram on the left, followed by the masked spectrogram and the corresponding model's reconstruction for different masking ratios. The reconstructed spectrograms closely match the originals, with reasonable differences. To evaluate the

TABLE III: Pretrained ViT Models

Model	Encoder					Decoder			
	patch size	embed dim*	depth†	attn. heads	# params (M)	embed dim*	depth†	attn. heads	# params (M)
ViT-S	16	512	12	8	38	256	8	16	7
ViT-M	16	768	12	12	85	512	8	16	26
ViT-L	16	1024	24	16	302	512	8	16	26

* *embed dim* is the embedding dimension which is also known as the transformer width.
 † *depth* is the number of transformer blocks.

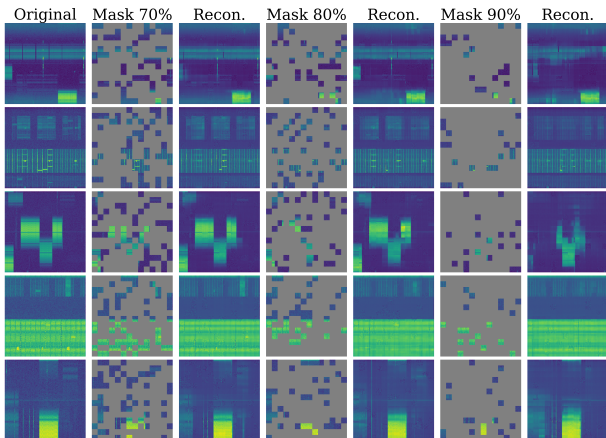


Fig. 4: Reconstruction results of ViT-M at various masking ratios pretrained with a 75% masking ratio.

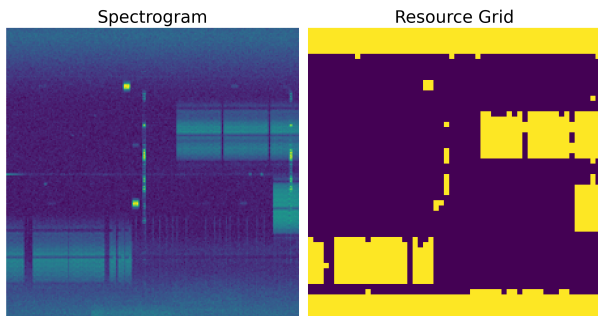


Fig. 5: A spectrogram and its corresponding resource grid using a pooling filter of size 4.

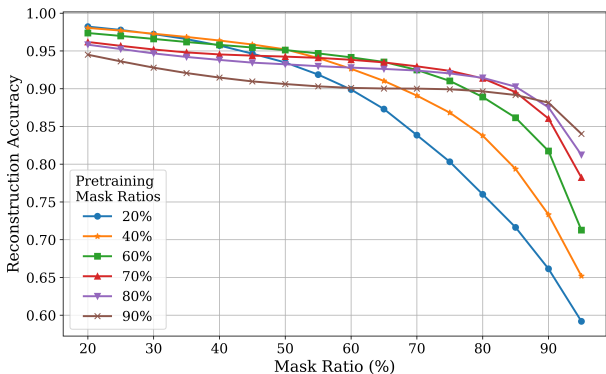


Fig. 6: Reconstruction Accuracy vs Mask ratio of ViT-S pretrained at various masking ratios.

reconstruction capability of the models, we need a robust reconstruction accuracy metric. Relying solely on visual comparison is not enough. Hence, we transform each spectrogram into a resource grid composed of resource blocks. To transform the spectrogram, average pooling is first applied without overlap between pooled patches (i.e., stride equal to the kernel size). A threshold, δ , is then applied to the pooled grid to binarize it, designating vacant resource blocks as 0 and occupied ones as 1. The threshold δ is determined empirically by the formula:

$$\delta = \mu + 0.5 \times \sigma \quad (4)$$

where μ and σ represent the mean and standard deviation of the spectrogram, respectively. A sample for the transformation is illustrated in Figure 5. We then evaluate the models’ reconstruction performance across various masking ratios, including but not limited to those used during pretraining. A model’s robustness is measured by its ability to maintain strong reconstruction performance even when applied to masking ratios it was not trained on. Although higher masking ratios increase reconstruction difficulty, the model is expected avoid collapse. As illustrated in Figure 6, pretrained models with higher masking ratios maintain their strong performance when dealing with different masking ratios. Similar to results for vision and audio, the ideal masking ratio for pretraining is around 70% to 80%. Hence, we only finetune the models pretrained with these masking ratios.

B. Finetuning Performance

To evaluate finetuning performance on the HSD and SD classification and segmentation datasets, we use confusion matrices (per-class accuracy) and overall accuracy. We present finetuning results for the HSD dataset first, followed by the SD dataset. For both datasets, we use the pretrained ViT encoder as a feature extractor which is kept frozen, and finetune the task-specific head. Table IV summarizes the accuracy results, including models pretrained at masking ratios of 70%, 75%, and 80%, as well as a baseline model trained from scratch directly on the HSD dataset. The highest accuracy is achieved by ViT-M trained from scratch, with a 5% accuracy margin compared to the pretrained ViT-M model. We attribute this difference to the inherent distinctions between CSI data and spectrograms, suggesting that more extensive pretraining could reduce this gap. Figure 7 displays the confusion matrices for ViT-M pretraining at a 75% masking ratio versus training from scratch. The primary source of accuracy differences lies in the

pretrained model’s tendency to confuse run and walk, due to their close distribution.

For the SD dataset, Table V presents the results, including models pretrained with masking ratios of 70%, 75%, and 80%, alongside a baseline model trained from scratch.

TABLE IV: Mean accuracy of ViT finetuned on the HSD dataset, pretrained at masking ratios of 70%, 75%, and 80%. The table also includes results for a model trained from scratch.

Model	Masking ratio (%)			Scratch
	70	75	80	
ViT-S	90.2	90.9	89.0	98.1
ViT-M	92.0	93.9	85.9	98.9
ViT-L	89.3	88.6	85.6	98.1

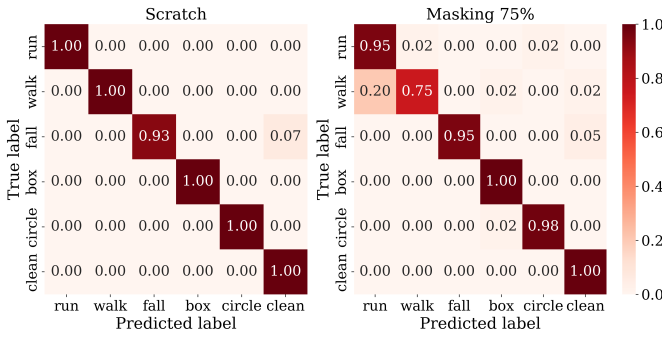


Fig. 7: Confusion matrices of ViT-M trained from scratch and pretrained with a 75% masking ratio.

TABLE V: Mean accuracy of ViT finetuned on the SD dataset, pretrained at masking ratios of 70%, 75%, and 80%.

Model	Masking ratio (%)			Scratch
	70	75	80	
ViT-S	97.0	96.8	96.4	97.2
ViT-M	97.9	97.6	97.5	97.1
ViT-L	97.5	97.3	97.5	97.7

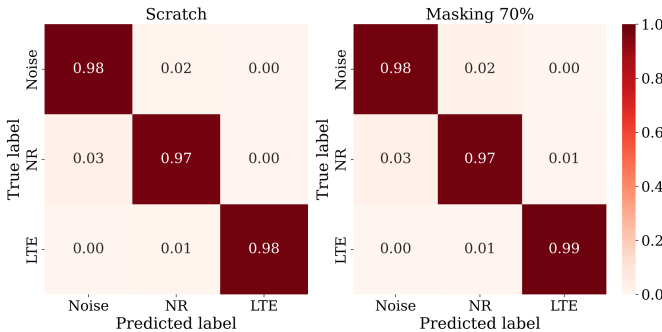


Fig. 8: Confusion matrices of ViT-L trained from scratch and ViT-M pretrained with a 70% masking ratio.

The best model is the pretrained ViT-M with a 70% masking ratio, which slightly outperforms the best scratch-trained model, ViT-L, while being four times smaller. Figure 8 provides confusion matrices for these models.

V. CONCLUSION

In this paper, we proposed ViT as a *radio foundation model* for spectrogram learning which offers superior modelling capabilities, support for variable-length input sequences and computational efficiency. We also introduce a Masked Spectrogram Modeling (MSM) approach to pretrain the ViT in a self-supervised fashion, and thoroughly evaluate the effects of masking ratios and transformer size on performance. Experimental results indicate that the ViT-based model generalizes well to unseen datasets, achieving comparable or superior performance to larger models trained from scratch, while utilizing fewer resources. Notably, the pretrained ViT model surpasses a four-times larger scratch-trained model on the spectrogram segmentation task and achieves competitive performance on the CSI-based human activity sensing task. We believe that this ViT-enabled MSM will enable scalable, large-scale pretraining, fostering the development of robust radio foundation models capable of generalizing across a wide range of tasks.

REFERENCES

- [1] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, “Self-supervised learning: Generative or contrastive,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 857–876, 2023.
- [2] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, “Self-Supervised Representation Learning: Introduction, advances, and challenges,” *IEEE Signal Processing Magazine*, vol. 39, pp. 42–62, May 2022.
- [3] Z. Yang, H. Du, D. Niyato, X. Wang, Y. Zhou, L. Feng, F. Zhou, W. Li, and X. Qiu, “Revolutionizing wireless networks with self-supervised learning: A pathway to intelligent communications,” *ArXiv:2406.06872*, 2024.
- [4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018.
- [5] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, “Conditional image generation with pixelcnn decoders,” *CoRR*, vol. abs/1606.05328, 2016.
- [6] P. Goyal, M. Caron, B. Lefaudeaux, M. Xu, P. Wang, V. Pai, M. Singh, V. Liptchinsky, I. Misra, A. Joulin, and P. Bojanowski, “Self-supervised pretraining of visual features in the wild,” *ArXiv*, vol. abs/2103.01988, 2021.
- [7] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” *ArXiv:2111.06377*, 2021.
- [8] F. Meng, P. Chen, L. Wu, and X. Wang, “Automatic modulation classification: A deep learning enabled approach,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10760–10772, 2018.
- [9] X. Wei, C. Hu, and L. Dai, “Deep learning for beamspace channel estimation in millimeter-wave massive mimo systems,” *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 182–193, 2021.
- [10] F. Ait Aoudia and J. Hoydis, “Waveform learning for next-generation wireless communication systems,” *IEEE Transactions on Communications*, vol. 70, no. 6, pp. 3804–3817, 2022.
- [11] J. Fontaine, A. Shahid, and E. De Poorter, “Towards a wireless physical-layer foundation model: Challenges and strategies,” in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–7, 2024.
- [12] L. Bariah, Q. Zhao, H. Zou, Y. Tian, F. Bader, and M. Debbah, “Large generative ai models for telecom: The next big thing?,” *IEEE Communications Magazine*, pp. 1–7, 2024.
- [13] J. Yang, X. Chen, H. Zou, D. Wang, Q. Xu, and L. Xie, “Efficientfi: Toward large-scale lightweight wifi sensing via csi compression,” *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13086–13095, 2022.

- [14] "Spectrum sensing with deep learning to identify 5g and lte signals," *Matlab Tutorials*.
- [15] A. Aboulfotouh, A. Eshaghbeigi, D. Karslidis, and H. Abou-Zeid, "Self-supervised radio pre-training: Toward foundational models for spectrogram learning," in *GLOBECOM 2024 - 2024 IEEE Global Communications Conference*, 2024. To appear.
- [16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv:2010.11929*, 2021.