Self-Supervised Radio Pre-training: Toward Foundational Models for Spectrogram Learning

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

***Abstract*—Foundational deep learning (DL) models are general models, trained on large, diverse, and unlabelled datasets, typically using self-supervised learning techniques - and have led to significant advancements especially in natural language processing. These pretrained models can be fine-tuned for related downstream tasks, offering faster development and reduced training costs, while often achieving improved performance. In this work, we introduce Masked Spectrogram Modeling, a novel self-supervised learning approach for pretraining foundational DL models on radio signals. Adopting a Convolutional LSTM architecture for efficient spatio-temporal processing, we pretrain the model with an unlabelled radio dataset collected from over-the-air measurements. Subsequently, the pretrained model is fine-tuned for two downstream tasks: spectrum forecasting and segmentation. Experimental results demonstrate that our methodology achieves competitive performance in both forecasting accuracy and segmentation, validating its effectiveness for developing foundational radio models.**

***Index Terms*—Self-Supervised Learning, Deep Learning, Foundational Models, Spectrum Forecasting, Spectrum Segmentation**

I. Introduction

A foundational model is a general model pretrained on a large-scale - usually unlabeled - dataset, typically through self-supervised learning [1]. Through this training, the model develops a solid understanding of the target modality, such as text in natural language processing (NLP) or images in computer vision. This understanding allows the model to be fine-tuned for diverse downstream tasks. Foundational models in NLP [2], [3] and computer vision [4] have driven significant advancements through leveraging the knowledge encoded in their pretrained representations. This facilitates quicker experimentation, more efficient resource utilization, and potentially, improved performance on downstream tasks that smaller models or those with more limited domain knowledge cannot achieve.

Deep learning has showcased promising results when applied in wireless communication [5]. The effectiveness has been demonstrated across various tasks, including automatic modulation classification [6], channel estimation [7], constellation and waveform design [8], among others. However, these models are highly specialized, echoing the early stages of deep learning's evolution in NLP and computer vision. The reliability of these models across data distribution shifts and their ability to generalize is also usually limited.

Introducing the concept of foundational models into wireless communication holds substantial promise to overcome these limitations [9]. We argue that as in NLP and computer vision, where a wealth of unlabeled data exists — communication signals can be harnessed for pretraining such foundational models through self-supervised learning, mitigating the expense associated with data labeling. Moreover, leveraging a foundational model as a backbone for multiple downstream tasks, which utilize its pretrained representations in subsequent processing, reduces computational demands. This approach can also improve generalization by leveraging the broader knowledge encoded within foundational model representations compared to highly specialized models which suffer from limited scope.

Drawing inspiration from these advancements, particularly in [2], [4], [10], we introduce a foundational radio model pretrained using masked spectrogram modelling (MSM) — a novel technique, we propose for wireless signals. This model is then fine-tuned to perform two different downstream tasks: spectrogram forecasting, which involves predicting future spectrogram based on past data, and spectrogram segmentation, which consists of distinguishing between background noise and other signal activities within the spectrogram. These tasks, while different, are complementary in the context of spectrum analysis and constitute a usage scenario for a foundational model integrated in a opportunistic spectrum access system. The primary contributions of our paper are:

- We propose and develop a novel self-supervised learning approach, MSM, for pre-training foundational models on radio signals. To the best of our knowledge, this work represents the first demonstration of radio foundational models for spectrogram learning using unlabeled data.
- We demonstrate the effectiveness of the proposed approach utilizing a real-world dataset that we collected over a software-defined radio testbed. The recordings are time-domain IQ samples received between 2.4 to 2.65 GHz.
- Our results show that the developed MSM approach is able to learn features that generalize to both related and unrelated downstream tasks. Fine-tuning the foundational model demonstrated competitive results for spectrum fore-

The authors would like to thank Qoherent Inc. and MITACS Accelerate for their support of this research. The authors would also like to thank Denvr Dataworks, Calgary, Canada for their high-performance compute used to conduct this research.

casting and spectrum segmentation which had a distinctly different and unseen data distribution.

The results of this paper highlight the significant potential that radio foundational models have to effectively enable multiple downstream spectrogram tasks. It is envisioned that such models will foster wider adoption of AI to enable reliable network performance and services.

The remainder of the paper is structured as follows: Section II presents the two datasets utilized for pretraining the foundational model, and for the spectrum forecasting and segmentation tasks. Section III outlines the architecture and algorithm of the self-supervised foundational model. Section IV presents numerical experiments conducted to evaluate the proposed methodology. Finally, Section V concludes the paper.

II. Testbed and Datasets

We leverage two datasets in this paper. The first is a Real-time Radio Dataset (RRD), captured in real-time using a software-defined radio (SDR) test-bed developed with PlutoSDRs. The second dataset simulates 5G New Radio (NR) and LTE transmissions in neighboring bands. This is called the Segmentation Dataset (SD). In both datasets, our primary emphasis is on processing spectrogram data rather than IQ samples, thus a significant portion of preprocessing and data preparation revolves around spectrogram computation.

A. Real-time Radio Dataset (RRD)

The RRD dataset consists of time-domain recordings of IQ samples, which represent both the in-phase (I) and quadrature (Q) components of the RF signal. Each recording corresponds to a distinct center frequency, sampling frequency, and running for a specific duration. The center frequency spans from 2.4 to 2.65 GHz, with the sampling frequency varying between 10 MHz and 60 MHz. The time duration typically averages around 100 ms. The data was collected in downtown Toronto, Canada. We utilize the dataset for foundational model pretraining and spectrum forecasting. There are 240 recordings in total corresponding to approximately 24 seconds of RF activity.

Spectrogram Computation. The spectrogram of each IQ recording is then computed as follows.

1) Divide the recording into non-overlapping 2 ms slices.
2) Compute the spectrogram for each 2 ms slice.
3) Convert each spectrogram from the linear to the log scale.

The parameters used to generate the RRD dataset are summarized in Table I.

B. NR-LTE Segmentation Dataset (SD)

The process of creating the SD dataset begins with the generation of 5G NR and LTE signals individually. Subsequently, these signals are transmitted through their respective wireless channels in adjacent bands. We employ the Matlab Communication Toolbox for signal generation, following the guidelines outlined in [11]. The parameters for generating 5G NR and LTE signals are presented in Tables II and III respectively.

TABLE I: RRD Dataset Generation Parameters

Parameters		Value
Spectrogram Parameters	FFT Size	1024
	Window Function	Hanning
	Window Size	512
	Hop Size	512
Slicing Parameters	Sentence Duration	10, 20 ms
	Sentence Shape	(256, 256)
	Token Shape	(256, 16)

TABLE II: 5G NR Signal Generation Parameters.

Parameter	Value
Bandwidth	$10, 15, \cdots, 50$ MHz
Sub-Carrier Spacing (SCS)	15, 30 KHz
Synch. Signal Block (SSB) Pattern	Cases A and B
Synch. Signal Block (SSB) Period	20 ms

TABLE III: LTE Signal Generation Parameters.

Parameter	Value
Bandwidth	5, 10, 15, 20 MHz
Reference Channel	R. {2, 4, 6, 8}
Duplex Mode	FDD

In more detail, the dataset creation process involves two main steps: signal generation and spectrogram computation.

Signal Generation

1) Randomly select a signal configuration from Table II for 5G NR and Table III for LTE. Generate 40 subframes of signal transmission, corresponding to 40 ms.
2) Apply the respective signal through its corresponding multipath fading channel. For 5G NR, the NR clustered delay line channel is utilized, while for LTE, the LTE fading channel is employed.
3) Perform frequency up-conversion on both signals to position them in neighboring bands, then mix the signals in time. Operate at a center frequency of 4 GHz with a sampling rate of 61.44 MHz. Randomly place the signals within the band-of-interest, ensuring no frequency overlap.

Spectrogram Computation

1) Compute the spectrogram for the resulting signal mixture.
2) Resize the spectrogram to the shape $(256, 256)$.
3) Create a label image of shape $(256, 256)$, assigning a value of 1 to pixels with NR signals, 2 to pixels with LTE signals, and 0 to pixels with noise.
4) Store the spectrogram and label image pair.

A sample spectrogram of this dataset is shown in Figure 1.

It is worth noting that the range of noise power is handled differently between the training and test sets. For the training

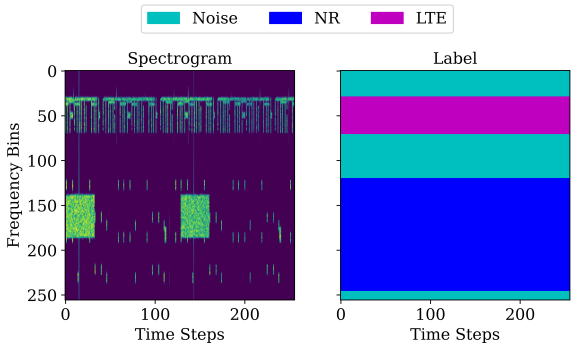


Fig. 1: A spectrogram and label pair for the segmentation task.

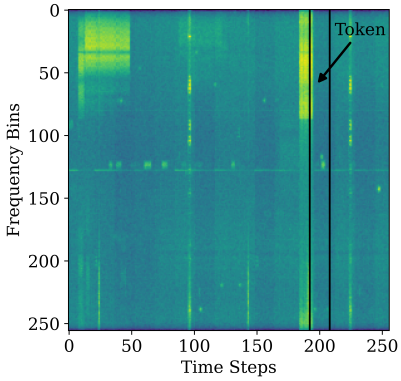


Fig. 2: A radio sentence created from the RRD dataset.

set, we utilize a normal distribution $\mathcal{N}(-70, 5)$ dBm, while for the test set, we employ a uniform distribution $\mathcal{U}(-90, -20)$ dBm. The reasoning behind this approach is as follows: in the training phase, we aim to prevent instances with high noise power from dominating the loss function, which could skew the training process. To achieve this, we undersample high noise instances by using a normal distribution. However, during testing, we want the model to be evaluated across all noise power levels equally. Therefore, we opt for a uniform distribution to ensure fair testing conditions.

III. Foundational Model for Spectrogram Learning

In this section we first present the methodology we propose to create the equivalent of *sentences* and *tokens* in the context of spectrograms. These radio sentences are then utilized by the proposed self-supervised masked spectrogram modelling approach which we present next. Here we utilize a convolutional LSTM (ConvLSTM) model introduced in [12]. This model is specifically crafted to capture crucial spatio-temporal features, aligning with our spectrogram learning needs. The convolutional component focuses on spatial properties, while the LSTM configuration handles temporal aspects. It accepts a sequence of two-dimensional spectrogram tokens as input and produces an output sequence of equal length. The details of the two downstream tasks that leverage this foundational ConvLSTM are then presented.

A. Creation of Radio Sentences and Tokens

(a) Randomly sample a sequence of successive spectrograms with a duration ranging from 10 to 20 ms.
(b) Concatenate the sequence of spectrograms along the time-axis.
(c) Resize the result to a shape of $(256, 256)$.
(d) Divide the result along the time-axis into a sequence of tokens with a shape of $(256, 16)$, allowing the sequence of tokens to be represented as a 3D array with a shape of $(16, 256, 16)$.
(e) Append the resulting sentence—a sequence of tokens—to the corpus, which will contain sentences of variable size.

A sample sentence is illustrated in Figure 2, where the token is also labelled. Next we discuss our proposed approach to use these radio sentences and tokens to pretrain and develop a radio foundation model.

B. Masked Spectrogram Modelling

We propose a technique, we refer to as masked spectrogram modeling (MSM) for pretraining the foundational model. This approach involves inputting a spectrogram into a deep learning model and masking a portion of it—typically 20%. Masking involves replacing the actual content of the spectrogram with white noise as shown in Figure 4. The model's objective is to reconstruct the original spectrogram from the masked version, effectively denoising it in the process. To achieve this, the model analyzes the surrounding context and infers what was likely in the masked positions. Throughout the learning process, the model is expected to develop an understanding of radio signals as represented by spectrograms, creating an internal representation that enables it to accurately recover the original spectrograms. A notable advantage of this approach is that it operates without the need for labels. Radio signals can be recorded and fed directly into the model pipeline, which then leverages them to refine its internal representation. The pretrained model can then be fine-tuned for any related downstream task, the procedure is illustrated in Figure 3 and a general algorithm is described in Algorithm 1.

TABLE IV: ConvLSTM Hyperparameters

Parameter	Value
Layers	5 ConvLSTM + 1 Conv3D
Number of kernels per layer	64
Kernel size	3
Activation function	ReLU

A ConvLSTM model is used for pretraining, utilizing the hyperparameters listed in Table IV. The mean-square error is

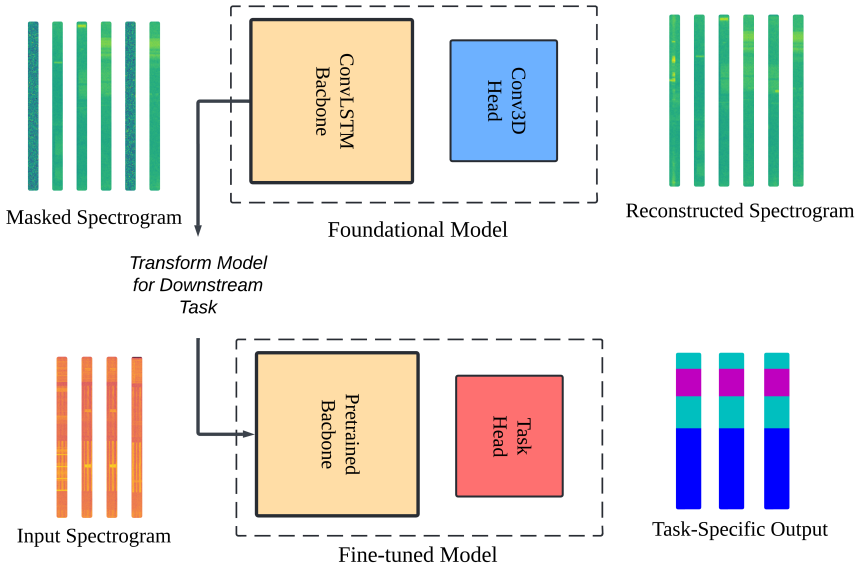


Fig. 3: Illustration of the proposed methodology for MSM pretraining and downstream task fine-tuning.

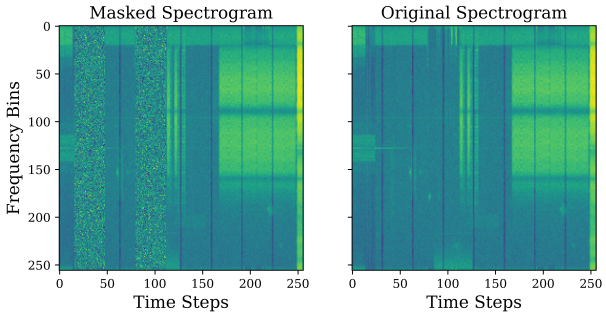


Fig. 4: Masked and original spectrogram pair.

used as the loss function, computed only for the *masked tokens*. The loss function $\mathcal{L}_{\text{MSM}}$ of MSM task can be written as:

$$\mathcal{L}_{\text{MSM}} = \sum_{n=1}^{N} \sum_{t=1}^{T} \left\| \text{vec}\left(\mathbf{W}_t^{(n)}\right) - \text{vec}\left(\hat{\mathbf{W}}_t^{(n)}\right) \right\|_2^2 \mathbb{I}_{\text{masked}}(n,t) \quad (1)$$

where N is the batch size, $\mathbf{W}_t^{(i)} \in \mathbb{R}^{256\times16}$ is the t^{th} input token from sample n, $\hat{\mathbf{W}}_t^{(i)} \in \mathbb{R}^{256\times16}$ is the t^{th} predicted token for sample n, T is the number of input tokens, vec denotes the vectorization operation, $\|\cdot\|_2$ is the L_2 norm and $\mathbb{I}_{\text{masked}}(i,t)$ is an indicator function that outputs 1 if token t from sample n was masked and 0 otherwise. The resulting self-supervised pre-trained model serves as our radio foundational model and is used for the following two downstream radio tasks.

C. Spectrum Forecasting

In this task, the model takes a sequence of tokens $\left\{\mathbf{W}_1^{(n)}, \cdots, \mathbf{W}_T^{(n)}\right\}$ as input and aims to predict the next token $\hat{\mathbf{W}}_{T+1}^{(n)}$. Training involves minimizing the mean-square-error loss between the predicted token and the actual next token for a batch of inputs $\left(\left\{\mathbf{W}_1^{(n)}, \cdots, \mathbf{W}_T^{(n)}\right\}_{n=1}^{N}, \left\{\mathbf{W}_{T+1}^{(n)}\right\}_{n=1}^{N}\right)$, where n is the sample index and N represents the batch size. This loss function, denoted as $\mathcal{L}_{\text{SF}}$, is formulated as follows:

$$\mathcal{L}_{\text{SF}} = \sum_{n=1}^{N} \left\| \text{vec}\left(\mathbf{W}_{T+1}^{(n)}\right) - \text{vec}\left(\hat{\mathbf{W}}_{T+1}^{(n)}\right) \right\|_2^2 \quad (2)$$

The model architecture described in Table IV is used, yet with two notable modifications: first, only the initial token of the output Conv3D layer is considered, disregarding the remaining outputs. Second, the backbone of the model, consisting of the 5 ConvLSTM layers, is frozen, and its weights are initialized with those obtained from the MSM task. Consequently, the features learned during the MSM task are utilized, while only the final layer undergoes fine-tuning for spectrum forecasting purposes.

D. Spectrogram Segmentation

In this task, the model processes a spectrogram of size $(256, 256)$, which is then tokenized into 16 tokens of shape $(256, 16)$. Consequently, the input comprises a sequence of tokens $\left\{\mathbf{W}_1^{(n)}, \cdots, \mathbf{W}_{16}^{(n)}\right\}$ while the output is a segmented image $\mathbf{Y}^{(n)} \in \{0,1\}^{(256,\ 256,\ 3)}$ that is one-hot encoded and the model prediction is denoted as $\hat{\mathbf{Y}}^{(n)} \in [0,1]^{(256,\ 256,\ 3)}$. For a batch of size N, we utilize the cross entropy loss written as:

$$\mathcal{L}_{\text{SG}} = -\sum_{i}\sum_{j}\sum_{n=1}^{N} \mathbf{Y}_{ij}^{(n)} \cdot \log\left(\hat{\mathbf{Y}}_{ij}^{(n)}\right) \quad (3)$$

Similar to spectrum forecasting, the backbone consists of 5 ConvLSTM layers. Subsequently, the backbone's output is concatenated to form a shape of $(256, 256)$, serving as input to

Algorithm 1: Self-Supervised Pre-training and Downstream Fine-Tuning Framework

Pretraining subalgorithm
Input : *initial_model*, *IQ_recordings*
Output: *pretrained_model*
pretrained_model ← *initial_model*
Convert *IQ_recordings* to radio sentence representation using the procedure described in Section II
for *sentence* in all *radio_sentences* **do**
  masked_sentence ← RANDOM_MASK(*sentence*)
  predicted_sentence ← *pretrained_model*·FORWARD(*masked_sentence*)
  loss ← MSE_{MSM}(*masked_sentence*, *sentence*) using eq. (1)
  pretrained_model ← UPDATE(*pretrained_model*, *loss*)
end
end subalgorithm

Fine-tuning subalgorithm
Input : *pretrained_model*, *downstream_dataset*
Output: *finetuned_model*
finetuned_model ← *pretrained_model*
preprocessed_dataset ← PREPROCESS(*downstream_dataset*) using the same transformations utilized for the *pretrained_model*
for every (*input, target*) in *preprocessed_dataset* **do**
  prediction ← *finetuned_model*·FORWARD(*input*)
  loss ← LOSS_FN(*prediction*, *target*)
  finetuned_model ← UPDATE(*pretrained_model*, *loss*)
end
end subalgorithm

a two-layer Conv2D classifier. The backbone's weights remain fixed, the classifier is fine-tuned for the segmentation task.

IV. Results and Discussion

This section evaluates the proposed self-supervised radio pretraining methodology by comparing the performance of the resulting foundational model when fine-tuned on two downstream tasks—spectrum forecasting and segmentation—to a baseline. The baseline model shares the same architecture but is trained from scratch on identical data.

A. Downstream Task-1: Spectrum Forecasting

Data and Model Training: We partition the RRD dataset, allocating 50% for pretraining and reserving the remaining 50% for forecasting.

We fine-tune the pretrained model using the remaining 50% of the RDD dataset, which we further split into training and test sets with an 80% to 20% ratio.

Evaluation Metric: To evaluate the model's forecasting performance, relying solely on visual comparison between the target and predicted spectrograms will not suffice. Therefore, we adopt a more robust metric by transforming each spectrogram into a resource grid composed of resource blocks (RBs) with predefined time and frequency resolutions. This process involves dividing the spectrogram into these blocks and computing the mean value for each. Subsequently, a threshold is applied to the resulting grid, rendering it binary—where a value of 1 denotes an occupied block and 0 denotes a vacant one. Figure 5 depicts the outcome of this transformation. The threshold δ is empirically determined as:

$$\delta = \mu + 0.5 \times \sigma \tag{4}$$

where μ and σ represent the mean and standard deviation of the spectrogram, respectively.

Our primary evaluation metric focuses on the model's capacity to correctly predict the occupancy of a resource block when it is indeed occupied. Predicting vacancy is straightforward, given its prevalence as the dominant class. From the perspective of opportunistic spectrum access, it is crucial to consistently detect occupied blocks to mitigate potential collisions.

This metric is depicted in Figure 6 for predictions extending 4 tokens into the future, across various time and frequency resolutions for the resource block, with (b) representing the baseline. The specialized baseline outperforms the tuned foundational model, though by a small margin.

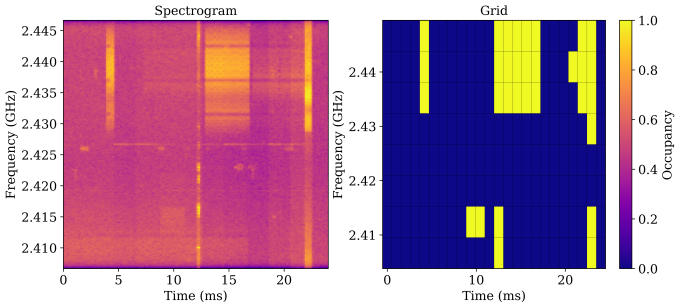


Fig. 5: Illustration of a spectrogram and its corresponding resource grid for a block size of (1 ms, 5 MHz).

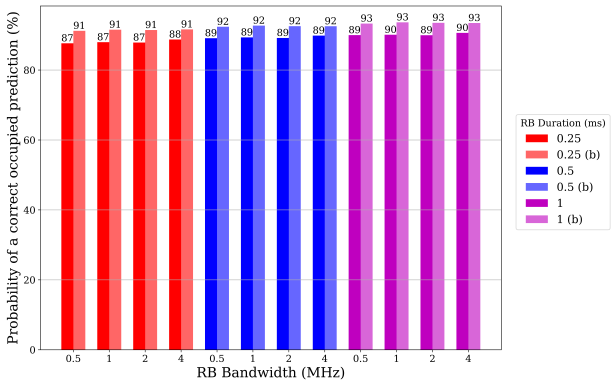
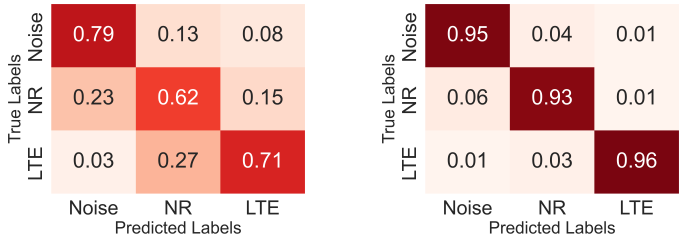
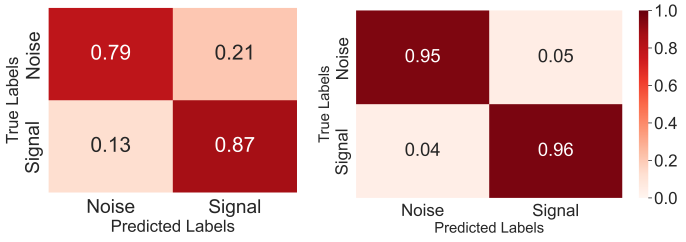


Fig. 6: Probability of correct occupied predictions. The solid lines are the foundational tuned model and (b) is the baseline.



Tuned Foundational Model Baseline Model

Fig. 7: Segmentation performance using confusion matrices.



Tuned Foundational Model Baseline Model

Fig. 8: Binary segmentation performance using confusion matrices.

B. Downstream Task-2: NR-LTE Segmentation

For the segmentation task, we utilize the SD dataset to fine-tune the foundational model. The main challenge here lies in the distinct nature of the spectrograms within this dataset compared to those used during the self-supervised pretraining. Consequently, the learned features may not generalize as effectively to segmentation as they do to forecasting. In addition, segmentation is a classification task and the model was pre-trained on regression only. Our objective is to examine the extent to which the learned representations of the foundational model can generalize under such distinct data distributions and across task types.

We evaluate the model's performance using confusion matrices for the three classes: Noise, NR, and LTE, which quantify the model's prediction accuracy for each class. Figure 7 presents the confusion matrices for both the baseline and fine-tuned models. Notably, the fine-tuned model struggles with distinguishing NR signals, while the baseline model demonstrates strong performance across all classes. This suggests that the features provided by the pretrained backbone are not sufficiently discriminative to differentiate NR signals from other classes, though they perform adequately in separating signals (NR or LTE) from noise. A more complex head and fine-tuning training process may also be needed.

To further illustrate this, we simplify the task to binary segmentation, merging NR and LTE into a single *signal* class. The resulting confusion matrices are shown in Figure 8. Here, while the baseline model still outperforms the fine-tuned model, the correct detection of signals is higher. We attribute this to differences in data distribution between the pretraining and SD datasets. Pretraining on a larger and more diverse dataset may help bridge this gap. Further research by the community will be needed in these directions to build large-scale foundational radio models.

V. Conclusion

In this paper, we introduced a self-supervised radio pretraining approach, MSM, to build a foundational model for spectrogram learning. Drawing inspiration from the success of foundational DL models in various domains, the goal was to learn features that could generalize to related downstream tasks. We demonstrated that by fine-tuning the developed MSM model for two downstream tasks: spectrum forecasting and segmentation. Our results show that the fine-tuned models exhibited competitive performance compared to baselines trained from scratch, while requiring much less training time to converge. We believe that extending the proposed MSM approach to larger models and utilizing large-scale, diverse datasets for pretraining has the potential to develop robust radio foundational models that yield competitive performance across various spectrogram learning 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] 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.

[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.

[4] P. Goyal, M. Caron, B. Lefaudeux, 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.

[5] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[6] 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. 10 760–10 772, 2018.

[7] 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.

[8] 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.

[9] 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.

[10] D. Chong, H. Wang, P. Zhou, and Q. Zeng, "Masked spectrogram prediction for self-supervised audio pre-training," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[11] "Spectrum sensing with deep learning to identify 5g and lte signals." [Online]. Available: https://www.mathworks.com/help/comm/ug/spectrum-sensing-with-deep-learning-to-identify-5g-and-lte-signals.html

[12] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc., 2015.