

GREAT AI for Foundation Models in 6G

Generative Radio Embeddings for Accelerated and Trustworthy AI

Ericsson Ottawa R&D Site

Hatem Abou-Zeid, PhD
Electrical & Software Engineering Department
University of Calgary, Canada

October 2024



GREAT AI for Foundation Models in 6G

Generative Radio Embeddings



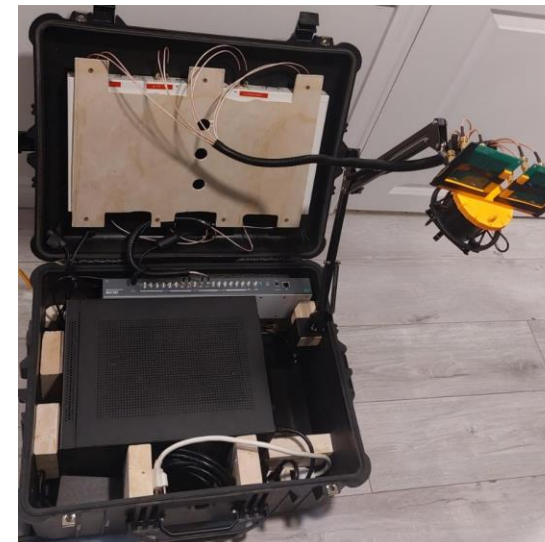
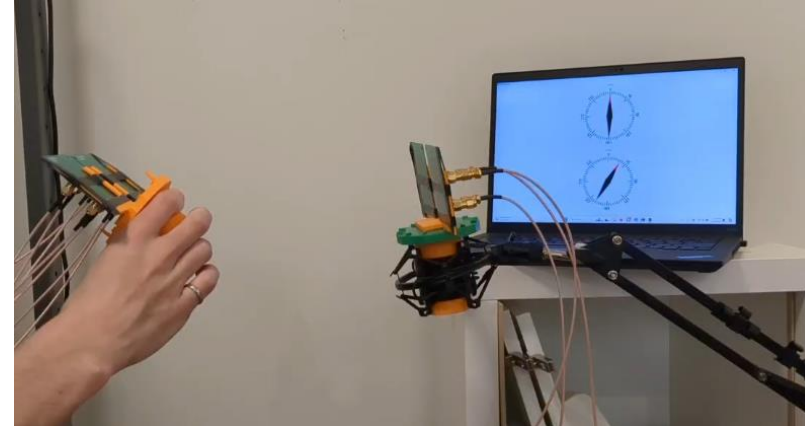
Accelerated & Trustworthy AI



Foundation Models in 6G

Challenges are we trying to solve

Models do not
generalize/adapt



When the Software
Defined Radio (SDR)
got hot the Angle of
Arrival model
accuracy went
down!

Challenges are we trying to solve

Models do not
generalize/adapt

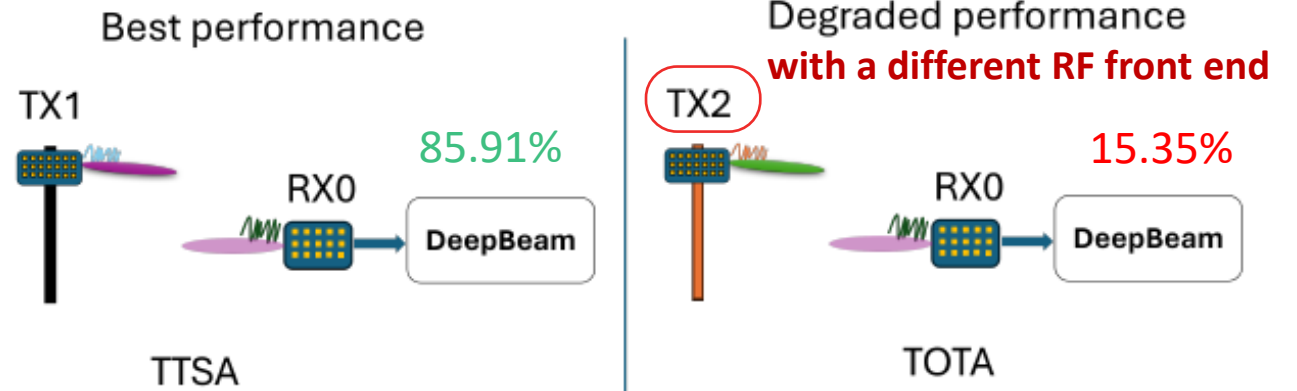
DeepBeam: Deep Waveform Learning for Coordination-Free Beam Management in mmWave Networks

Michele Polese, Francesco Restuccia, and Tommaso Melodia

Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, United States

ABSTRACT

Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and



Accelerated & Trustworthy AI will help us solve..

Models do not
generalize/adapt

Models are too
slow, high memory
and energy



Models need lots of
data that is labeled

Models are not
interpretable and
safe

Challenges are we trying to solve

Models do not
generalize/adapt

Models need lots of
data that is labeled

The need for so many very
specific models

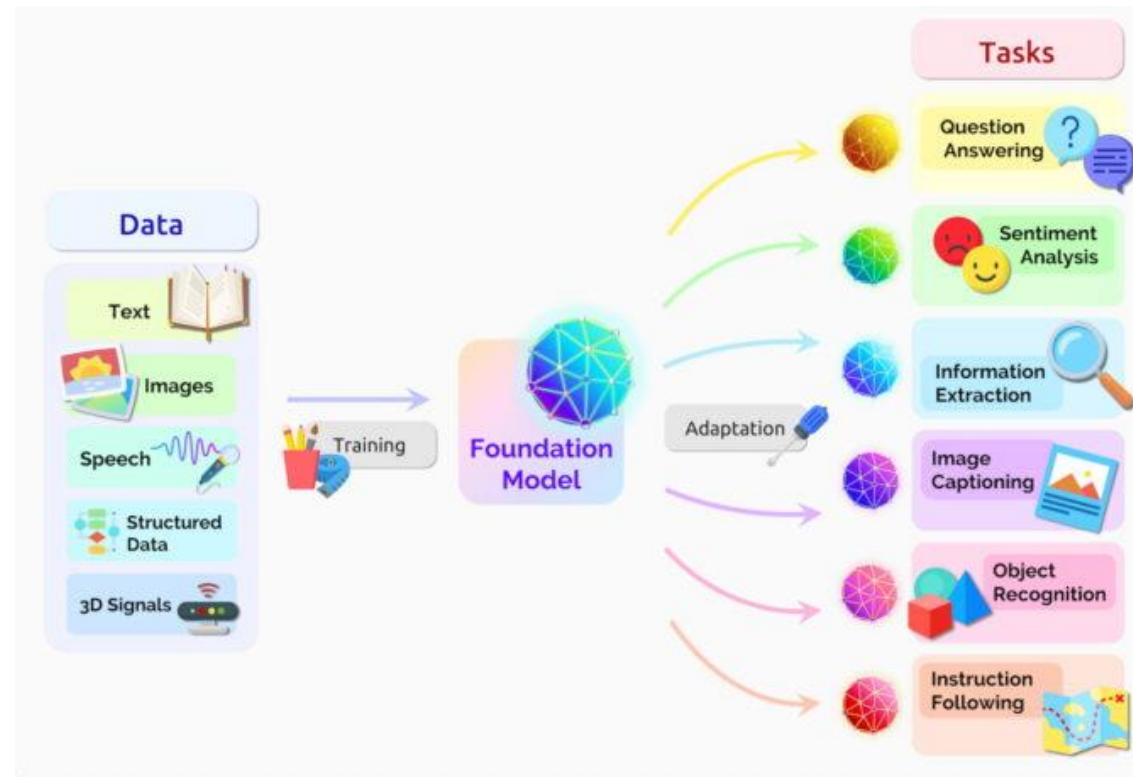
Models are too
slow, high memory
and energy

Models are not
interpretable and
safe

Foundation Models in 6G

Foundation Models (FM)

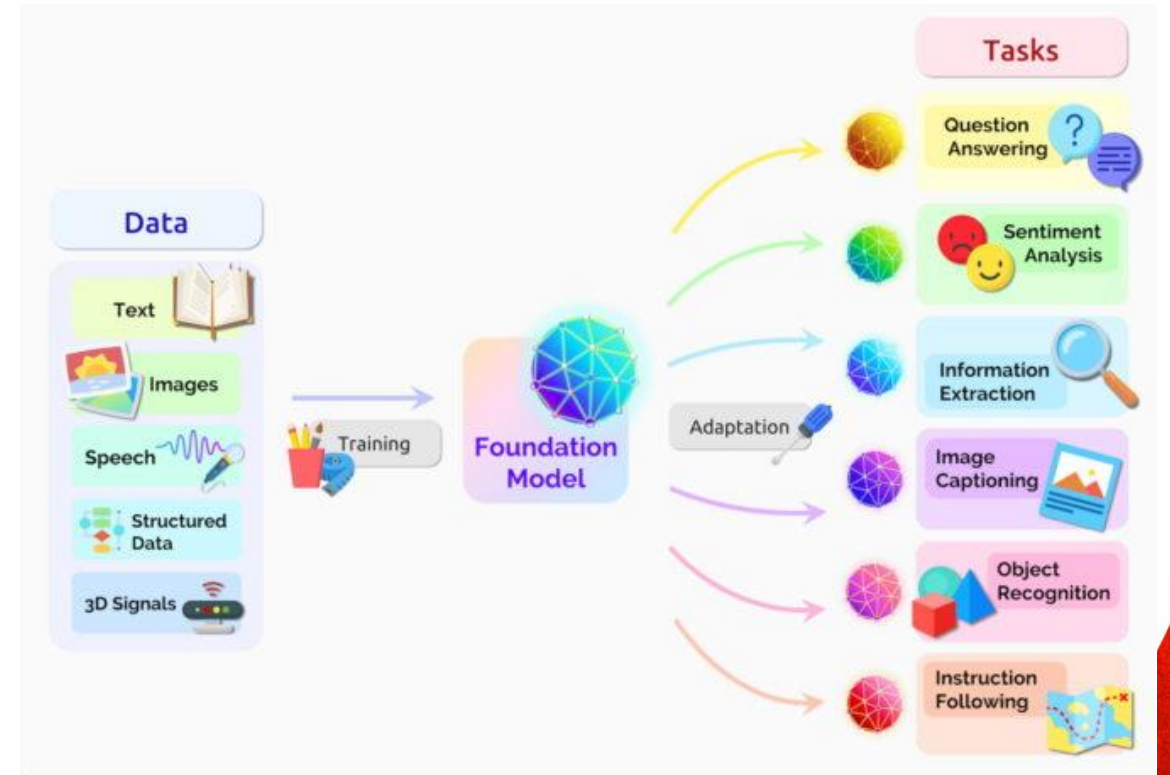
- Foundation models are a large, pre-trained machine learning models that serve as the basis/foundation for a wide range of downstream tasks.
- ChatGPT is built on a foundation model



<https://blogs.nvidia.com/blog/what-are-foundation-models/>

Foundation Models (FM) – Two key properties

1. Generalization across Tasks:
 - Fine-tuning enables diverse “downstream” tasks with relatively little additional training.
2. Do not require large labeled data for pre-training
 - Use methods such as self-supervised learning

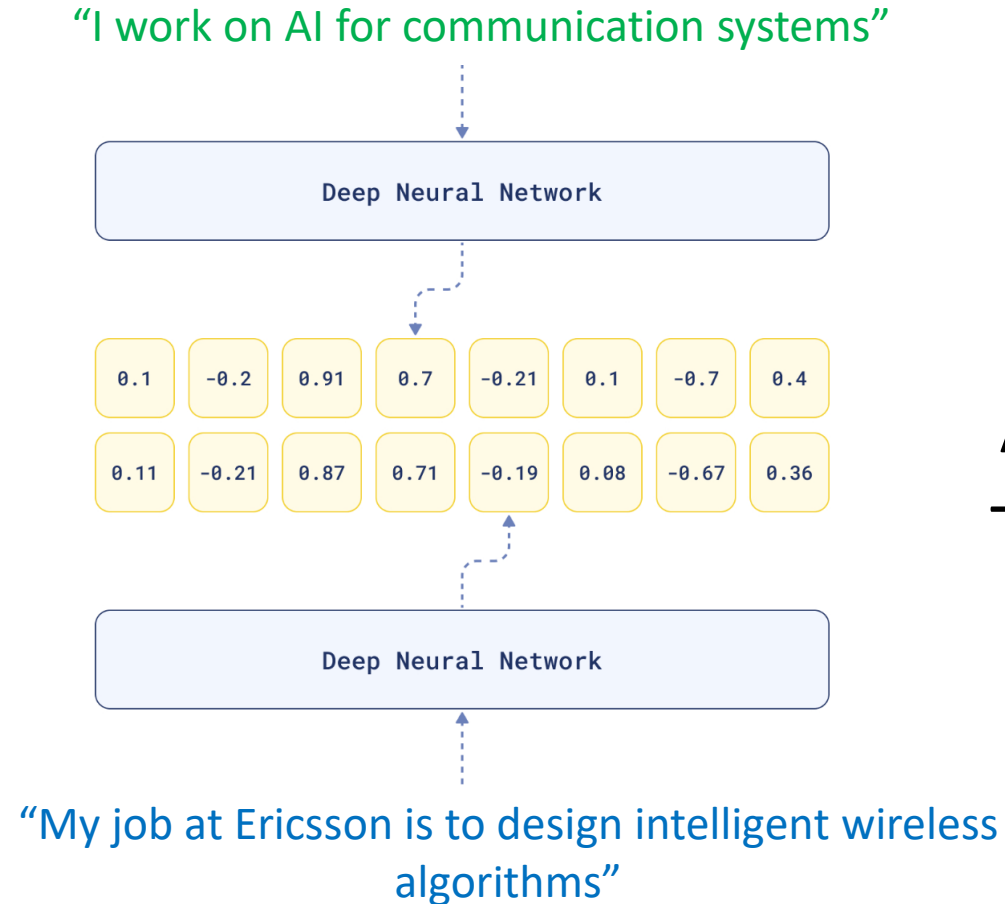


<https://blogs.nvidia.com/blog/what-are-foundation-models/>

GREAT AI: Background & Definitions

Embeddings

- In AI, **embeddings** are low-dimensional, dense vectors that represent high-dimensional data (words, images) in a more compact, mathematical form.
 - Embeddings can capture semantics, context.
 - Deep learning models generate *image embeddings*, and *text embeddings*.



Generative
Radio
Embeddings
for
Accelerated &
Trustworthy AI

GREAT AI: Background & Definitions

- Why create **embeddings**?
 - **Good Representations:** Capture semantics and context in numbers that can be used by machine learning models.
 - **Efficiency:** Embeddings are much smaller and more computationally efficient than high-dimensional raw data.
 - **Transferable Representations:** Good embeddings can be used across different tasks, e.g. word embeddings reused for sentiment analysis or translation.

Embeddings played a crucial role in
enabling foundation models

Generative
Radio
Embeddings
for
Accelerated &
Trustworthy AI

GREAT AI: Background & Definitions

- Generative Embeddings

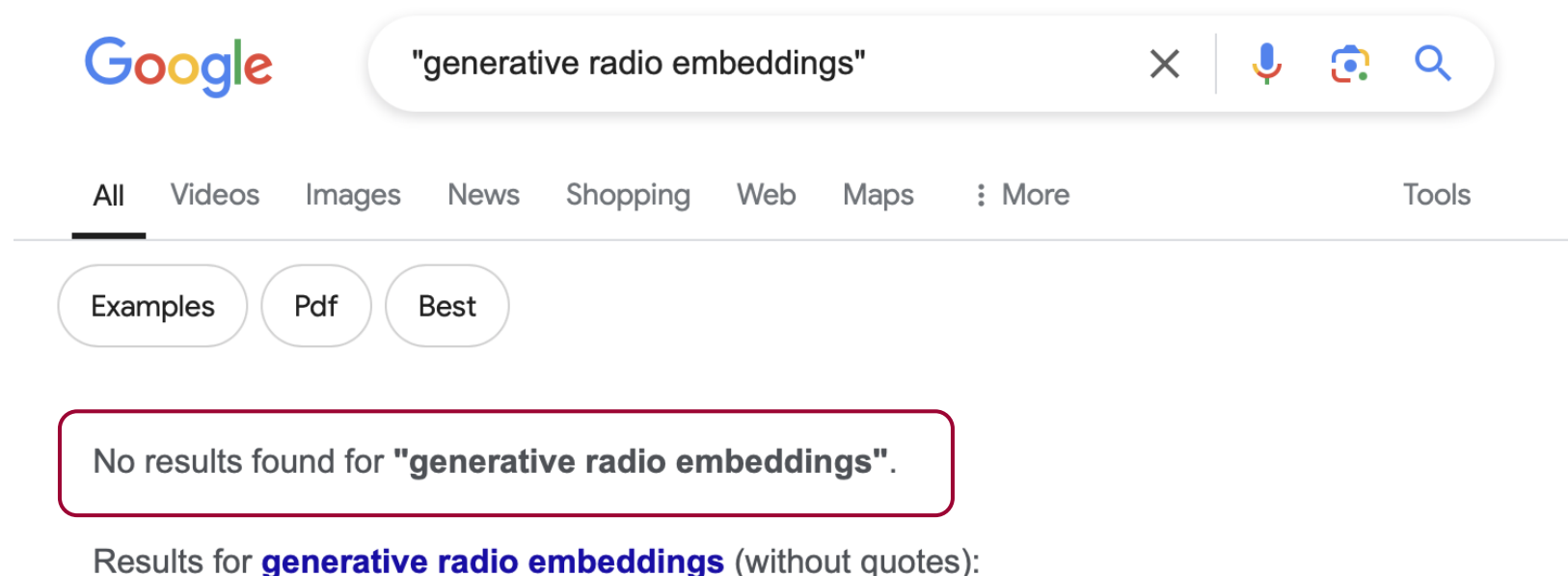
- **Embeddings** (representations of words, images, etc) that are produced using **generative models**.
- They not only represent the data but encode it in a way that allows the model to generate realistic new data points (text, images, etc.).
 - e.g. Variational Autoencoders (VAEs), GANs.
- Both non-generative and generative embeddings are useful in different ways

Generative
Radio
Embeddings
for
Accelerated &
Trustworthy AI

GREAT AI: Background & Definitions

- Generative Radio Embeddings

Let's see if Google knows!

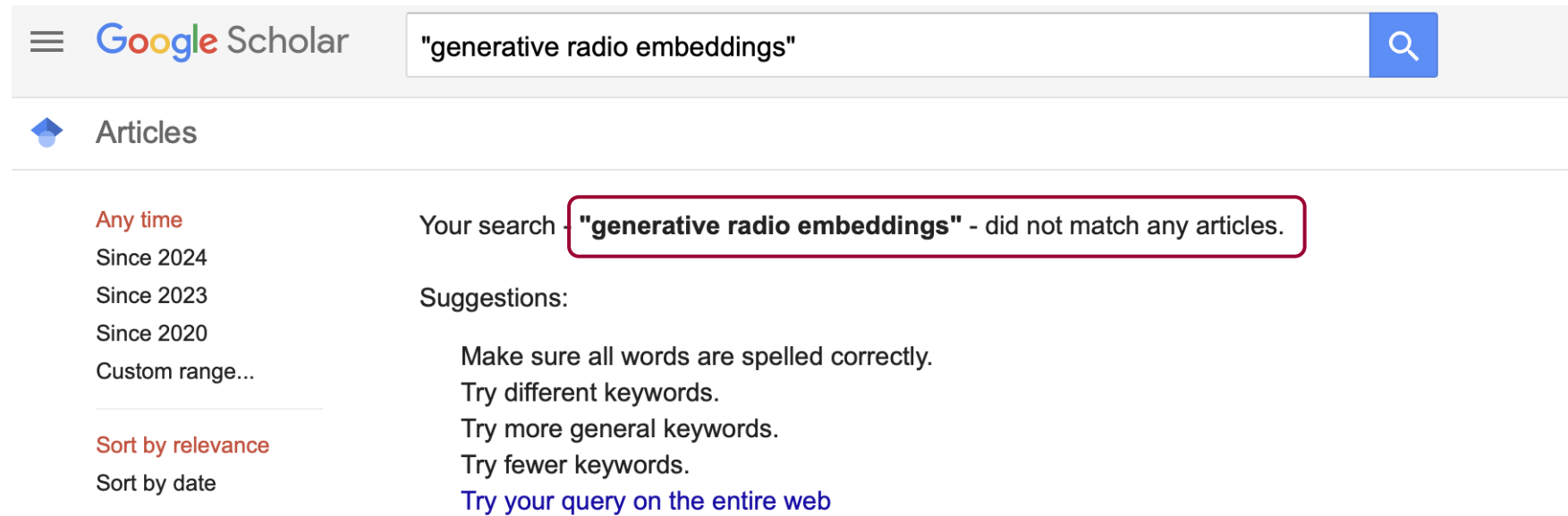


Generative
Radio
Embeddings
for
Accelerated &
Trustworthy AI

GREAT AI: Background & Definitions

- Generative Radio Embeddings

Let's see if GoogleScholar knows!



The screenshot shows the Google Scholar interface. The search bar contains the text "generative radio embeddings". Below the search bar, the results section is empty, displaying the message: "Your search 'generative radio embeddings' - did not match any articles." To the left of the search results, there are filters for "Articles" and a sidebar with options for "Any time", "Since 2024", "Since 2023", "Since 2020", "Custom range...", "Sort by relevance", and "Sort by date". Below the search results, there are suggestions: "Make sure all words are spelled correctly.", "Try different keywords.", "Try more general keywords.", "Try fewer keywords.", and "Try your query on the entire web".

Generative
Radio
Embeddings
for
Accelerated &
Trustworthy AI

GREAT AI: Background & Definitions

- Generative Radio Embeddings



ChatGPT

what are "generative radio embeddings"?



"Generative radio embeddings" is a concept that likely refers to embeddings created by generative models for use in radio frequency (RF) or wireless communications systems. This is an emerging area that could blend the power of generative AI with wireless network applications, such as signal processing, channel modeling, and efficient spectrum use.

Here's what the term might imply:

1. Radio Embeddings:

- Radio embeddings refer to low-dimensional representations of radio signals or wireless data, like channel states, modulation patterns, or other features of the RF spectrum.

Generative
Radio
Embeddings
for
Accelerated &
Trustworthy AI

GREAT AI: Background & Definitions

- Generative Radio Embeddings



ChatGPT

what are "generative radio embeddings"?

Generative
Radio
Embeddings

for

Accelerated &
Trustworthy AI

4. Benefits:

- Reduced Complexity: By creating embeddings that represent radio signals in a compact form, machine learning models can process complex wireless communication environments more efficiently.
- Data Augmentation: Generative radio embeddings can be used to generate synthetic signals that augment training datasets for machine learning applications in wireless networks.

+ other aspects of Accelerated AI such as models that are
fast to train and have smaller memory footprints

GREAT AI: Background & Definitions

- Generative Radio Embeddings



ChatGPT

what are "generative radio embeddings"?

Generative
Radio
Embeddings
for

Accelerated &
Trustworthy AI

4. Benefits:

- Adaptivity: Since generative models can create new embeddings based on learned distributions, they allow for adaptive communication systems that can predict and optimize responses to changing radio environments, like in 6G networks or intelligent wireless systems.

+ other aspects of Trustworthy AI such as
generalizability



GREAT AI: Background & Definitions

- Generative Radio Embeddings



ChatGPT

4. Benefits:

- Adaptivity: Since

Generative models like ChatGPT are able to “predict” and “speculate” what new terminology might mean (compared to Google which returned “no results”). Generative Radio Embeddings and Models should help networks navigate these situations!

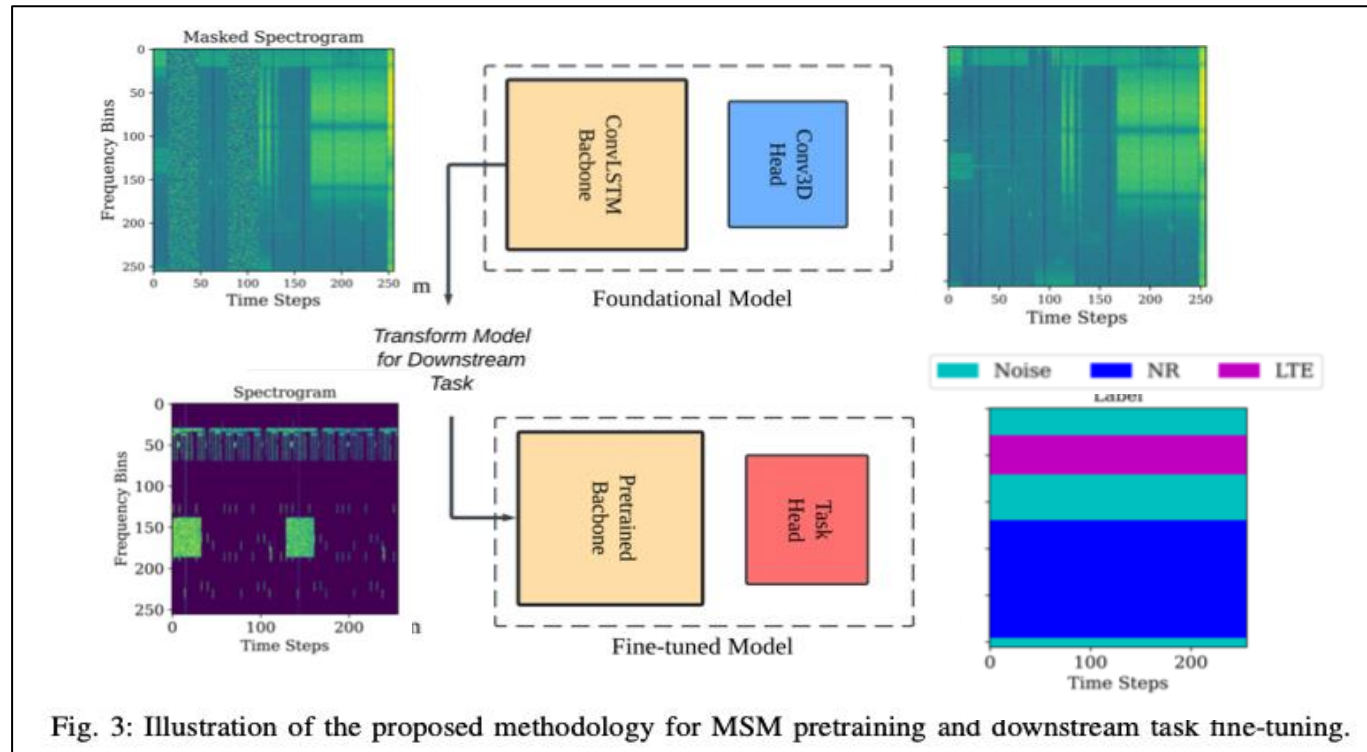
embeddings based on learned
communication systems that can predict and optimize
aspects, like in 6G networks or intelligent wireless systems.

Aspects of Trustworthy AI such as
generalizability

Generative
Radio
Embeddings
for
Accelerated &
Trustworthy AI

Foundation Models in the Wireless Context

- A Foundation Model for Spectrogram Learning
 - Can be used for different tasks like spectral occupancy prediction, classification of signals in the spectrum, various forms of sensing, channel estimates, etc
 - Should be able to generalize to different SNRs, bandwidth, frequency ranges, FDD/TDD etc



A. Aboufotouh, et. al. "Self-Supervised Radio Pre-training: Toward Foundation Models for Spectrogram Learning." IEEE GLOBECOM 2024

Talk Overview

- ✓ The notion of GREAT AI for Foundation Models in 6G

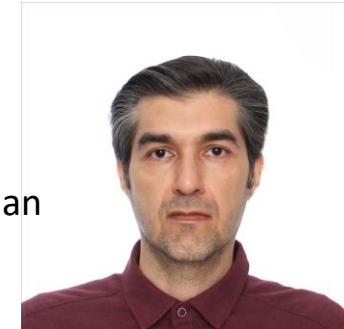
3 Topics

- Generalizable AI using Radio Embeddings
 - Beam prediction AI models that do not adapt well to the different RF fronts
 - A solution using Prototypical Networks
- Foundation Model using Generative Pre-training
 - (How) Can we develop Foundation Models for Spectrogram Learning?
- Accelerated AI (time permitting)
 - Models are too slow and too big
 - Solutions using early exits and model pruning
- Summary & Future Research
- Q&A, discussion (10 mins, lunch, and Fika!)

WAVES Lab Super Stars

This research would not be possible without this amazing group of researchers

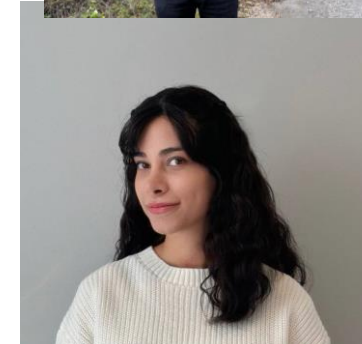
Elsayed Mohamed Ogechukwu Kanu Mohammadreza Behboodi Brian Irvine Omar Mashaal Fazal Khan
Mohamed Hallaq Ahmed Nagib Ahmed Aboufotouh Morvarid Lelanoor Sampreet Vaidya



Dr. Bakhshi
Fellow & Lab Manager



Dr. Rajaram
Industrial PDF



<https://www.hatem-abouzeid.com/waves-lab>

Generalizable AI using Radio Embeddings



Omar Mashaal, PhD Candidate
& Alberta Innovates Fellow



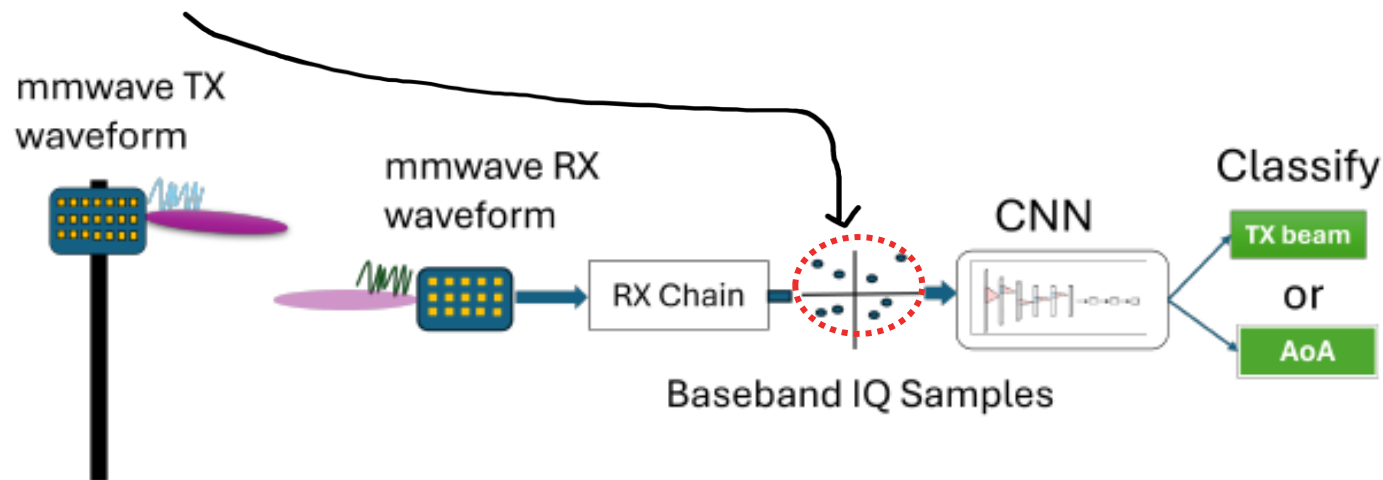
- A framework for waveform-level downlink for beam management.
- Infers transmit beams (TXB) without TX-RX coordination. Identifies which of the 24 beams the TX is using
- Processes I/Q samples directly

DeepBeam: Deep Waveform Learning for Coordination-Free Beam Management in mmWave Networks

Michele Polese, Francesco Restuccia, and Tommaso Melodia
Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, United States

ABSTRACT

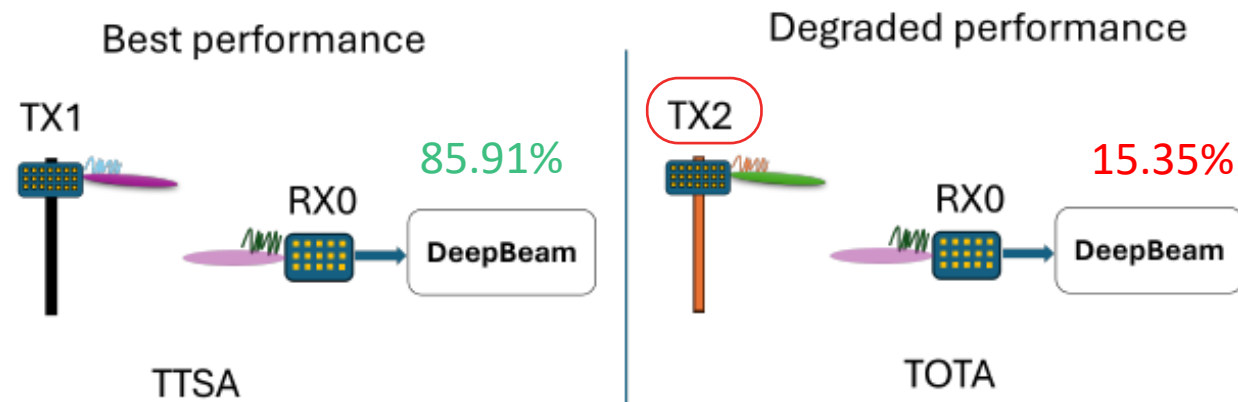
Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '21), July 26–29, 2021, Shanghai, China. ACM.



M. Polese, F. Restuccia, and T. Melodia. "DeepBeam: Deep waveform learning for coordination-free beam management in mmWave networks." *Proc. Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 2021.

The Generalization Challenge: Different RF Front-ends

- 4 different SiBeam 60 GHz frontends were used to evaluate generalization
- The accuracy dropped from 85.91% to 15.35% when the the model is deployed with a different RF front end RF front end than the one it was trained on



TTSA: Train and Test on the *Same Antenna* (TX1)

TOTA: Train on One and Test on *Another Antenna* (TX2)

What is non-Generalizable AI?

- The AI model is too specific to the data it was trained on
 - Does not generalize to out-of-distribution (OOD) data for the same task or to other related tasks

What is non-Generalizable AI?

Many Causes of Domain Shifts in Wireless

Table 1.1: Illustrative Examples of Causes of Domain Shift in Wireless Signals and Their Direct Impacts

Cause of Domain Shift	Direct Effect on Wireless Signals
Hardware Variations	Power level changes, IQ imbalance, and oscillator frequency drift.
Environmental Changes	Variations in signal propagation paths, shadowing, and multi-path effects.
User Mobility	Doppler shifts, variability in multi-path profiles, and temporal fading.
Interference	Increased noise floor, signal distortion, and potential overlap of frequency bands.

What is non-Generalizable AI?

Many Causes of Domain Shifts in Wireless

Table 1.1: Illustrative Examples of Causes of Domain Shift in Wireless Signals and Their Direct Impacts

Cause of Domain Shift	Direct Effect on Wireless Signals
Hardware Variations	Power level changes, IQ imbalance, and oscillator frequency drift.
Environmental Changes	Variations in signal propagation paths, shadowing, and multi-path effects.
User Mobility	Doppler shifts, variability in multi-path profiles, and temporal fading.
Interference	Increased noise floor, signal distortion, and potential overlap of frequency bands.

- 1- Difficulty in generalizing across distribution shifts.
- 2- Limitations in continuous learning across diverse scenarios.
- 3- Inability to rapidly adapt to unseen scenarios.

ProtoBeam: Generalization to RF Front-ends

ProtoBeam: Generalizing Deep Beam Prediction to Unseen Antennas using Prototypical Networks

Omar Mashaal[‡], Elsayed Mohammed[‡], Alec Digby*, Lorne Swersky*, Ashkan Eshaghbeigi*, Hatem Abou-Zeid[‡]

[‡]Department of Electrical and Software Engineering, University of Calgary, Canada

*Qoherent Inc., Toronto, Ontario, Canada

Abstract—Deep learning (DL) techniques have recently emerged to efficiently manage mmWave beam transmissions without requiring time consuming beam sweeping strategies. A fundamental challenge in these methods is their dependency on hardware-specific training data and their limited ability to generalize. Large drops in performance are reported in literature when DL models trained in one antenna environment are applied in another. This paper proposes the application of Prototypical Networks (PN) to address this challenge – and utilizes the DeepBeam real-world dataset [1] to validate the developed solutions. Prototypical Networks (PN) excel in extracting features to establish class-specific prototypes during the training, resulting in precise embeddings that encapsulate the defining features of the data. We demonstrate the effectiveness of PN to enable generalization of deep beam predictors across unseen antennas. Our approach, which integrates data normalization and prototype normalization with the PN, achieves an average beam classification accuracy of 74.11% when trained and tested on different antenna datasets. This is an improvement of 398% compared to baseline performances reported in literature that do not account for such domain shifts. To the best of our knowledge, this work represents the first demonstration of the value of Prototypical Networks for domain adaptation in wireless networks, providing a foundation for future research in this area.

Beam Management, Domain Adaptation, Prototypical network, angle-of-arrival, mm-wave.

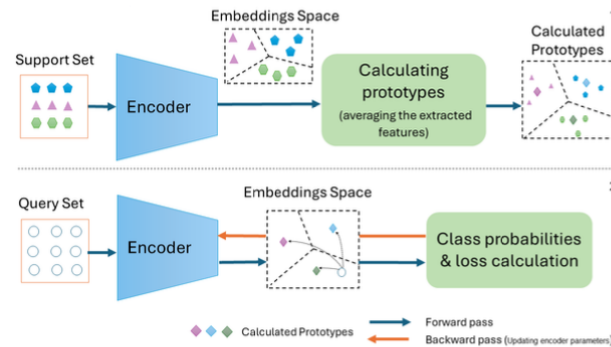
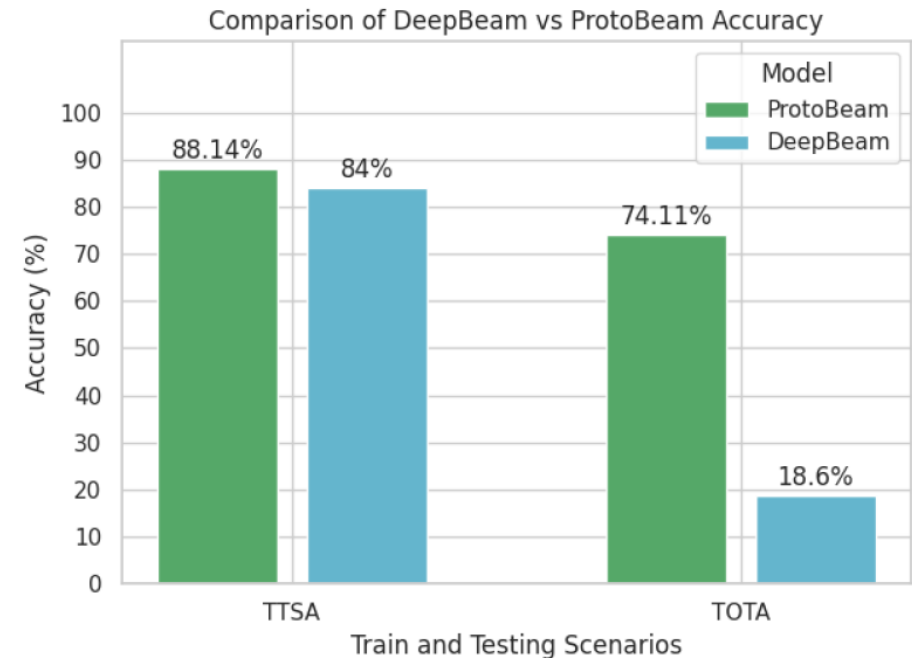


Fig. 1: Prototypical Network Architecture and Training [4].

transmission. Deep learning (DL) has emerged as a powerful tool for refining beam management strategies, enabling dynamic beam prediction and alignment [1]–[3]. DeepBeam [1] leverages deep learning to optimize beam selection using I/Q data. This framework is designed to infer the Angle of Arrival (AoA) and identify the beam used by the trans-



Prototypical Networks & Few-shot Learning

Prototypical networks for few-shot learning

[J Snell](#), [K Swersky](#), [R Zemel](#) - *Advances in neural ...*, 2017 - *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
... We propose **Prototypical Networks** for the problem of **few-shot** classification, given only a small number of examples of each new class. **Prototypical Networks** learn a metric space in which classification can be performed by computing distances to prototype representations of each class. Compared to recent approaches for few-shot learning, they reflect a simpler inductive bias that is beneficial in this limited-data regime, and achieve excellent results. We provide an analysis showing that some simple design decisions can yield substantial improvements over recent approaches involving complicated architectural choices and meta-learning. We further extend Prototypical Networks to zero-shot learning and achieve state-of-the-art results on the CU-Birds dataset.

☆ Save Cite Cited by 9453 Related articles All 12 versions

- Prototypical networks are a type of few-shot learning model
- Few-shot learning is to learn from a limited number of labeled examples (or "shots") per class.

Prototypical Networks for Few-shot Learning

Jake Snell
University of Toronto*
Vector Institute

Kevin Swersky
Twitter

Richard Zemel
University of Toronto
Vector Institute
Canadian Institute for Advanced Research

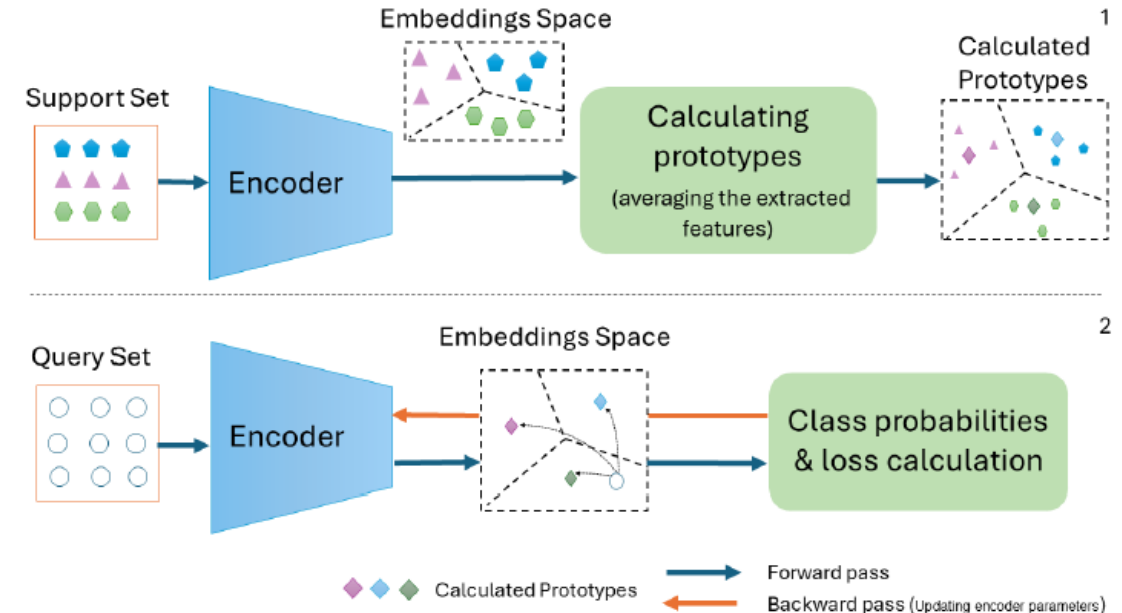
Abstract

We propose *Prototypical Networks* for the problem of few-shot classification, where a classifier must generalize to new classes not seen in the training set, given only a small number of examples of each new class. Prototypical Networks learn a metric space in which classification can be performed by computing distances to prototype representations of each class. Compared to recent approaches for few-shot learning, they reflect a simpler inductive bias that is beneficial in this limited-data regime, and achieve excellent results. We provide an analysis showing that some simple design decisions can yield substantial improvements over recent approaches involving complicated architectural choices and meta-learning. We further extend Prototypical Networks to zero-shot learning and achieve state-of-the-art results on the CU-Birds dataset.

Snell, J., Swersky, K., and Zemel, R., "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

How Prototypical Networks Work

- **Embedding Function:** Each input (e.g., an image) is passed through an embedding function (like a neural network) to produce a feature representation.
- **Prototypes:** For each class, the model calculates the prototype by averaging the feature representations of the few labeled examples provided for that class.
- **Distance Metric:** New (unlabeled) samples are classified by measuring their distance (typically using Euclidean distance) to the prototypes in the feature space. The sample is assigned to the class whose prototype is closest.



Snell, J., Swersky, K., and Zemel, R., "Prototypical networks for few- shot learning," Advances in neural information processing systems, vol. 30, 2017.

Prototypical Networks

- Prototypical networks are effective for tasks with scarce labeled data, making them ideal for applications where data collection is challenging or expensive.

Many successful applications

- **Medical Diagnosis:** In medical imaging, prototypical networks help classify diseases when there are only a few examples of certain conditions available.
- **Speech Recognition:** Used for recognizing new speech patterns or accents with limited training data.

Snell, J., Swersky, K., and Zemel, R., "Prototypical networks for few- shot learning," Advances in neural information processing systems, vol. 30, 2017.

Can PNs learn good Radio Embeddings?

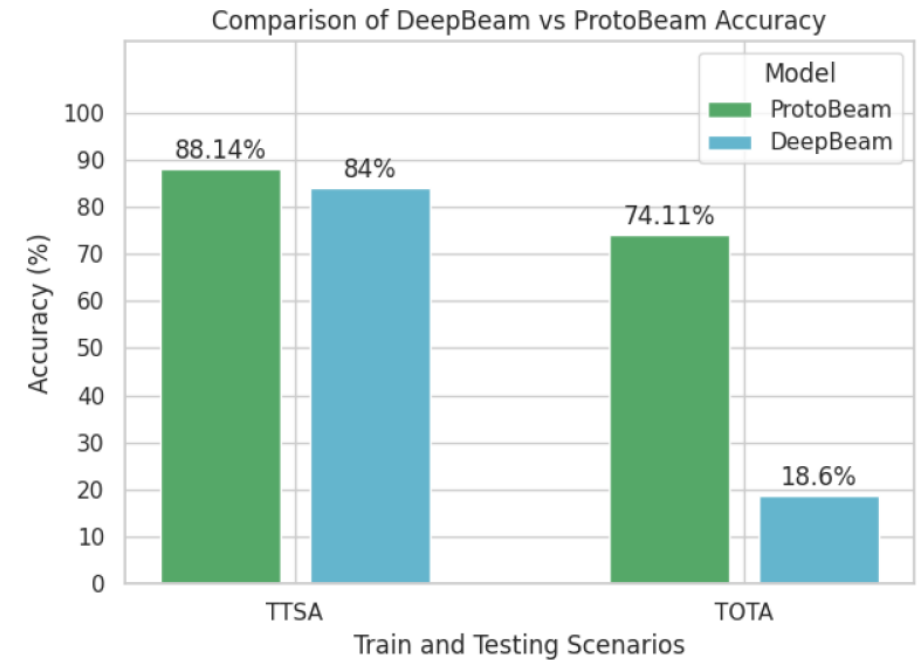
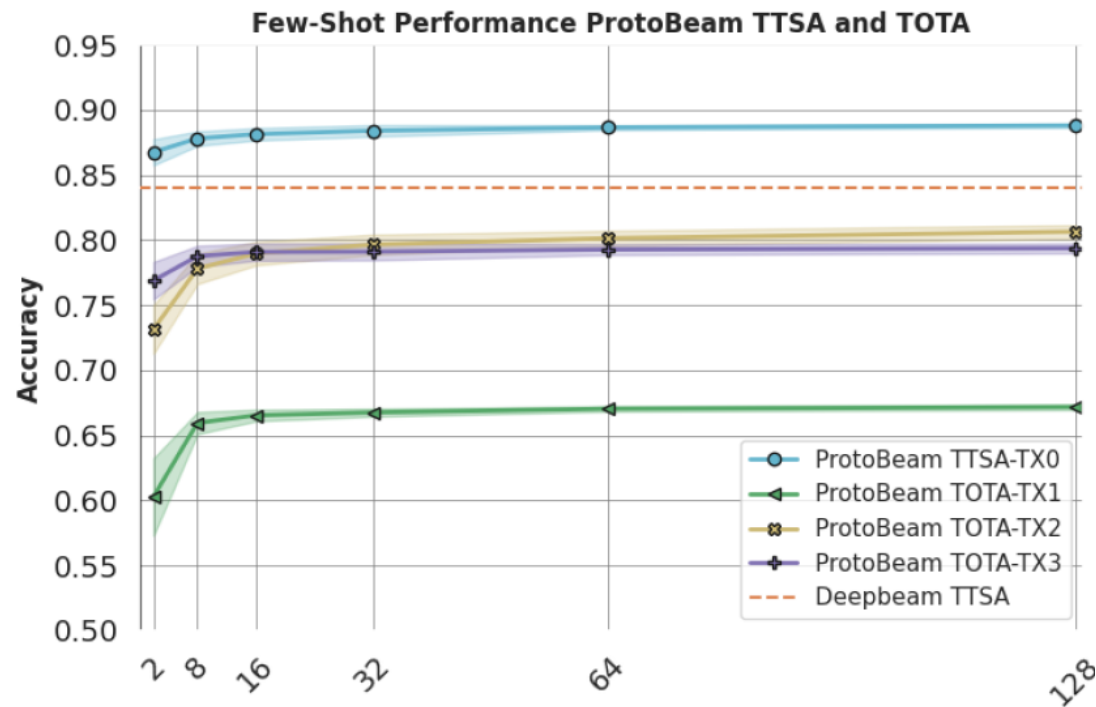
PNs were proposed for few-shot learning in images – can we use :

- their property of learning general representations to create *radio embeddings* that are robust to wireless domain shifts?
- and the few-shot property to quickly adapt to these domain shifts?

**GOAL: Learn generalizable radio embeddings/
representations that can be quickly adapted with
a few shots of calibration if the domain changes.**

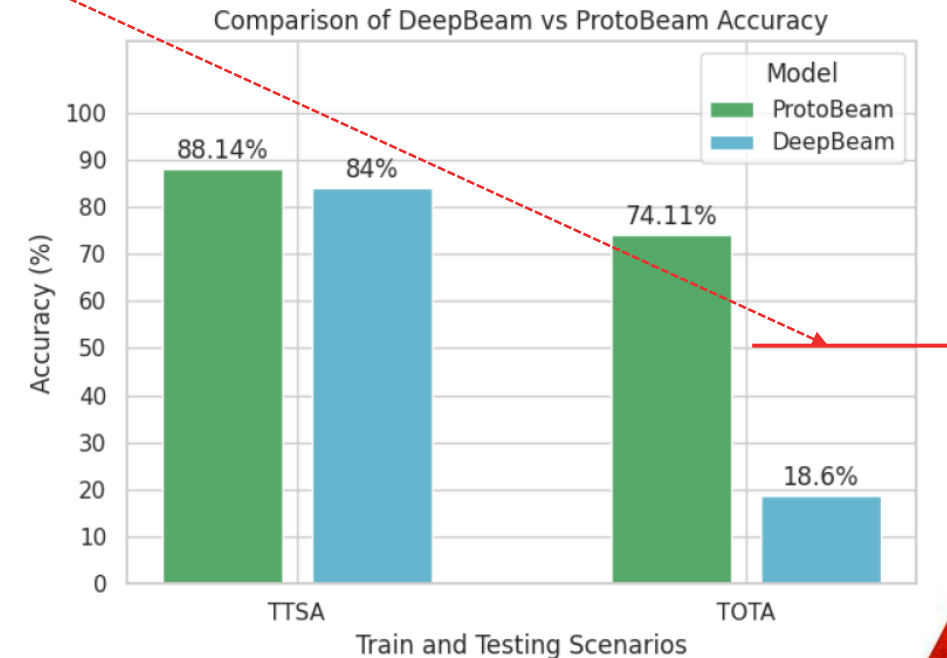
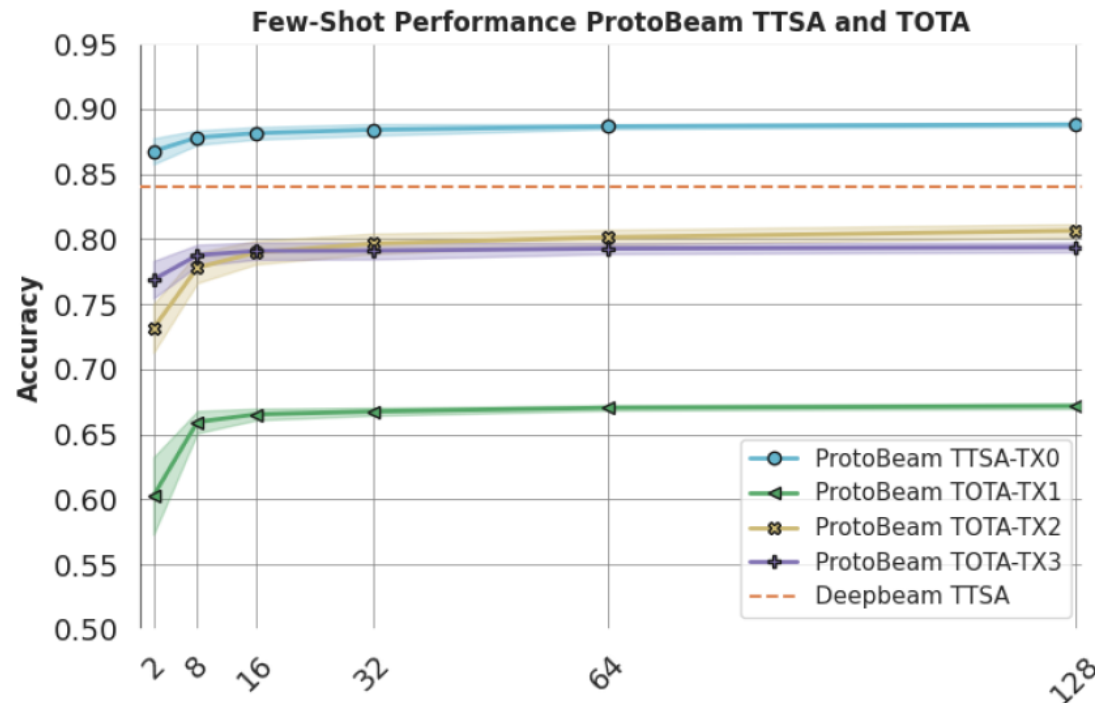
ProtoBeam: *Generalization to Unseen Antennas*

- ProtoBeam improved the accuracy significantly in the TOTA scenario with only 2-8 samples/beam of labeled data from the different RF front-end



ProtoBeam: *Generalization to Unseen Antennas*

- ProtoBeam performed better than a mixed training setting with all the RF front-end data (50%)
- ProtoBeam also improved the baseline TTSA accuracy



ProtoBeam Training

Algorithm 1 Proposed ProtoBeam Training Algorithm

Inputs: Training set $\mathcal{D} = \{(x_1, y_1), \dots, (x_{n_b}, y_{n_b})\}$, where x_i is the I/Q sample and $y_i \in \{1, \dots, B\}$ is the target beam. \mathcal{D}_b denotes the subset of \mathcal{D} containing all elements (x_i, y_i) for target beam b .

Parameters: n_b is the number of baseband I/Q samples. B is the number of target beams, $N_B \leq B$ is the number of target beams per episode.

n_S is the number of I/Q support examples per target beam.

n_Q is the number of I/Q query examples per target beam.

$\text{RandSample}(\mathcal{S}, N)$ denotes a set of N elements chosen uniformly at random from set \mathcal{S} , without replacement.

Output: Updated model parameters after backpropagation.

procedure TRAINPROTOBEAM (\mathcal{D})

Select indices for target beams in this episode

$V \leftarrow \text{RandSample}(\{1, \dots, B\}, N_B)$

for $b \in V$ **do**

$I_b \leftarrow \text{RandSample}(\mathcal{D}_b, n_S)$ \triangleright Support

$Q_b \leftarrow \text{RandSample}(\mathcal{D}_b \setminus I_b, n_Q)$ \triangleright Query

$p_b \leftarrow \frac{1}{n_S} \sum_{(x_i, y_i) \in I_b} f_\phi(x_i)$ \triangleright Compute Prototypes

end for

$L \leftarrow 0$ \triangleright Initialize loss for this episode

for $b \in V$ **do**

for $(x_i, y_i) \in Q_b$ **do**

$$L \leftarrow L + \frac{1}{N_B n_Q} \left[d(f_\phi(x_i), p_b) + \log \left(\sum_{b'} \exp(-d(f_\phi(x_i), p_{b'})) \right) \right]$$

end for

end for

Compute gradients of L w.r.t. model parameters ϕ

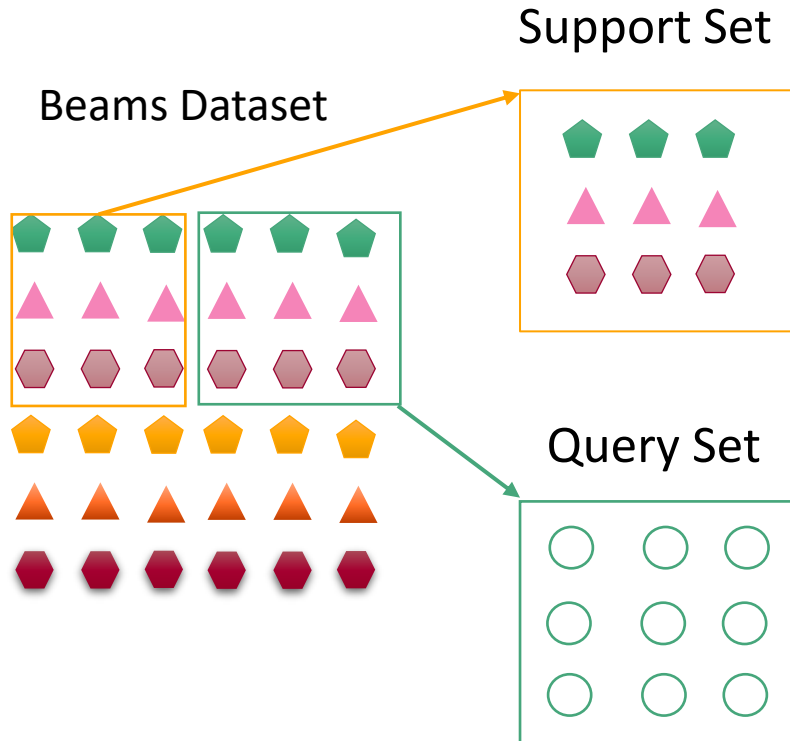
Perform backpropagation to update model parameters

$\phi \leftarrow \phi - \alpha \cdot \nabla_\phi L$ \triangleright Update ϕ with learning rate α

return ϕ

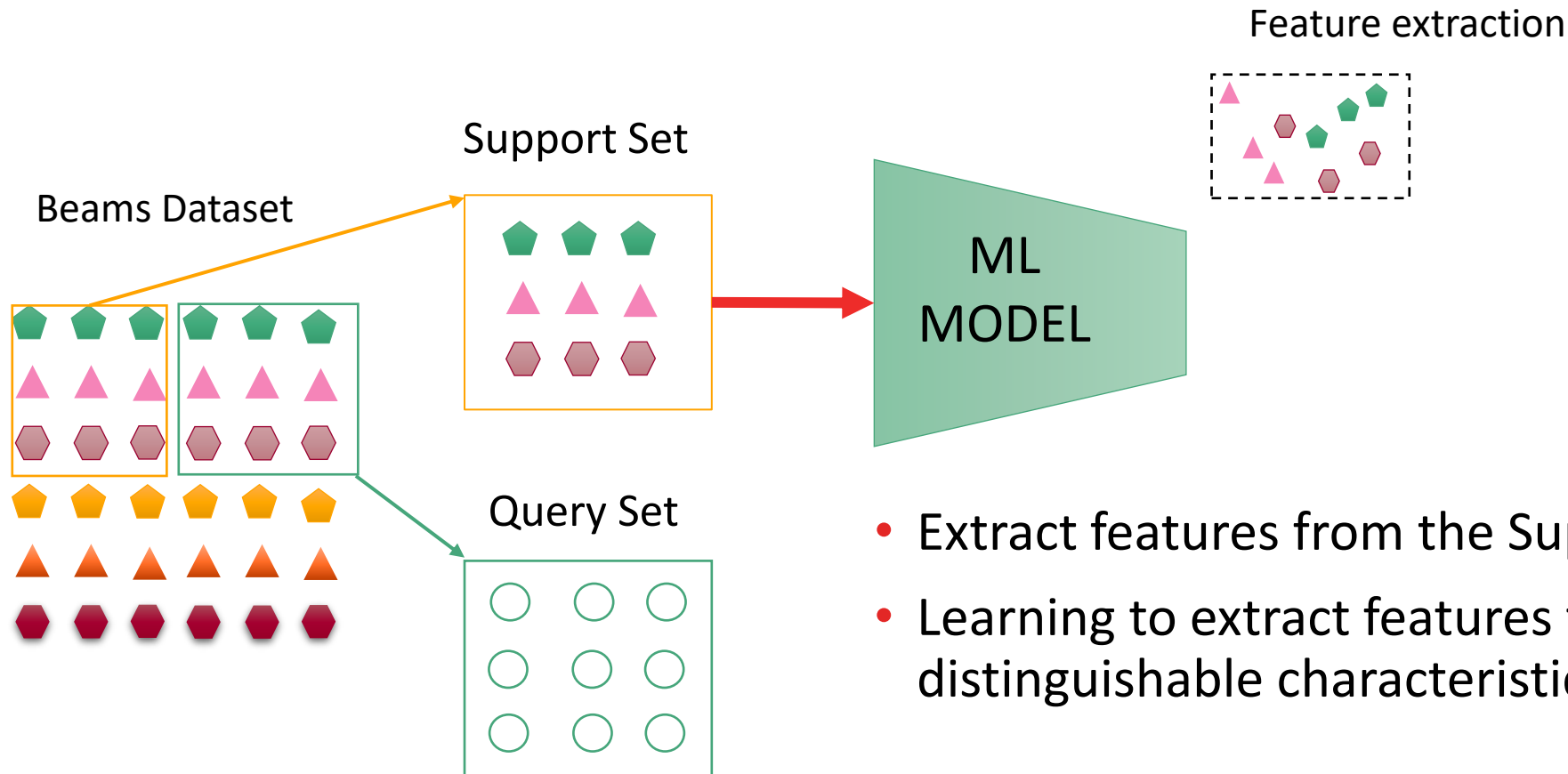
end procedure

ProtoBeam Training : Data sampling

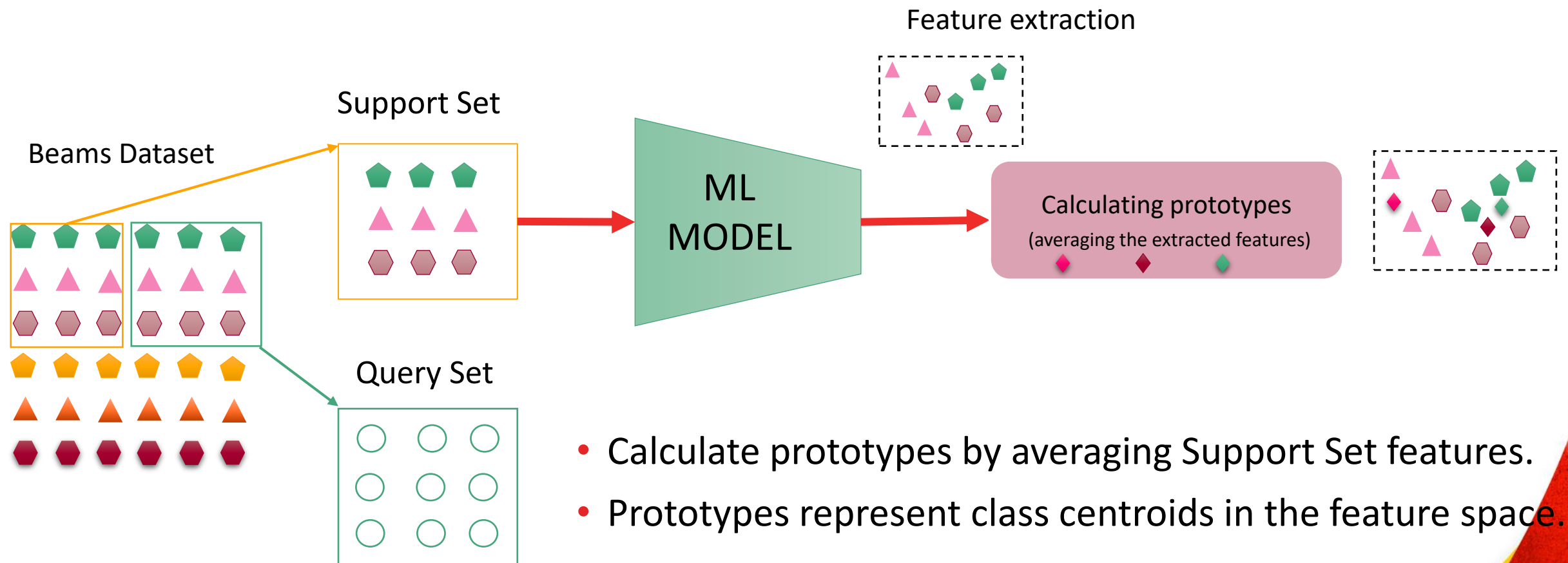


1- Randomly select of two beams sets :

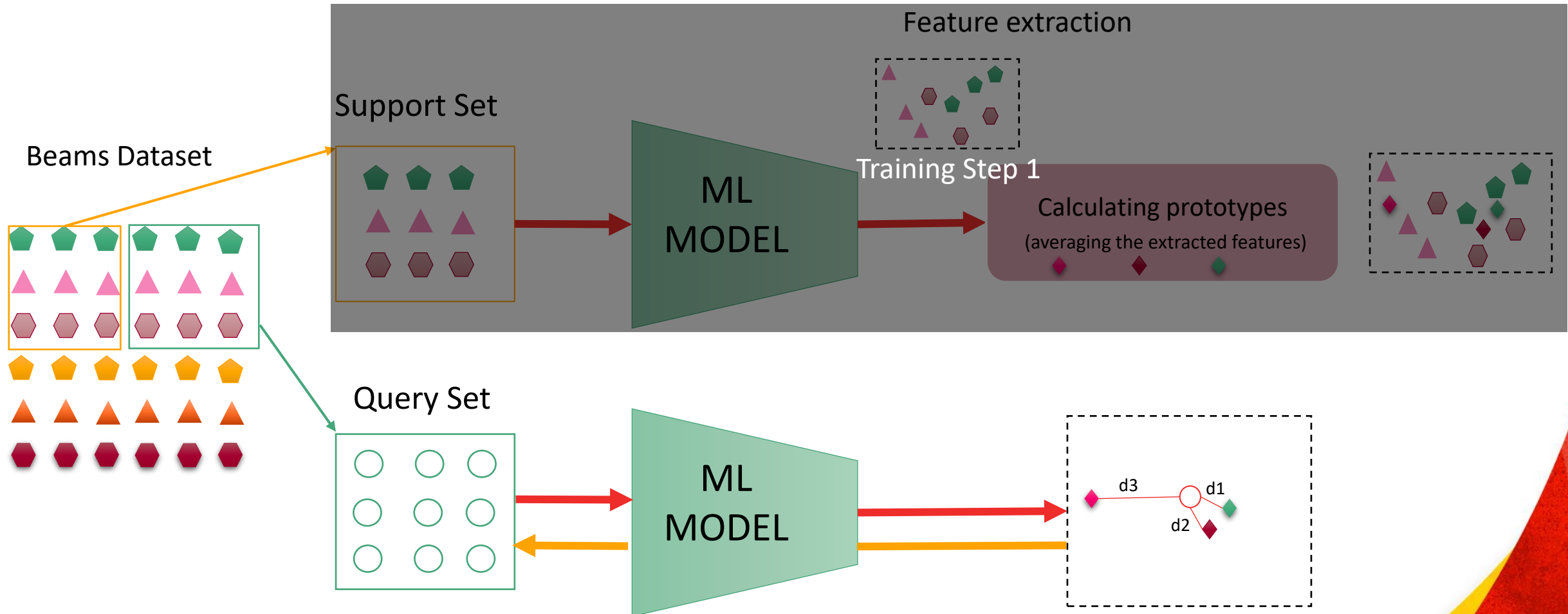
- **Support Set:** Labeled examples for each beam.
- **Query Set:** Examples used later for evaluation. The loss from these examples will update the weights of the encoder



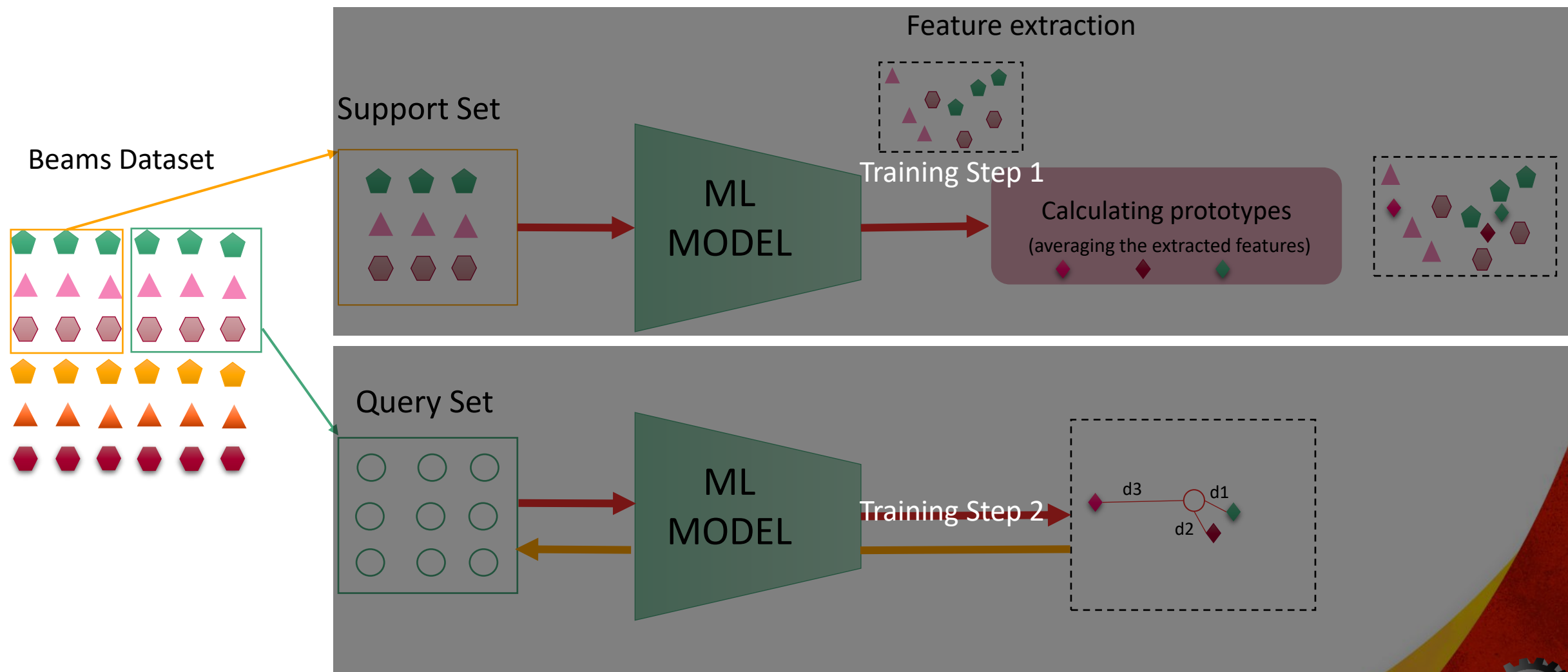
- Extract features from the Support Set using a ML model.
- Learning to extract features that represent the distinguishable characteristics of each beam.

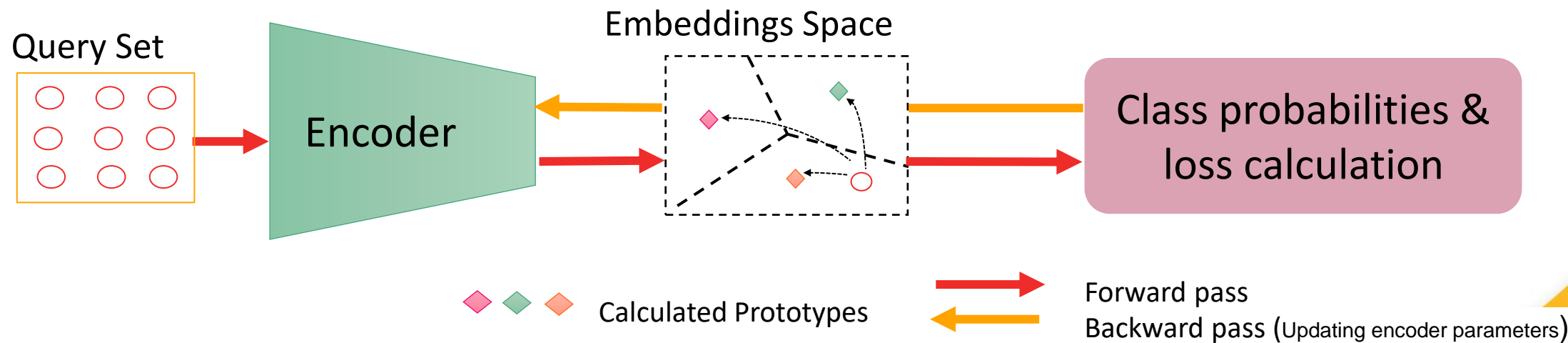
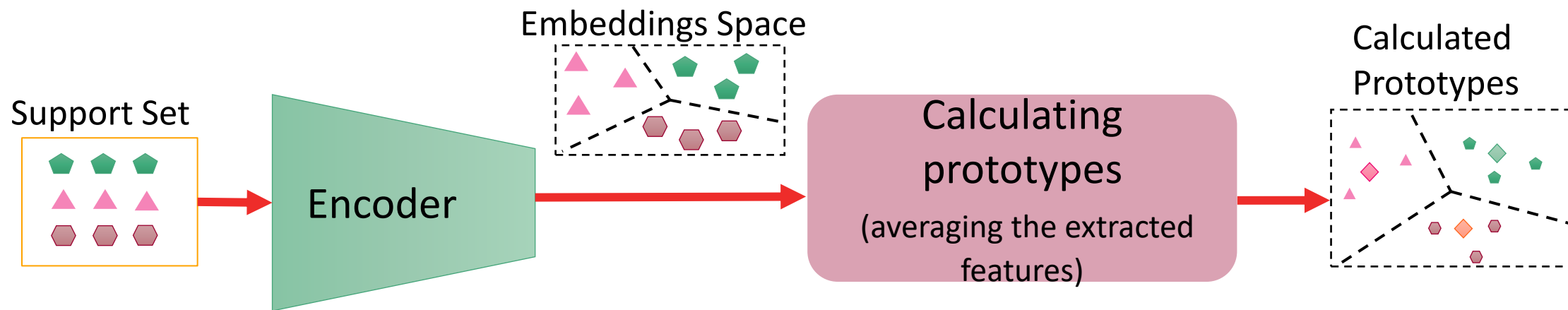


- Calculate prototypes by averaging Support Set features.
- Prototypes represent class centroids in the feature space.



- Compute loss by comparing Query Set features to prototypes.
- Minimize distance to improve classification accuracy.





Enhancing ProtoBeam with Augmentations

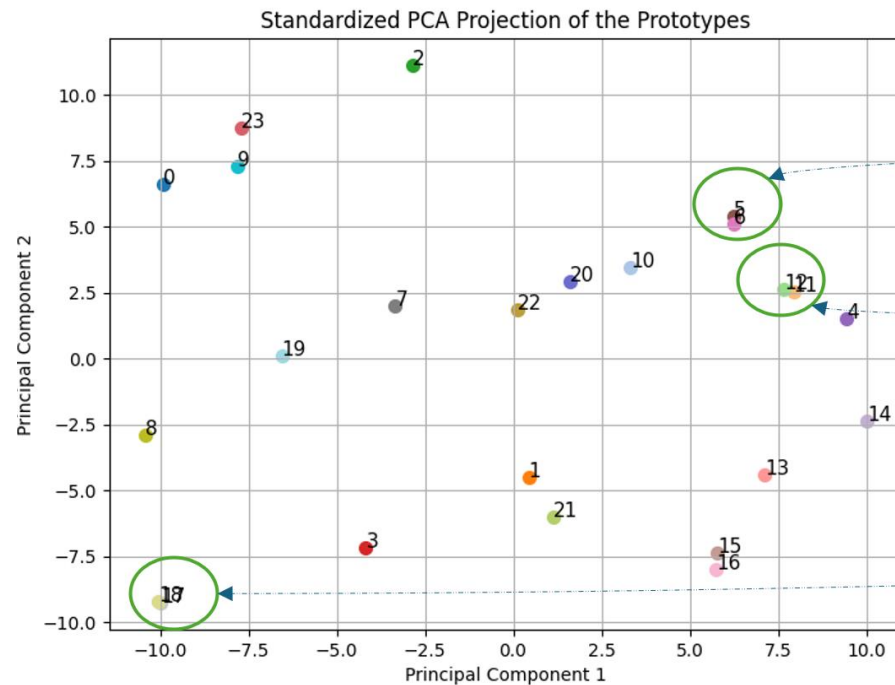
- We tested normalization, data augmentation, and prototype normalization on model accuracy. Starting from a 38.67% baseline against different antenna configurations, each technique was applied sequentially

TABLE I: ProtoBeam Performance with Data Augmentation and Normalization.

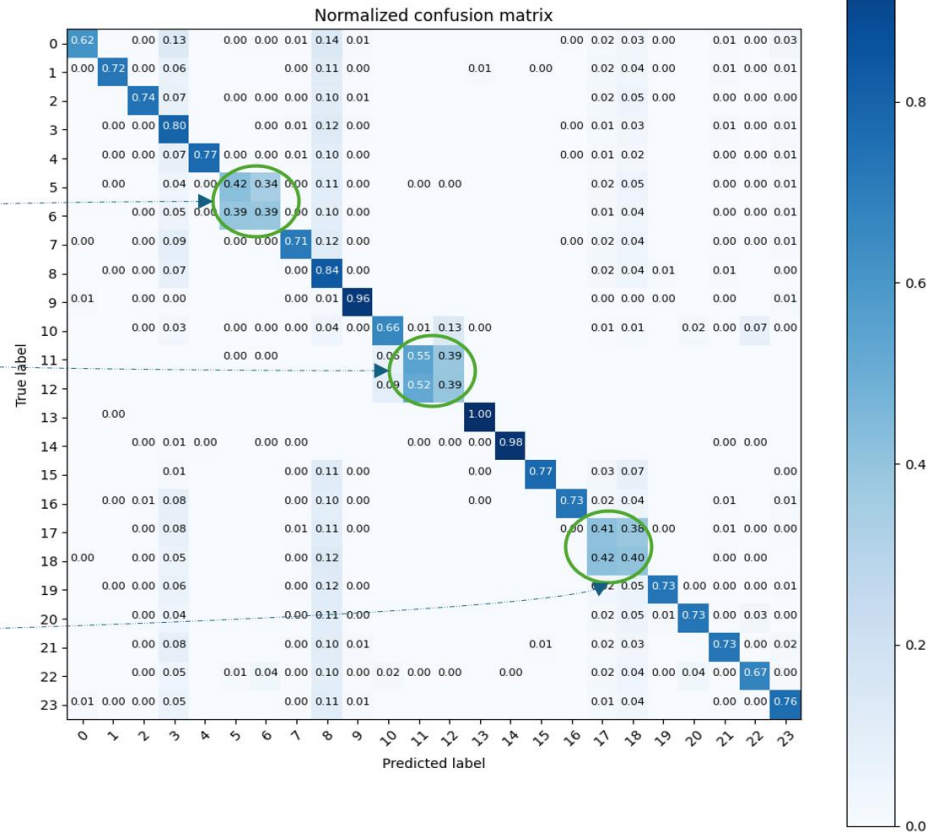
Experimental Setup	TTSA (%)		TOTA (%)	
	2-shot	32-shot	2-shot	32-shot
w/o Data Norm. or Augm.	61.2	73.4	38.67	42.8
Data Normalization only	77.3	83.3	45.8	56.8
Data Norm. & Augm.	79.69	83.68	49.9	60.4
Prototypes Norm. + Data Norm & Augm.	81.9	84.5	55.26	64.2

Interpreting the Embedding Space via PCA

- ProtoBeams mistakes were mainly in adjacent beams
- PCA projections of the embeddings show that different beams have distinct embeddings



PCA projections



Key Takeaways [1/3]

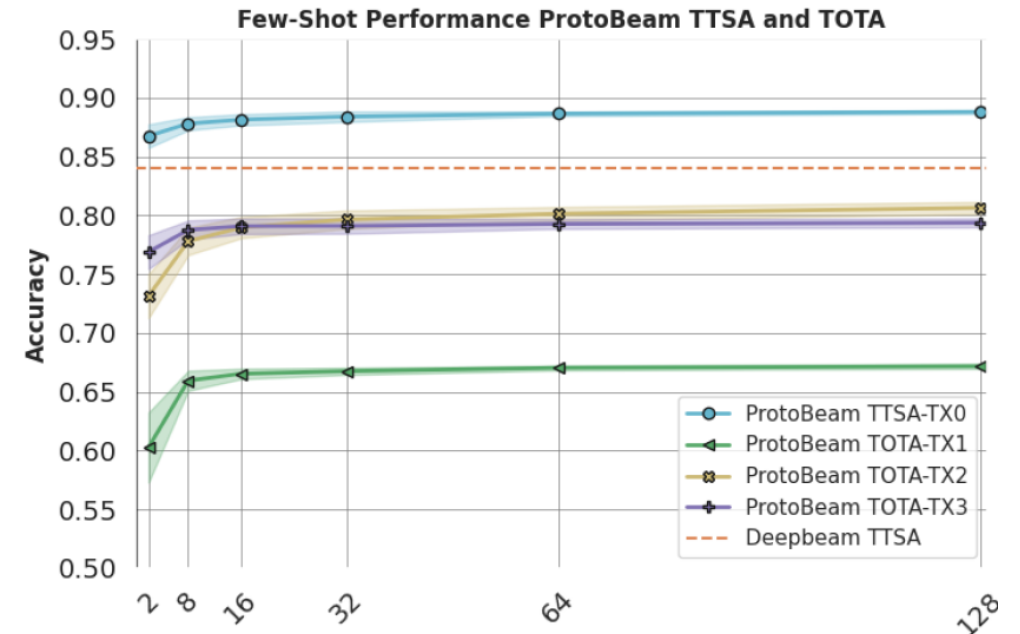
(Generative)

Radio

Embeddings

enabled

Accelerated domain adaptation with
2-8 labeled samples per
beam only (fine-tuning in
the LLM terminology)



Key Takeaways [2/3]

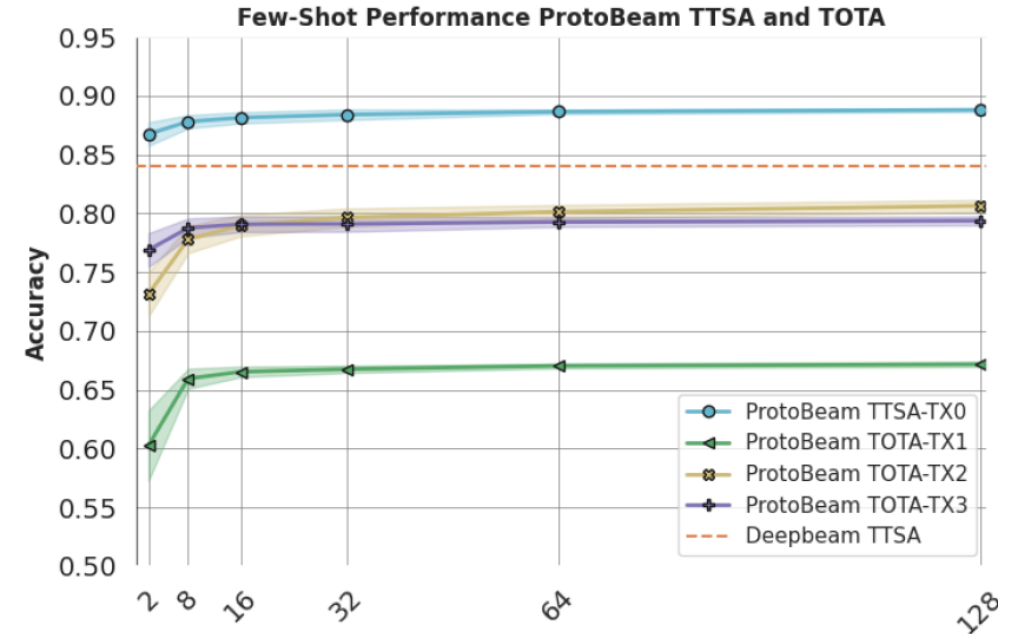
(Generative)

Radio

Embeddings

enabled

Trustworthy AI models that can
generalize to multiple RF
front ends



Key Takeaways [3/3]

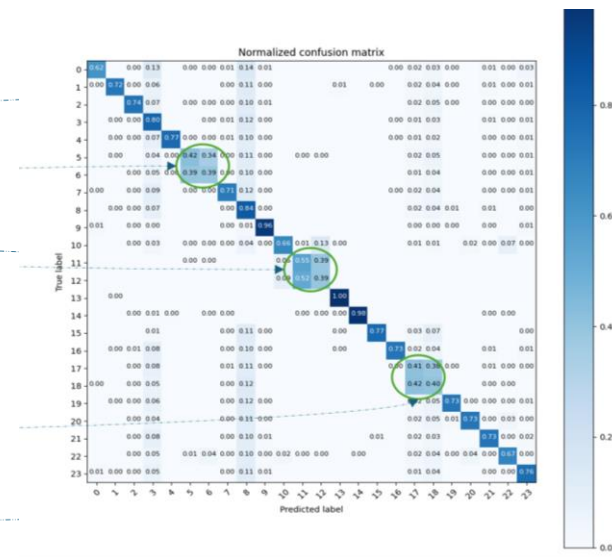
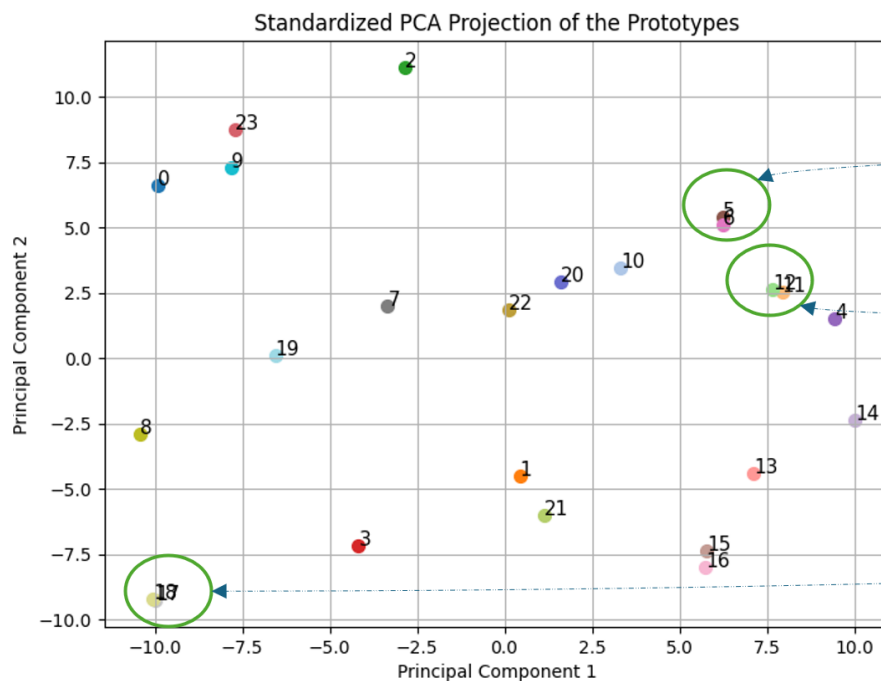
(Generative)

Radio

Embeddings

enabled

Trustworthy *representations*
that are
interpretable in
lower feature
spaces



Generative Radio Embeddings for Foundation Models in 6G



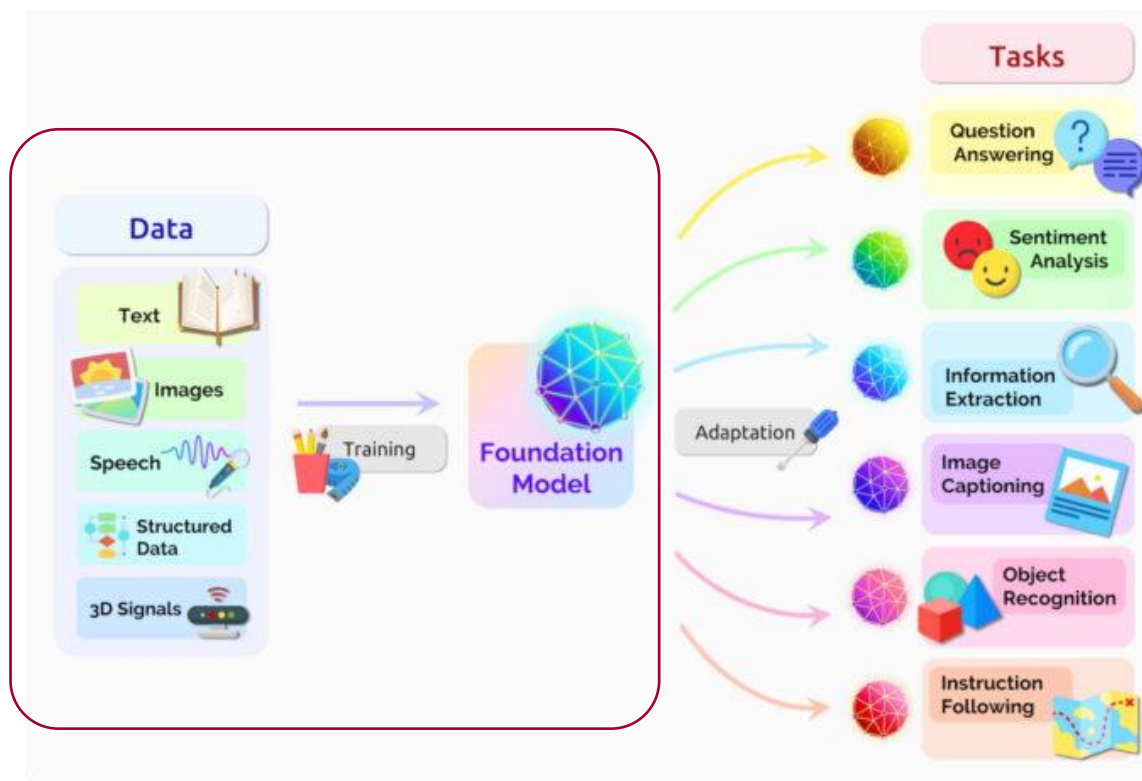
Ahmed Aboufotouh
PhD student



Can we do better than Domain Adaptation?

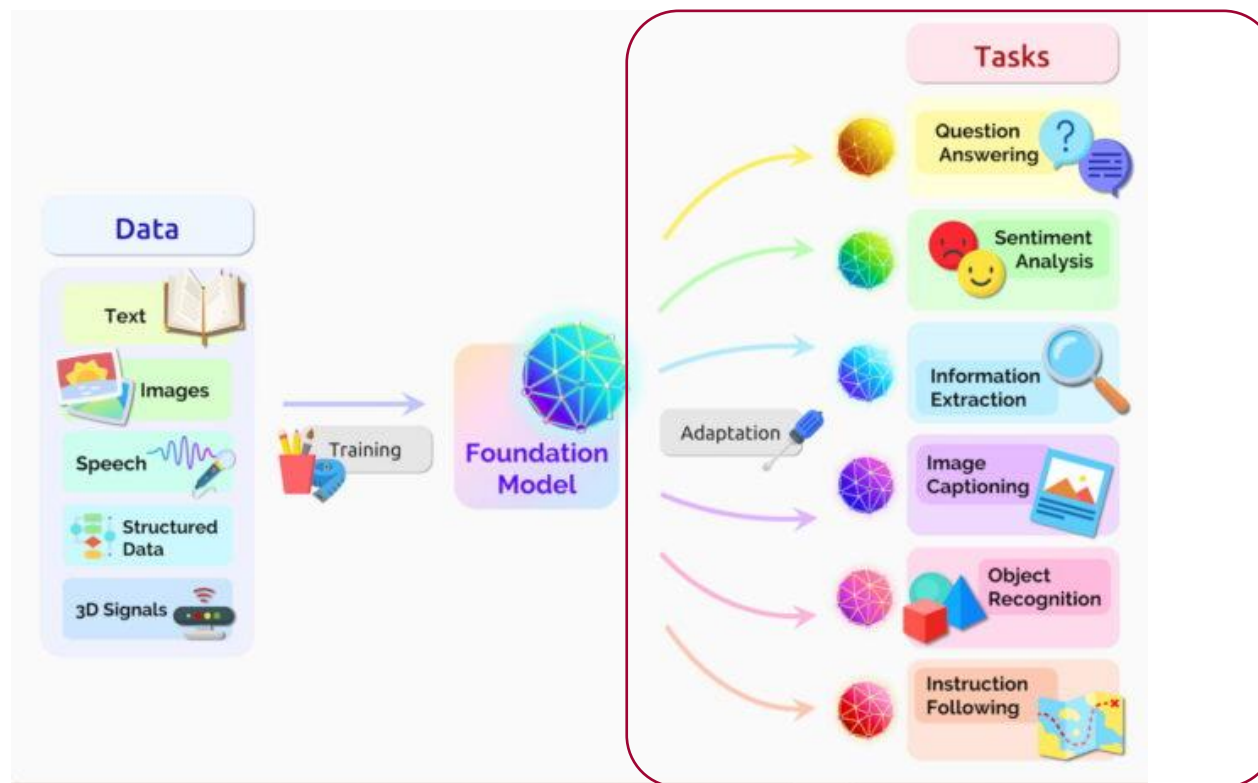
1. Can we learn more generic representations from wireless signals for multiple-tasks?
 - **Generative** Radio Embeddings that enable **multiple downstream tasks**?
2. Can we learn these embeddings in a self-supervised fashion without requiring labeled data?

Foundation Models (FM): *Pre-training*



<https://blogs.nvidia.com/blog/what-are-foundation-models/>

Foundation Models (FM): *Fine-tuning*



<https://blogs.nvidia.com/blog/what-are-foundation-models/>

Toward a Foundation Model for Spectrogram Learning

- Can be used for different tasks like spectral occupancy prediction, classification of signals in the spectrum
- Broader goal is to enable more complex and diverse spectrogram-based tasks

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.

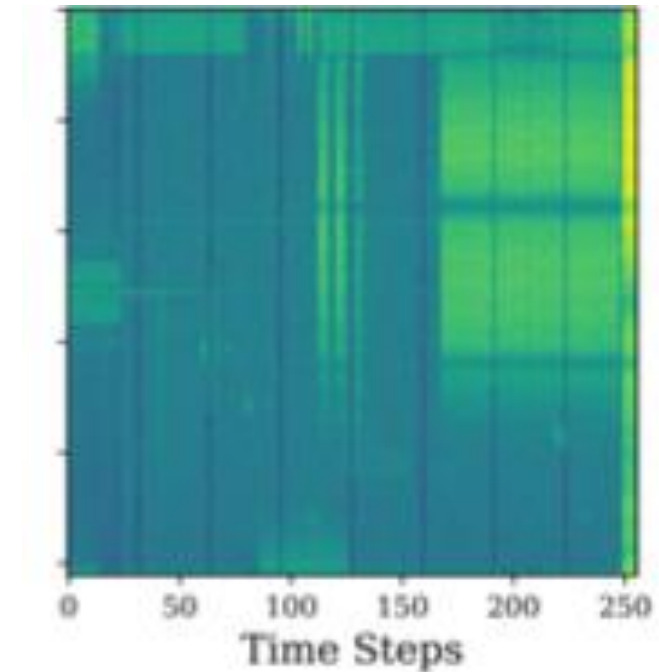
Index Terms—Self-Supervised Learning, Deep Learning, Foun-

evolution in NLP and computer vision. The reliability of these models across data distribution shifts and their ability to generalize is also usually limited.

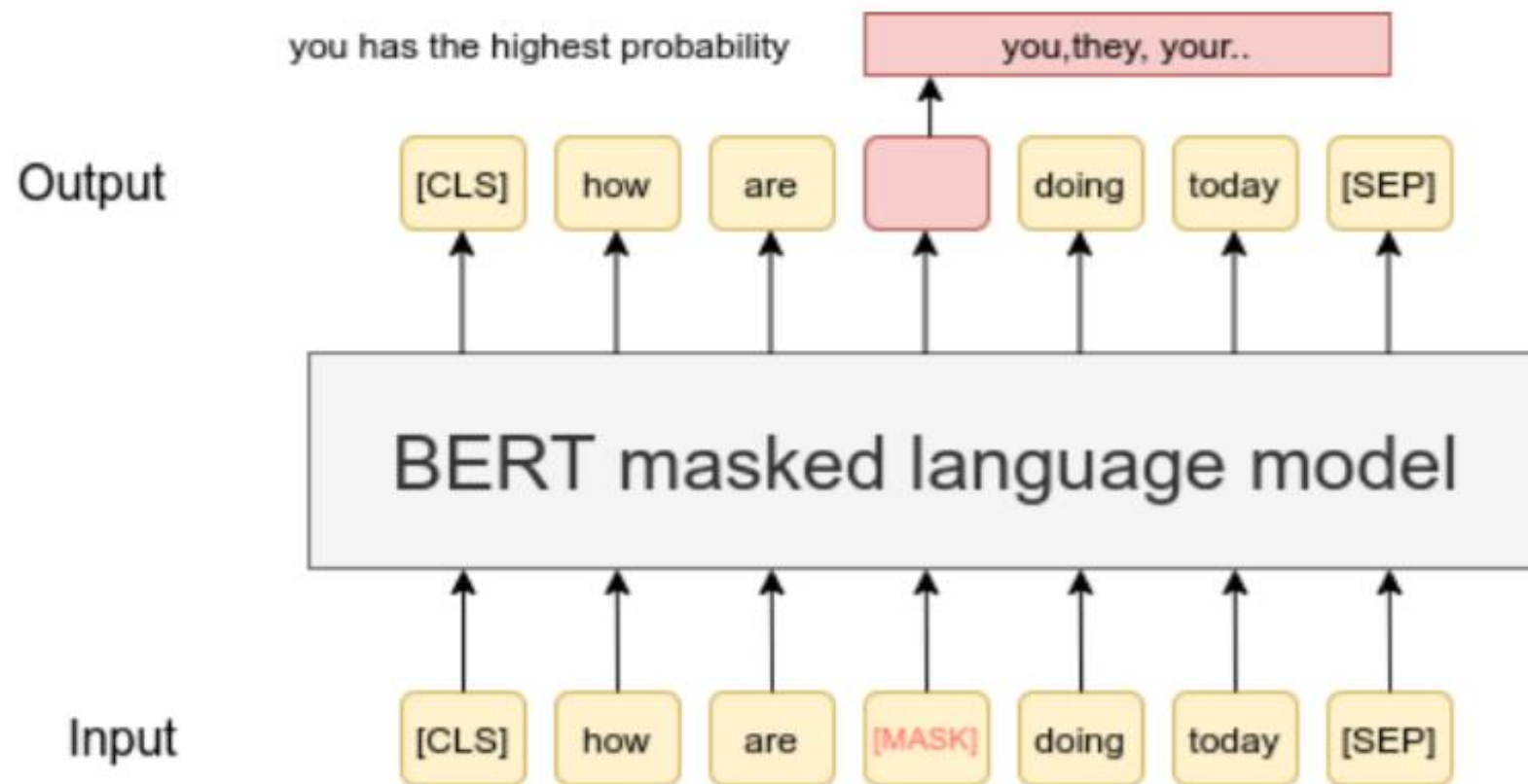
Introducing the concept of foundational models into wireless communication holds substantial promise to overcome these limitations [9]. We argue that as in NLP and computer vision, where a wealth of unlabeled data exists — communication signals can be harnessed for pretraining such foundational models 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

The Dataset

- Time-domain recordings of IQ samples in the frequency range of 2.4 to 2.65 GHz, with BW between 10 MHz and 60 MHz using Pluto and Ettus SDRs.
- The samples were converted to spectrograms and used for foundational model pretraining



“tokens”, “sentences”, and “masking” in language models



“radio tokens” and “radio sentences”

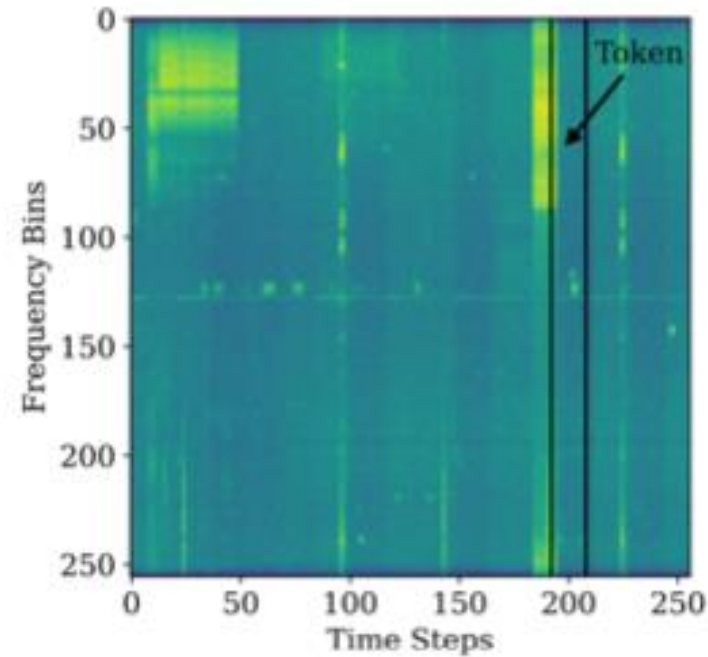
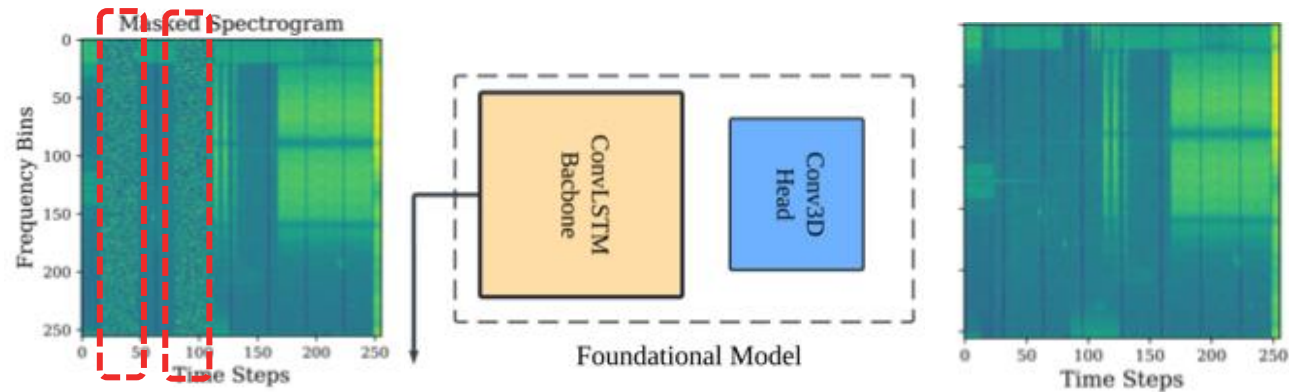


Fig. 2: A radio sentence created from the RRD dataset.

Masked Spectrogram Modeling

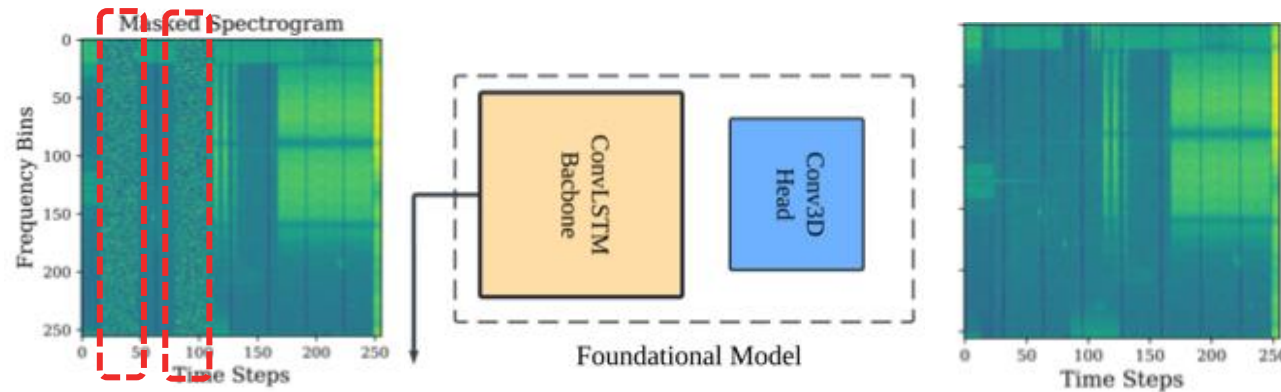
- Foundation model pre-training (without labels)



- Masking involves replacing the actual content of the spectrogram with white noise.
- 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.

Masked Spectrogram Modeling

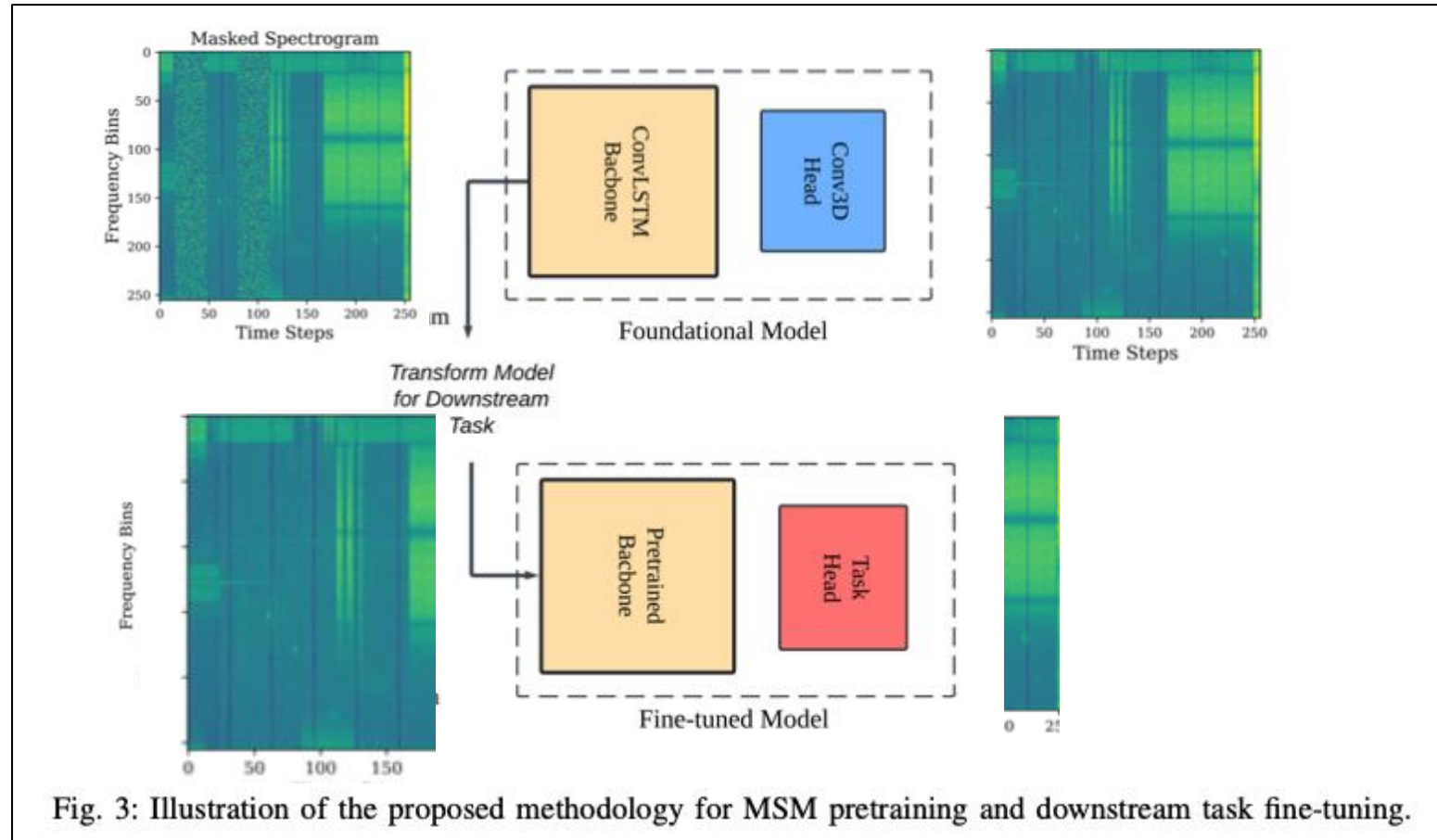
- Foundation model pre-training (without labels)



Lots of design choices

- Masking involves replacing the actual content of the spectrogram with white noise.
- 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.

Fine-tuning for a Spectrogram *Forecasting* Task



Pretrained
Backbone weights
were frozen

Results

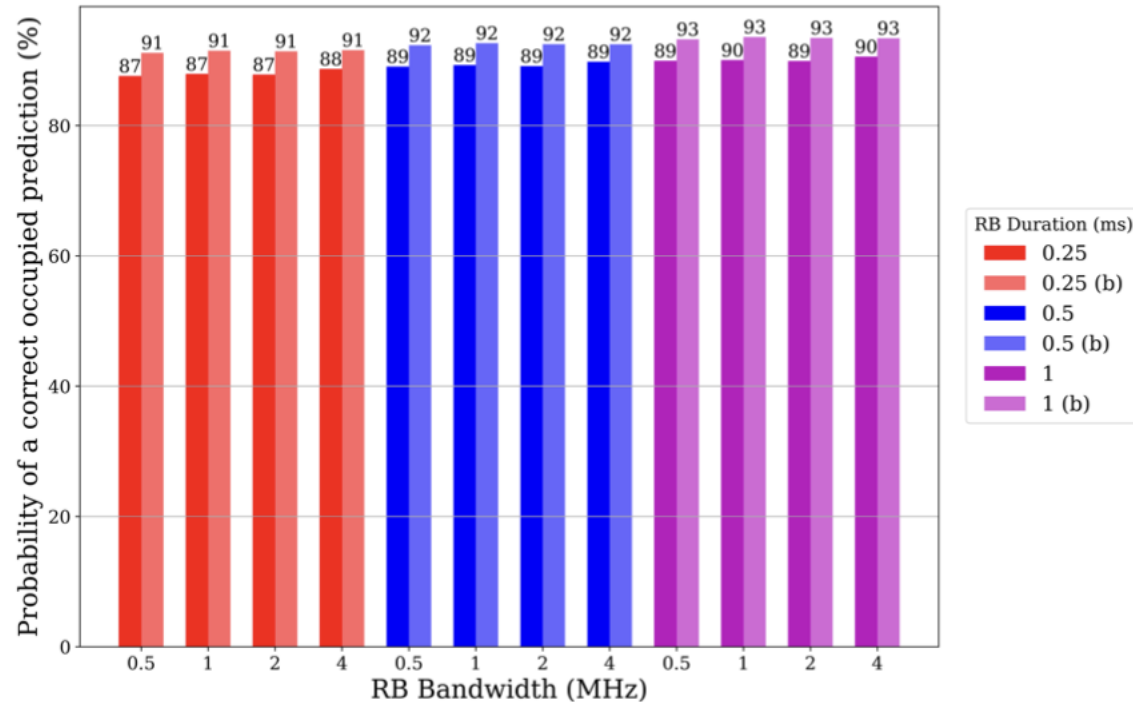
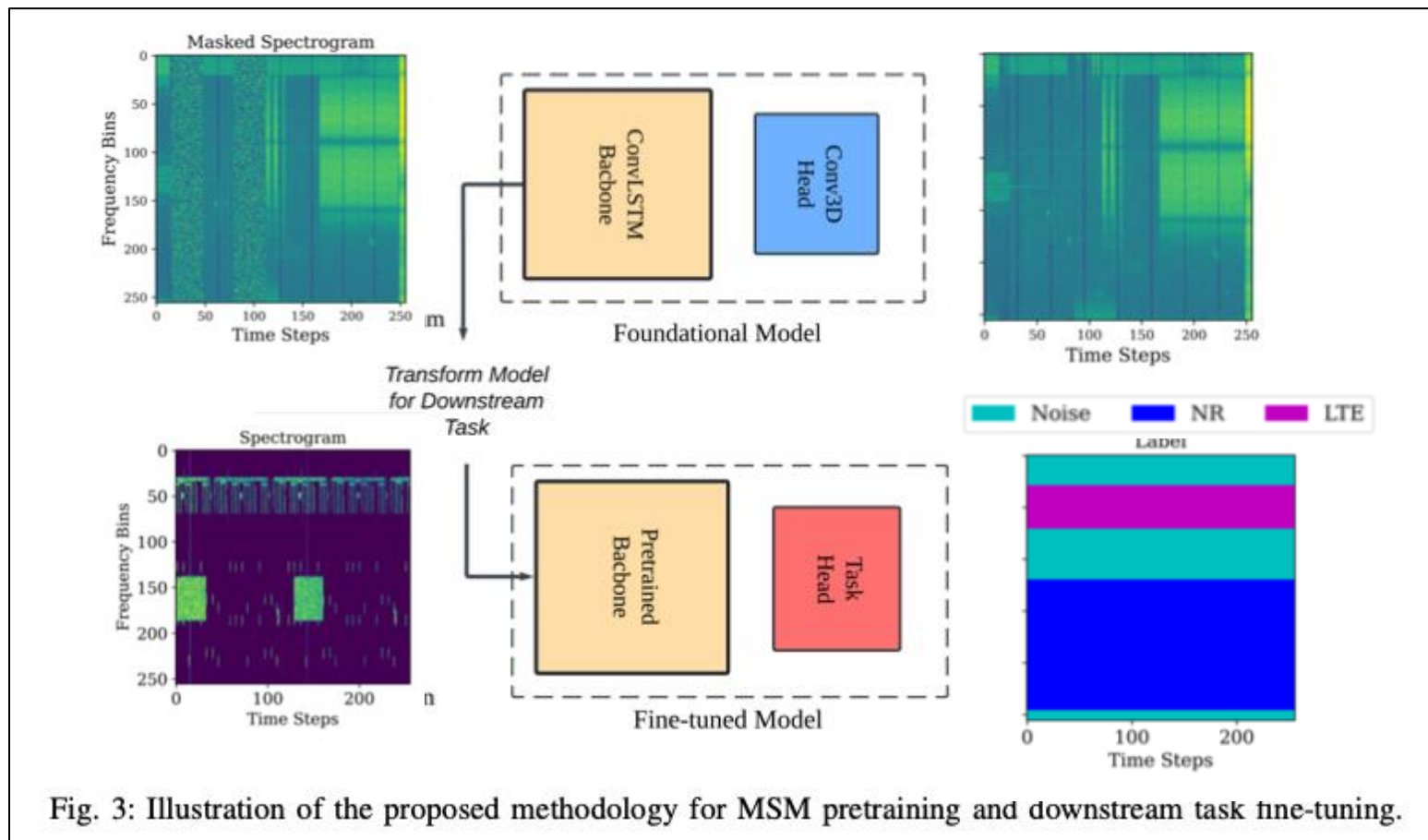


Fig. 6: Probability of correct occupied predictions. The solid lines are the foundational tuned model and (b) is the baseline.

- Results show very close performance to the baselines models that are trained from scratch.
- Important to note that the backbone weights are frozen and only the conv3D head was fine-tuned.
- Fine-tuning took at least an order of magnitude less time to train

Fine-tuning for a very different “Segmentation Classification Task”



Pretrained
Backbone weights
were frozen

Results

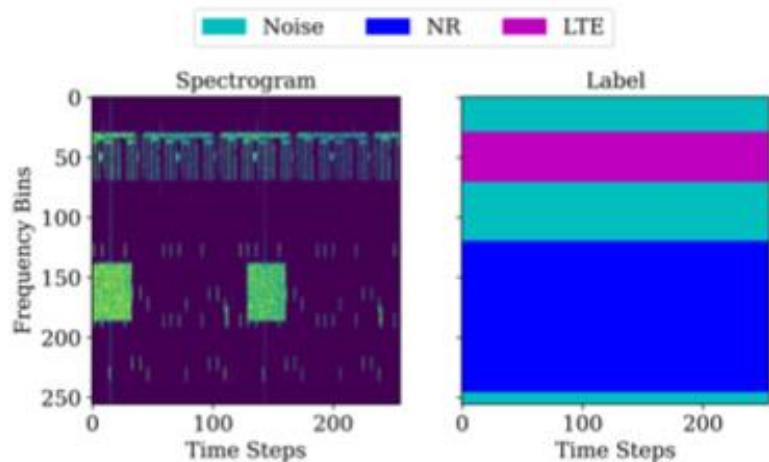


Fig. 1: A spectrogram and label pair for the segmentation task.

- Results show promise but are not on par with a trained model from scratch.
- Important to note that the backbone weights are frozen and were trained on regression task of spectrogram “filling” and not a classification task
- Consequently, the learned features may not generalize as effectively to segmentation as they do to forecasting.

True Labels	Noise	NR	LTE
	Noise	NR	LTE
Noise	0.79	0.13	0.08
NR	0.23	0.62	0.15
LTE	0.03	0.27	0.71

Tuned Foundational Model

True Labels	Noise	NR	LTE
	Noise	NR	LTE
Noise	0.95	0.04	0.01
NR	0.06	0.93	0.01
LTE	0.01	0.03	0.96

Baseline Model

True Labels	Noise	Signal
	Noise	Signal
Noise	0.79	0.21
Signal	0.13	0.87

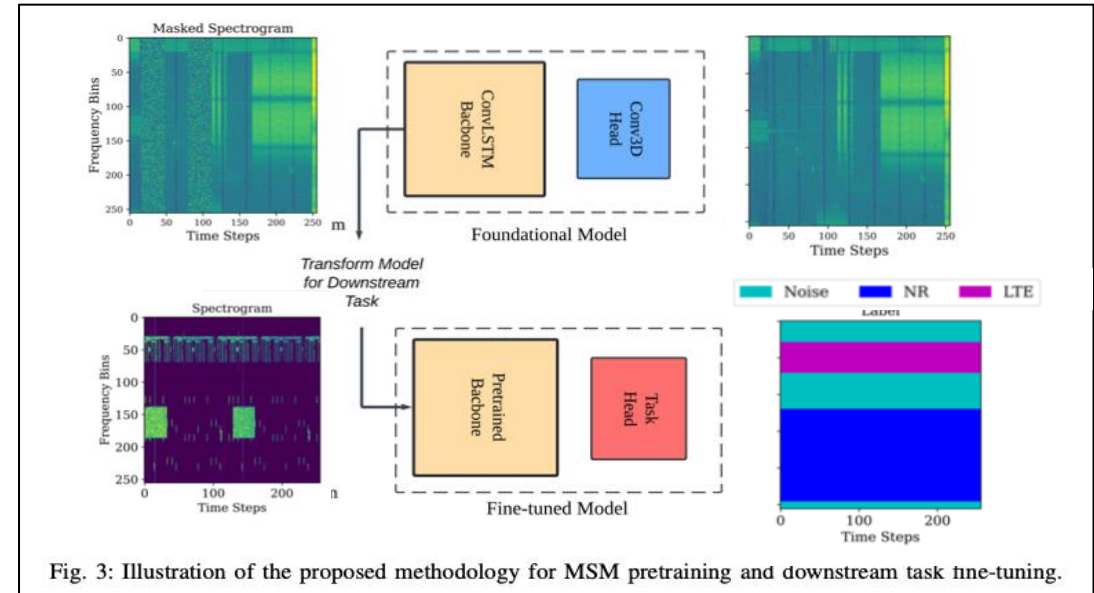
Tuned Foundational Model

True Labels	Noise	Signal
	Noise	Signal
Noise	0.95	0.05
Signal	0.04	0.96

Baseline Model

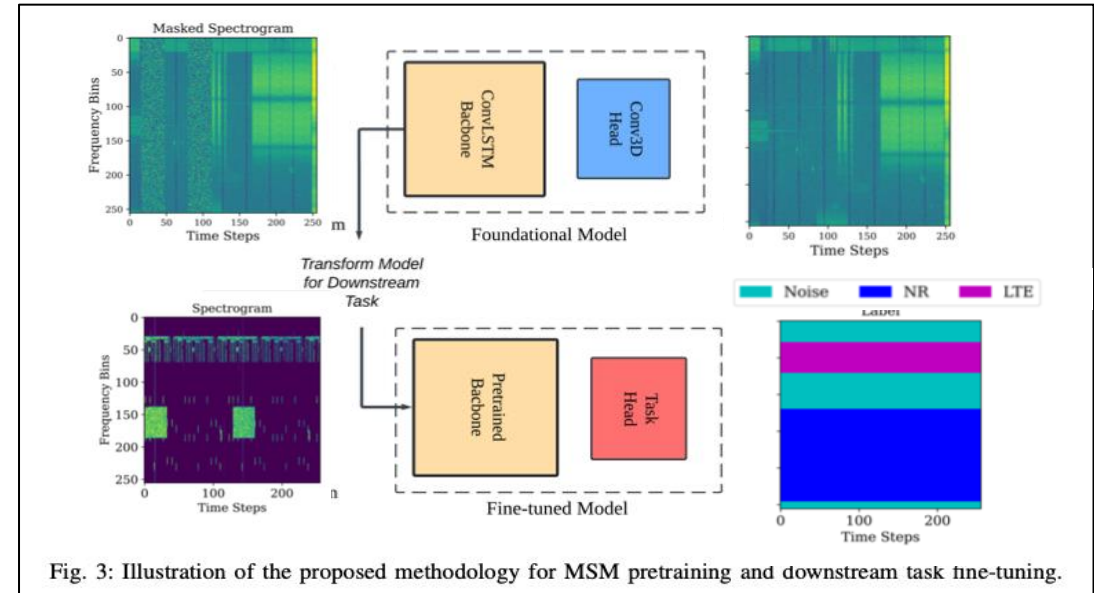
Key Takeaways [1/2]

- The 'Masked Spectrogram Modeling' pre-training approach was able to learn radio embeddings with a relatively small dataset of unlabeled spectrogram data.
- The model was successfully fine-tuned to two different spectrogram tasks



Key Takeaways [2/2]

- The 'Masked Spectrogram Modeling' pre-training approach was able to learn radio embeddings with a relatively small dataset of unlabeled spectrogram data.
- The model was successfully fine-tuned to two different spectrogram tasks
- This is a first step! More advanced multi-modal data and pre-training approaches need to be investigated.
- Other backbone architectures, radio tokenization and radio embedding are needed!



Accelerated AI



Fazal Khan
Ph.D Student



Elsayed Mohammed
M.Sc. Student

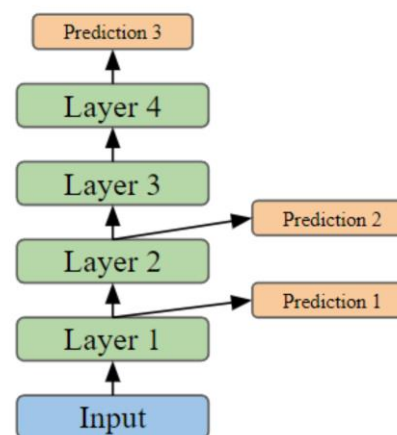


Mohammad Hallaq
M.Sc. Student



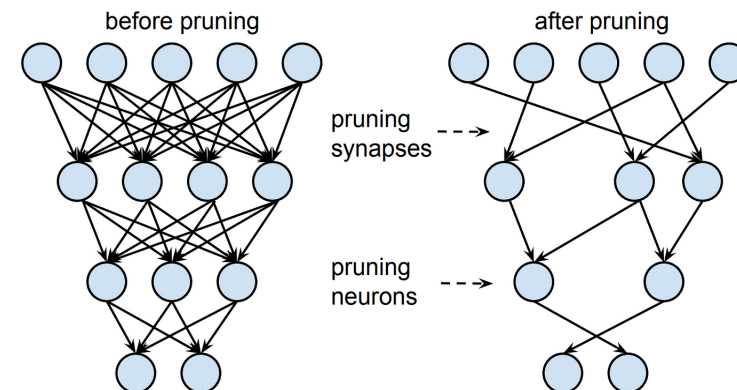
Accelerating Deep Learning for Wireless

1. Multi-branch neural networks for faster inference in wireless networks



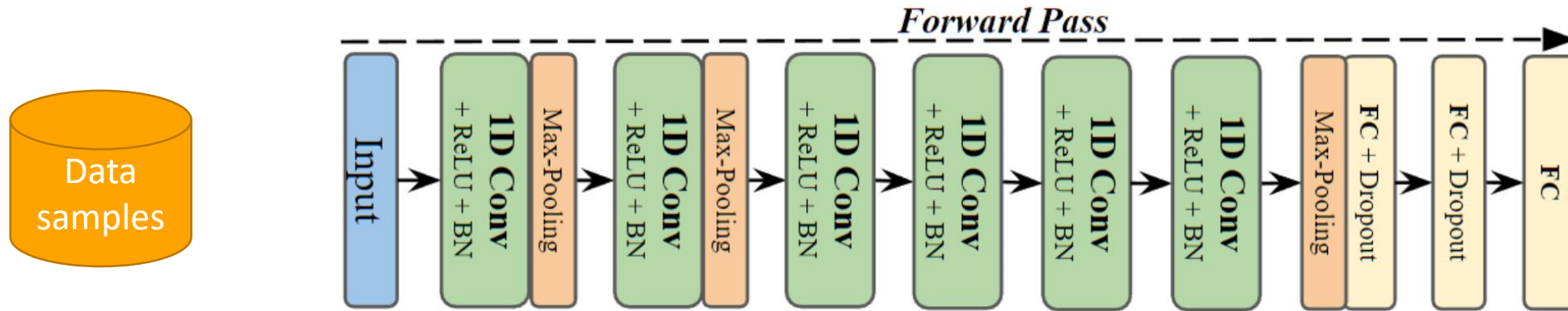
Demonstration of Early Exiting Inference.

2. Model pruning and quantization



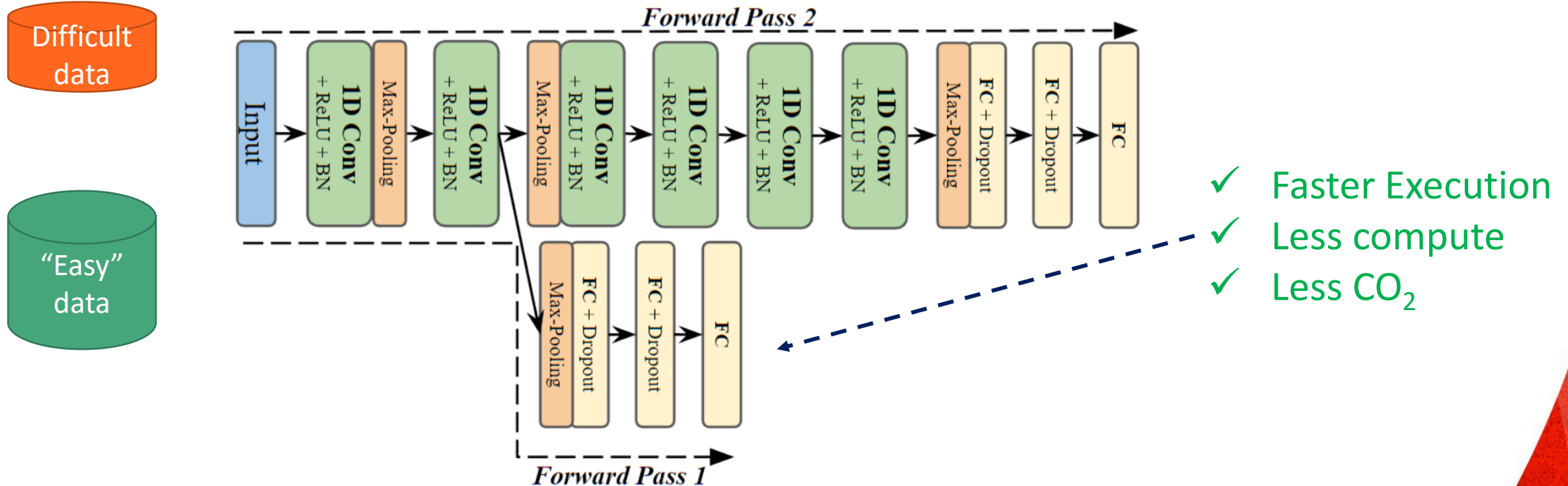
Han, Song, et al. "Learning both weights and connections for efficient neural network." *Advances in neural information processing systems* 28 (2015).

Deep Neural Networks typically have 1 exit



- High accuracies often require *deep* neural network architectures
- However, the deep architectures may not be needed for many samples within the dataset

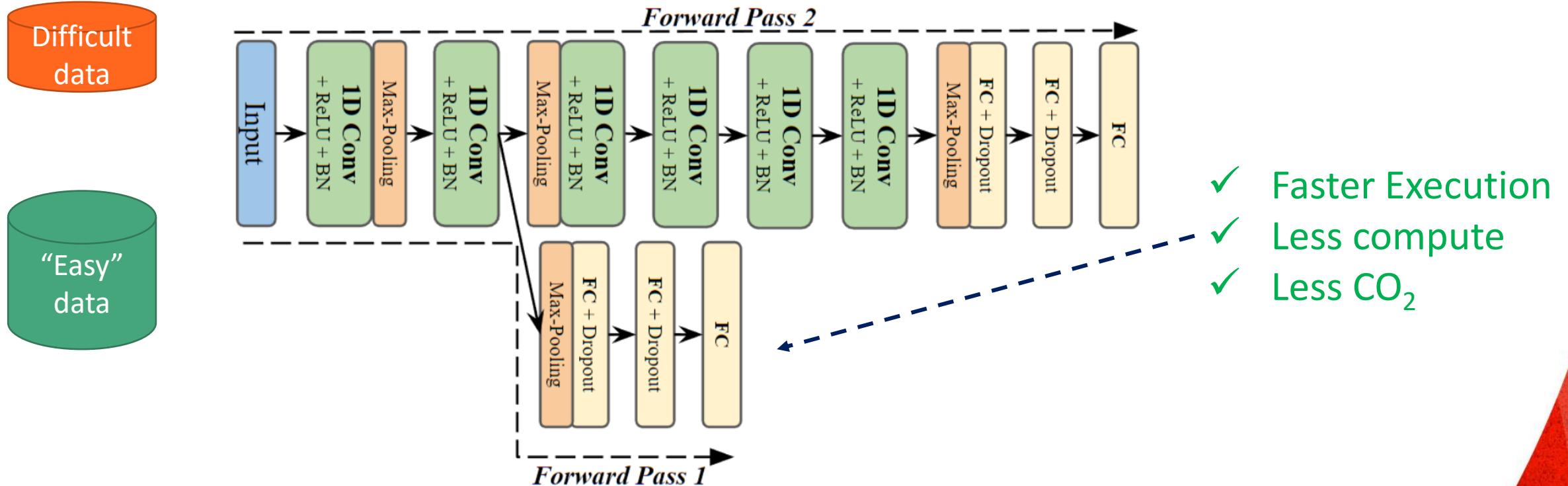
Deep Neural Networks with Multiple Exits



- It is possible to create multi-branch neural architectures (proposed in “Branchynet” ICPR, 2016)
- “Easy” tasks use the short branch and difficult tasks use longer branches
- This should happen dynamically

S. Teerapittayanon, B. McDanel, and H.-T. Kung, “Branchynet: Fast inference via early exiting from deep neural networks,” in IEEE International Conference on Pattern Recognition (ICPR), pp. 2464–2469, IEEE, 2016.

Deep Neural Networks with Multiple Exits



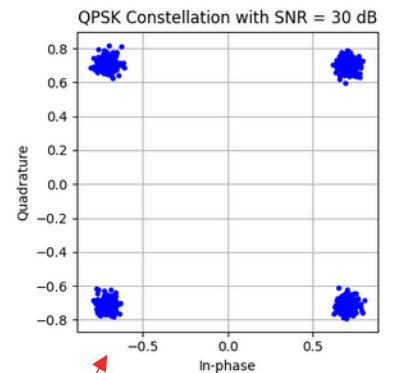
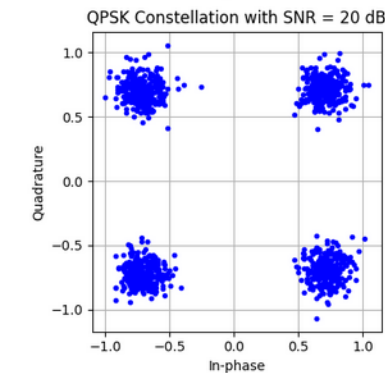
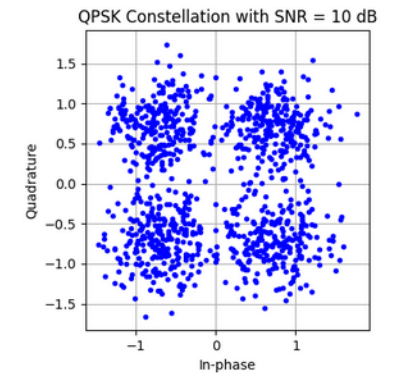
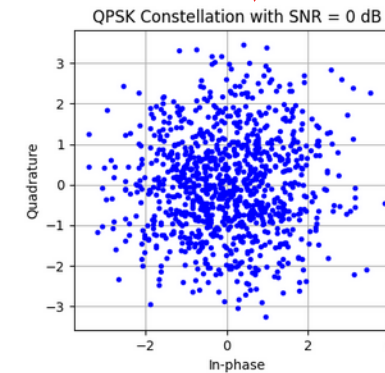
Main idea:

- Execute forward pass 1 first, evaluate confidence of classification (can be entropy),
- If this confidence is greater than a threshold then exit, else continue to evaluate forward path 2

Can multi-exit DL models handle different Signal to Noise ratios efficiently?

- High SNR goes through the short branch, medium to low SNR through longer branches
- Consider the task of Automatic Modulation Classification (AMC) from IQ data
 - Noise affects the difficulty of the classification tasks.

Difficult to classify: noise obscures the signal



Easy to classify: clear and distinct

Using Early Exits for Fast Inference in Modulation Classification

2023 IEEE Global Communications Conference: Mobile and Wireless Networks

Using Early Exits for Fast Inference in Automatic Modulation Classification

Elsayed Mohammed, Omar Mashaal, and Hatem Abou-Zeid

Department of Electrical and Software Engineering, University of Calgary, Calgary, Canada
{elsayed.mohammed, omar.mashaal1, hatem.abouzeid}@ucalgary.ca

Abstract—Automatic modulation classification (AMC) plays a critical role in wireless communications by autonomously classifying signals transmitted over the radio spectrum. Deep learning (DL) techniques are increasingly being used for AMC due to their ability to extract complex wireless signal features. However, DL models are computationally intensive and incur high inference latencies. This paper proposes the application of early exiting (EE) techniques for DL models used for AMC to accelerate inference. We present and analyze four early exiting architectures and a customized multi-branch training algorithm for this problem. Through extensive experimentation, we show that signals with moderate to high signal-to-noise ratios (SNRs) are easier to classify, do not require deep architectures, and can therefore leverage the proposed EE architectures. Our experimental results demonstrate that EE techniques can significantly reduce the inference speed of deep neural networks without sacrificing classification accuracy. We also thoroughly study the trade-off between classification accuracy and inference time when using these architectures. To the best of our knowledge, this work represents the first attempt to apply early exiting methods to AMC, providing a foundation in this area.

Index Terms—Automatic modulation classification, Cognitive radio, Software-defined networks, CNNs, Early exiting.

I. INTRODUCTION

Automatic modulation classification (AMC) plays a critical role in wireless communications by autonomously classifying signals transmitted over the radio spectrum. It facilitates a

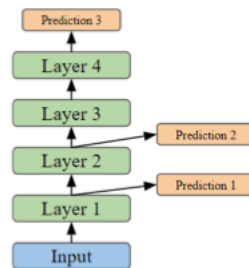
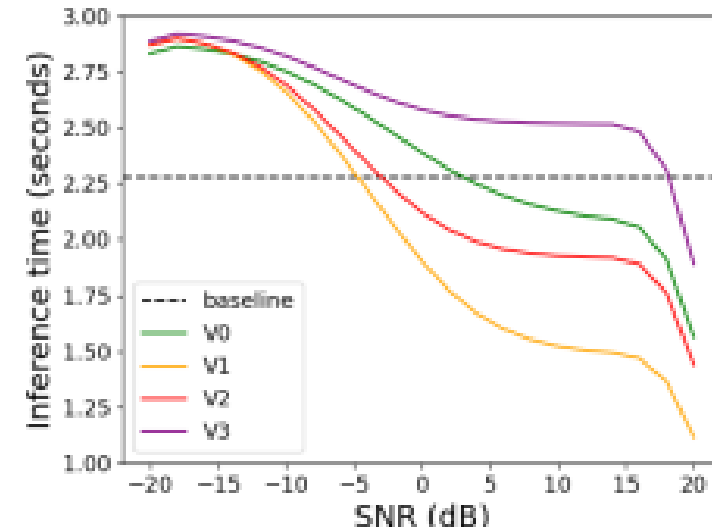


Fig. 1. Demonstration of Early Exiting Inference.

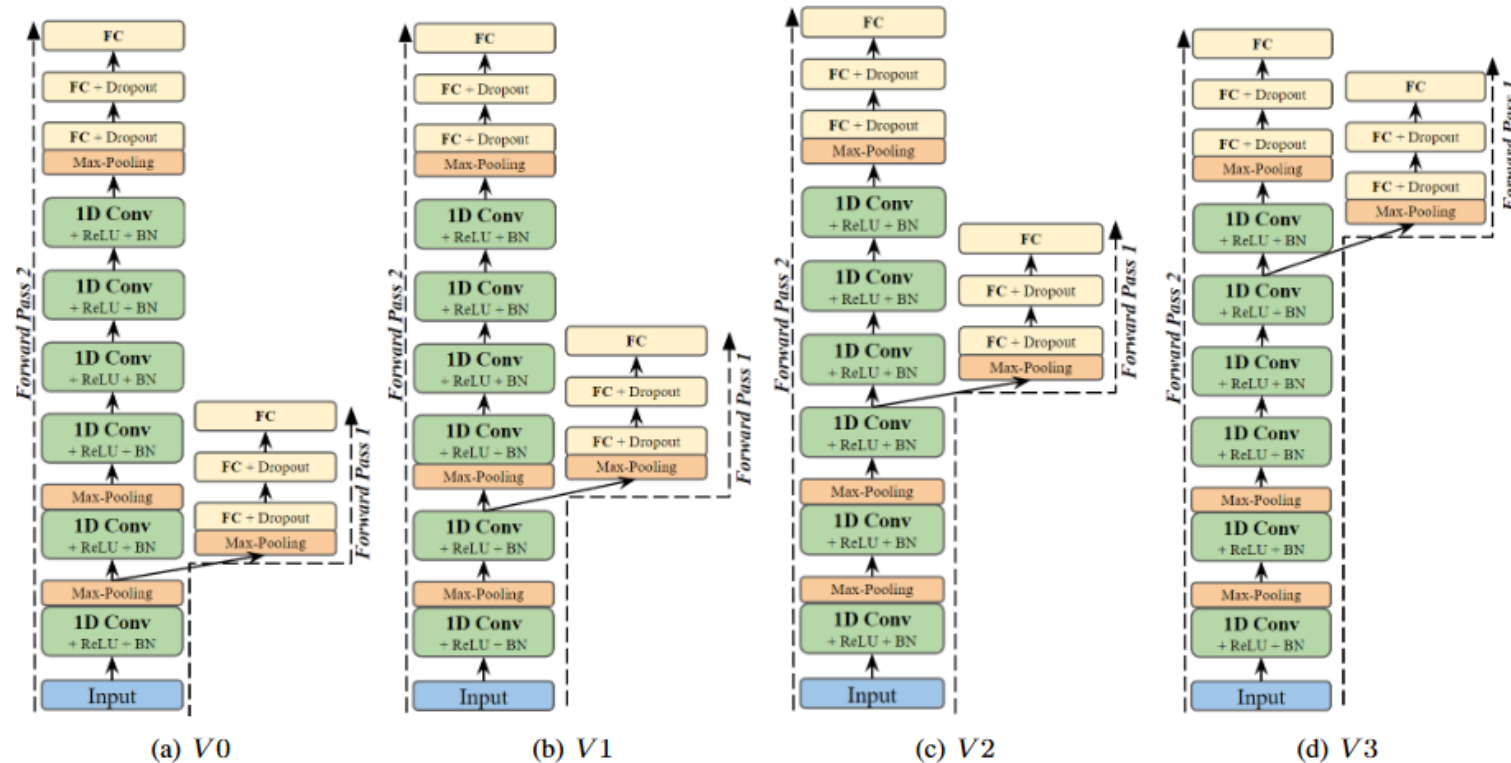
However, complex DL models with a large inference time and energy consumption may be unnecessary for signals received at high signal-to-noise (SNR) levels. Our intuition is that signals with a high SNR should be easier to classify than signals with a low SNR. In such cases, it may be sufficient to utilize shallower DL architectures that require significantly less computational resources. However, those shallow architectures may not succeed when tasked to classify signals with low SNRs. This motivates the question of whether deep learning architectures with *multiple* branches can be used to address this dilemma? The goal would be for signals with high SNRs



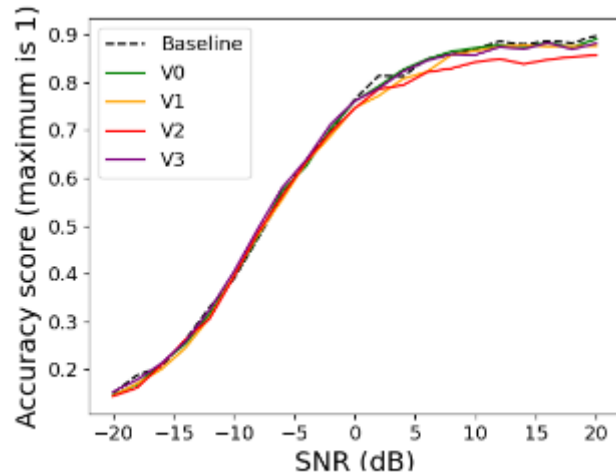
This work represents the first application of early exiting multi-branch neural networks in wireless communications tasks

Where to exit?

- Developed and analyzed four unique early exiting architectures alongside a custom multi-branch training algorithm.
- Two main design criteria: 1) where to exit, 2) what confidence threshold to use when exiting

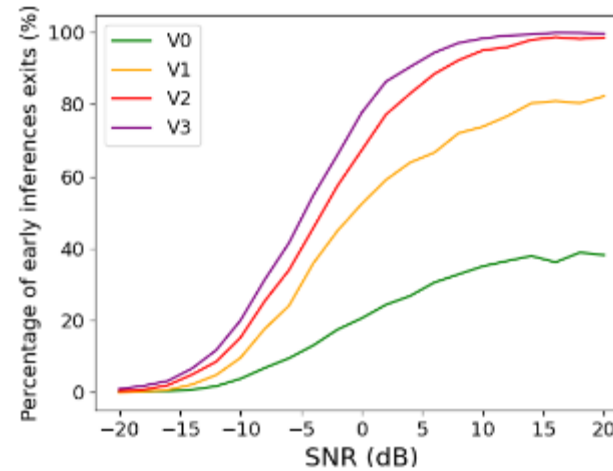


Results



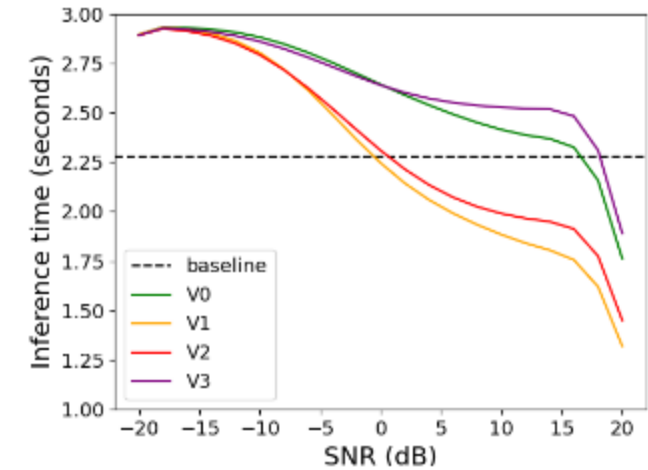
(a) Accuracy

- V0, V1, and V3 perform very close to the baseline backbone mode
- V2 was a little “over-confident”



(b) Early Inference Percentage

- Higher SNRs increase the frequency of successful early exits, indicating less need for deep architectures at these levels



(c) Inference Time

- Architectures V1 and V2 showed significant inference time reductions at positive SNRs compared to the baseline

Effect of the Exiting Threshold Value

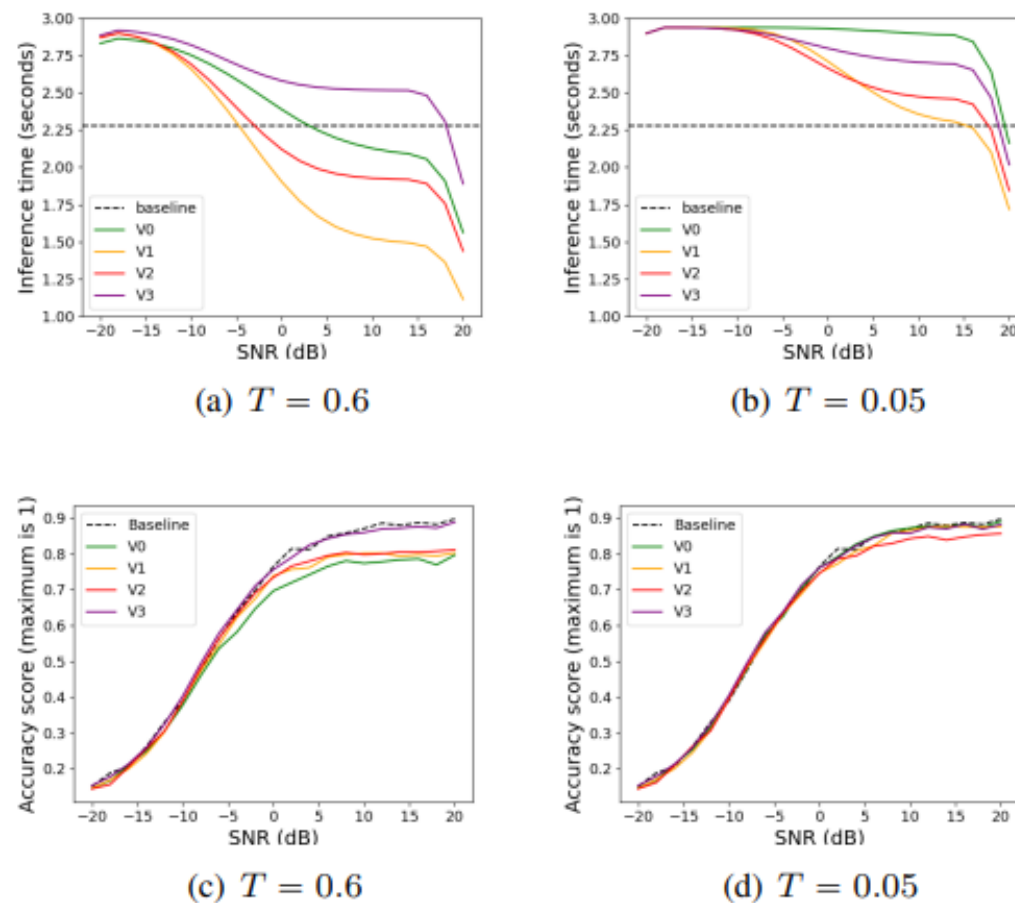


Fig. 6. Inference time and accuracy results for the proposed early exit models at different SNR levels with $T = 0.6$ and $T = 0.05$.

Training & Inference Procedures

Algorithm 1 Proposed Early Exiting Training

Inputs: Training data X , target labels y , loss function \mathcal{L}
optimizer \mathcal{O} , number of epochs E

Outputs: Trained network parameters θ_1, θ_2

// θ_1 : Forward Pass 1 layers

// θ_2 : Forward Pass 2 layers except for the common layers

procedure TRAINING(X, y)

 Initialize network parameters θ_1, θ_2

for epoch=1 $\rightarrow E$ **do**

for batch in X **do**

 Forward Pass 1: compute network output \hat{y}_1

 Forward Pass 2: compute network output \hat{y}_2

 Compute Loss 1: $loss_1 \leftarrow \mathcal{L}(\hat{y}_1, y)$

 Compute gradients 1: $\nabla_1 \leftarrow \nabla_{\theta_1} \mathcal{L}(loss_1)$

 Update parameters $\theta_1 \leftarrow \mathcal{O}(\theta_1, \nabla_1)$

 Compute Loss 2: $loss_2 \leftarrow \mathcal{L}(\hat{y}_2, y)$

 Compute gradients 2: $\nabla_2 \leftarrow \nabla_{\theta_2} \mathcal{L}(loss_2)$

 Update parameters $\theta_2 \leftarrow \mathcal{O}(\theta_2, \nabla_2)$

end for

end for

return θ_1, θ_2

end procedure

Algorithm 2 Proposed Inference Algorithm

Inputs: Data X , Confidence threshold T

Outputs: Predicted labels \hat{y}

procedure INFERENCE(X, T)

for sample in X **do**

 Forward Pass 1: compute network output \hat{z}_1

 Common layer's output Q is saved

if $entropy(z_1) < T$ **then**

append(\hat{y}, z_1)

else

 Forward Pass 2 begins with Q

 Forward Pass 2: compute network output \hat{z}_2

append(\hat{y}, z_2)

end if

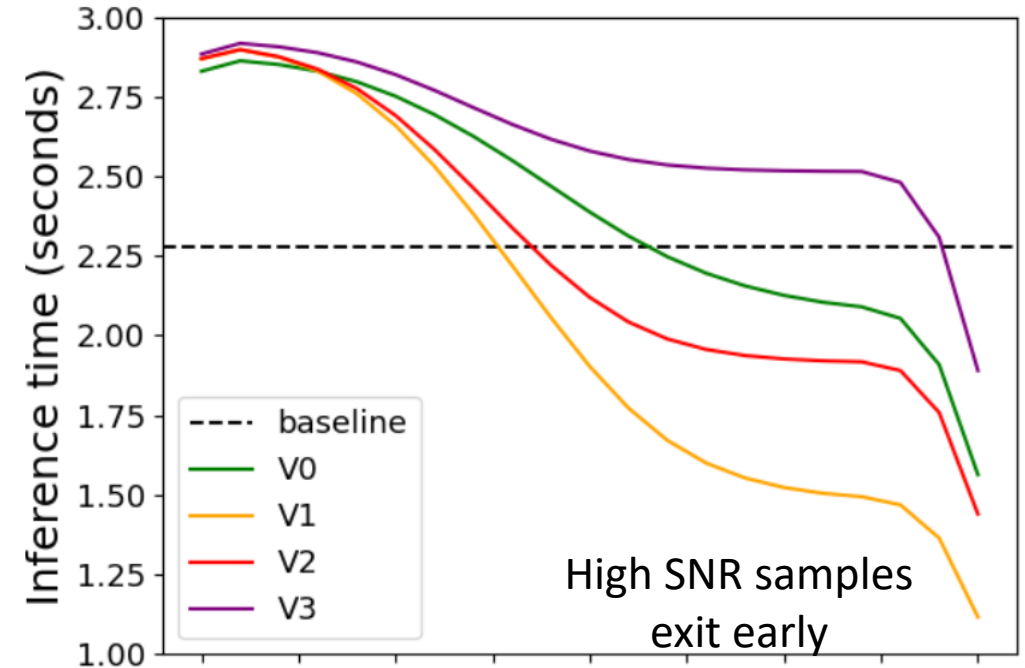
end for

return \hat{y}

end procedure

Key Takeaways

- Results show that $> 50\%$ reductions in inference time are possible for AMC – further improvements are also possible.
- The benefits of exiting early are even more pronounced if most of the time the input stream is easy / does not need the worst case scenario (i.e. the data is not uniform)
- The benefits increase for deeper networks and networks that increase in complexity as the depth increases
- Other parallel architectures are also possible



E. Mohammed, O. Mashaal, and, H. Abou-Zeid "Using Early Exits for Fast Inference in Automatic Modulation Classification," *IEEE GLOBECOM, 2023*

Compressing Deep Neural Networks

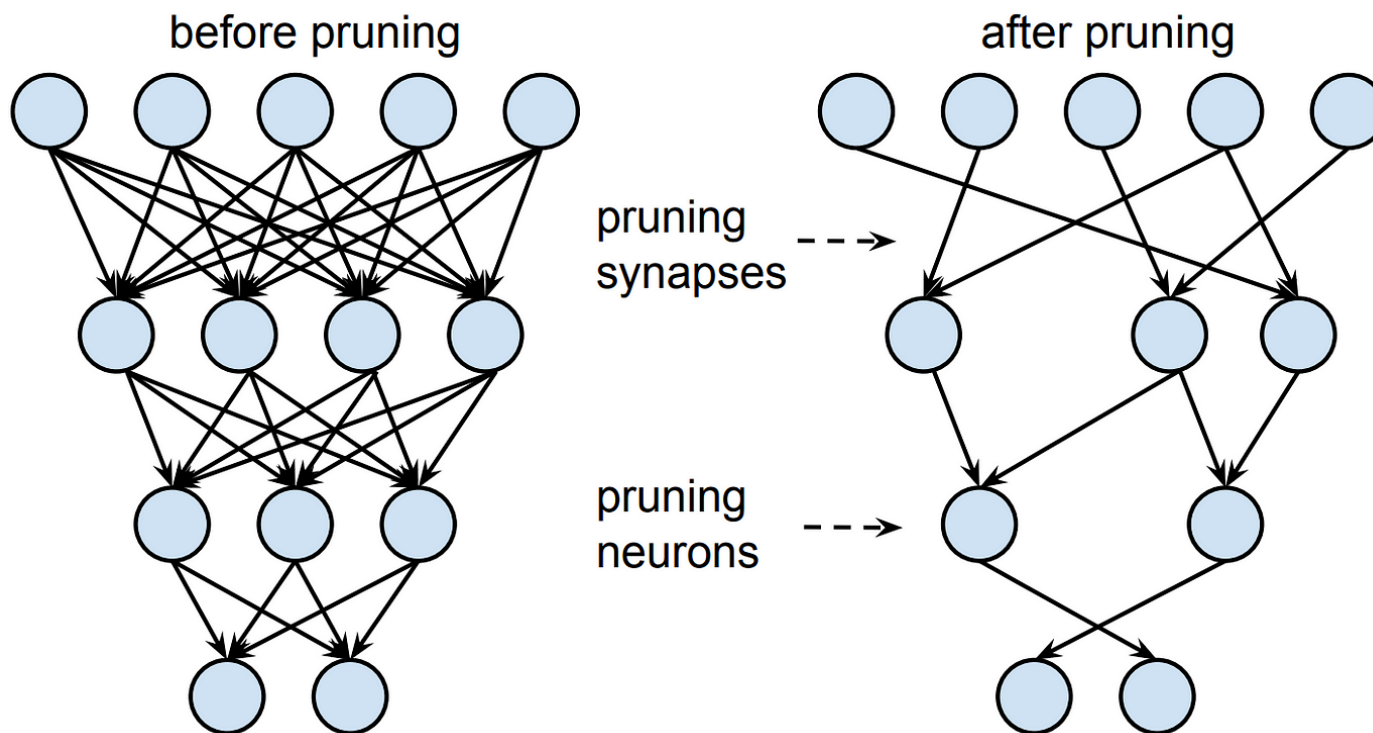
- ⇒ Deep compression enables neural networks to be much more efficient (memory + processing) without compromising accuracy
- ⇒ Advances in this area have demonstrated > 100x compression rates for computer vision applications



Many Deep Compression Techniques

- **1. Magnitude-Based Weight Pruning:** ranks connections in a network according to the absolute values of their weights. A target sparsity ratio is selected and low-weight connections that have minimal impact are set to 0 and then removed to achieve the targeted sparsity ratio.
- **2. Post-Training Quantization:** works by storing weights and activations with a lower precision, for example, weights and activations are stored as 8 bit integers instead of the full precision of 32 bit floating point values.
- Many more advanced techniques!

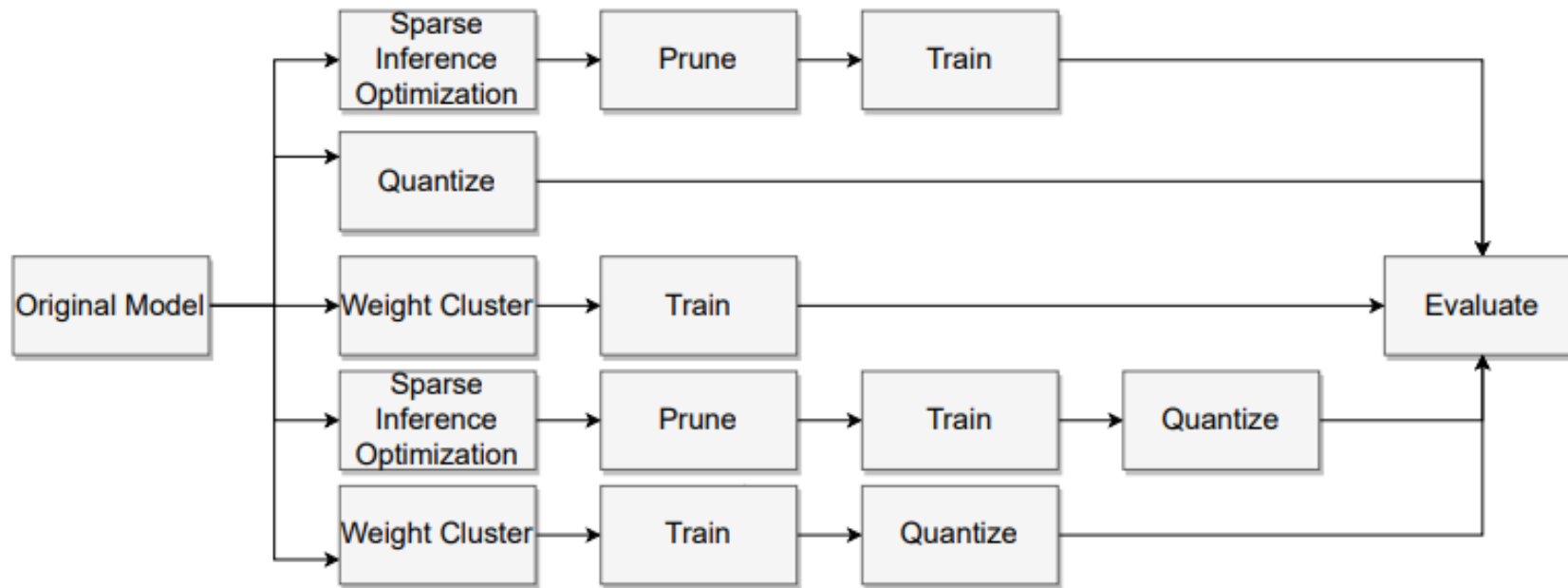
Network Pruning



Han, Song, et al. "Learning both weights and connections for efficient neural network." *Advances in neural information processing systems* 28 (2015).

Compression techniques can be jointly applied or cascaded

⇒ Multiple compression techniques can be designed to be applied together

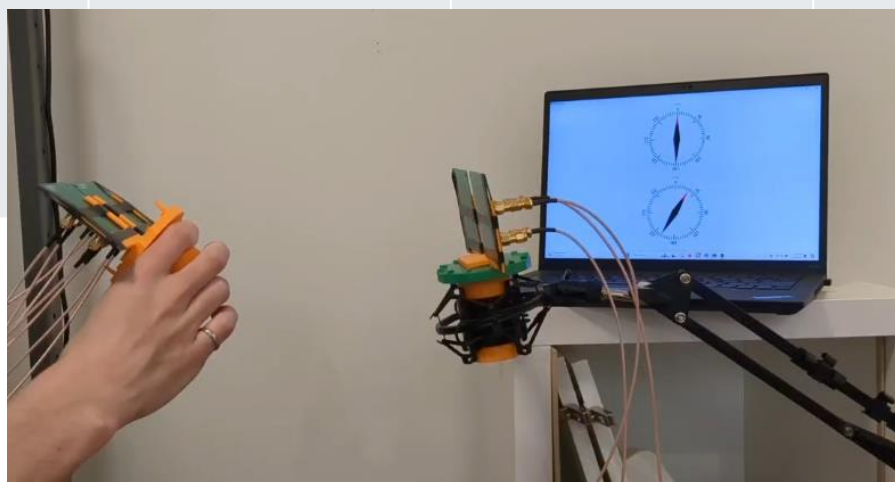


- ✓ Compression gains can be multiplicative
- ✓ Performance is hardware dependent
- ✓ Accelerators are needed to reap the gains

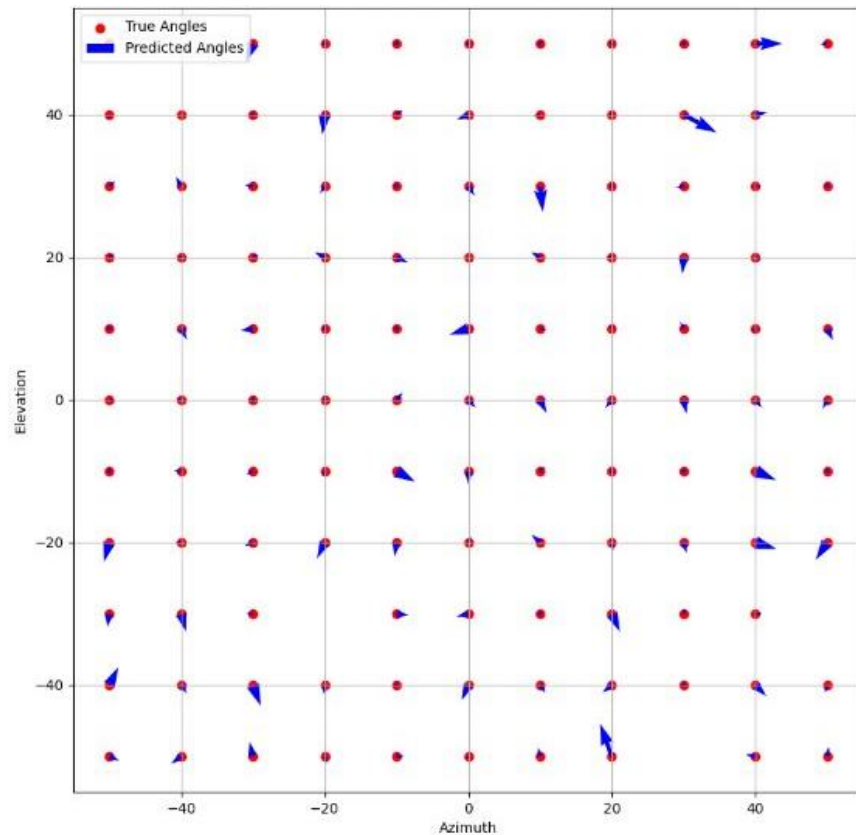


Model Pruning on an AoA Deep Learning Model

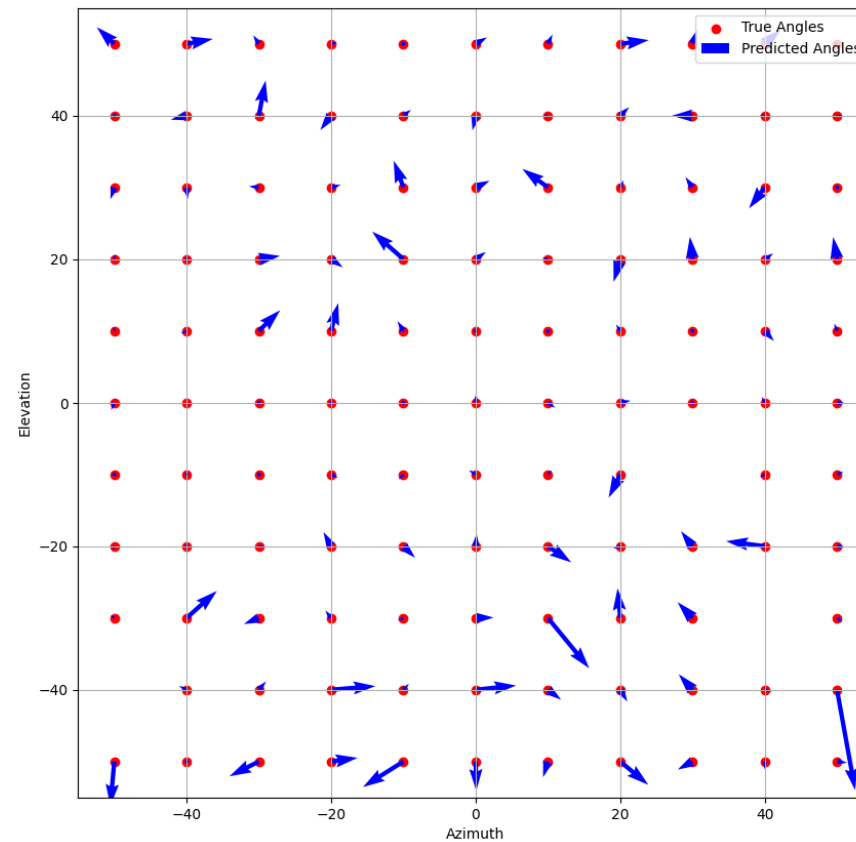
The model	Model size (MB)	GFLOPs (GFLOPs)	Total Params (count)	Inference Time on CPU (s)	Average angle deviation
Original model	2.197	23439648	548482	0.1769	Azimuth =1.55 Elevation =1.16
Pruned model (96% non-uniform structured pruning)	0.167	4221640	20011	0.0495	Azimuth =2.2 Elevation =2.1



Model Pruning on an AoA Deep Learning Model



Quiver plot of Original Model



Quiver plot of Compressed Model

Summary & Future Research

Summary

- We discussed key AI challenges in wireless networks and the vision of Generative Radio Embeddings for Accelerated and Trustworthy (GREAT) AI for Foundation Models in 6G
- We presented a use-case where IQ sample embeddings were learned for Beam Prediction using Prototypical Networks and enabled few-shot domain adaptation to different RF-front ends
- We presented an initial step toward building a spectrogram learning foundation model that uses a masked spectrogram pre-training approach that does not need labeled data and subsequently fine-tuned it to two tasks
- We presented the hypothesis for and application of early-exiting deep learning to wireless communications and deep compression techniques applied to various wireless use-cases

Challenges & Questions

- Availability of real-world wireless data
- The density and size can of the data can be very large, get terabytes quickly at high sampling rates
- Structure of the wireless data is very different at different layers
- Many different functions at the different layers
- What constitutes a foundational model in wireless? What “groups of functions” are best suited for these types of models?
- What are good base models or architectures for different tasks?
- What are good data augmentation and label-free pre-training techniques for wireless?

Opportunities & Future Research

- A lot of incremental and fundamental contributions are likely needed before “foundational models”.
- While AI advances are happening very rapidly and there is a lot to learn from successes in other domains (e.g. vision, language, and robotics) that can be useful in wireless communications
 - Prototypical networks
 - Multi-branch early exiting
 - Self-supervised learning techniques
 - Active learning, continual learning
 - Many others!
- Representation, embeddings, and architectures matter. A lot of research in the NLP and vision community made advances in these areas first before reaching what we have today!

Thank You!

Questions?

hatem.abouzeid@ucalgary.ca