AI Engineering
https://www.sciltp.com/journals/aieng



Check for updates

Review

Foundation Models for Wireless Communication: Fundamentals, Challenges, and Future Directions, with Case Studies in Modulation Signal Constellation Diagram Analysis

Mohammad Rostami [1,†], Atik Faysal [1,†], Reihaneh Gh. Roshan [2], Nikhil Muralidhar [2], Yu-Dong Yao [3] and Huaxia Wang [1,*]

[1] Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ 08028, USA
[2] Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ 07030, USA
[3] Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA
* Correspondence: wanghu@rowan.edu
† These authors contributed equally to this work.

How To Cite: Rostami, M.; Faysal, A.; Roshan, R.G.; et al. Foundation Models for Wireless Communication: Fundamentals, Challenges, and Future Directions, with Case Studies in Modulation Signal Constellation Diagram Analysis. *AI Engineering* **2026**, *2*(2), 23. https://doi.org/10.53941/aieng.2026.100023

Received: 16 March 2026
Revised: 29 October 2026
Accepted: 3 November 2026
Published: 18 November 2026

Abstract: Modern wireless networks must cope with far more devices, and a far wider mix of traffic, than the task-specific models they were built around were designed to handle. Foundation models offer a single large network, pre-trained on unlabeled data with self-supervised objectives, can be reused across many downstream wireless tasks instead of being retrained from scratch for each one. This paper introduces the idea for the wireless setting, with an emphasis on modulation signals generated from their constellation diagrams. We walk through the full pipeline, from what to pre-train on and with which objectives when the inputs are RF signals rather than text or images, to how the pre-trained model is adapted afterward through full fine-tuning, linear probing, or parameter-efficient methods. Two of our own frameworks, NMformer and DenoMAE, serve as concrete case studies of what these choices look like in practice. We give insights about where the models still falls short today, namely its computational cost, the scarcity of labeled wireless data, and the difficulty of adapting in real time. Looking ahead, we argue that lightweight and edge-native architectures, physics-informed synthetic data, and energy-efficient computing are the directions most likely to make foundation models practical in deployed, resource-constrained wireless systems.

Keywords: foundation models; wireless communication; spectrum awareness; interference mitigation; machine learning

1. Introduction

The wireless spectrum has become a victim of its own success. As billions of IoT devices join 5G networks [1] and millimeter-wave bands open for massive MIMO (Multiple Input Multiple Output), spectrum management has evolved from an engineering challenge into an intractable complexity crisis. Modern base stations must simultaneously sense spectrum across multiple bands, classify dozens of modulation schemes, detect and mitigate interference from unknown sources, and adapt in real-time to channel conditions that change millisecond by millisecond. Traditional machine learning approaches buckle under this complexity: each task requires a specialized model, extensive labeled training data, and unreliable assumptions about the operating environment [2].

Yet at precisely this moment of crisis, artificial intelligence has undergone its own revolution. Foundation models are large neural networks pre-trained on huge amounts of unlabeled data, have already reshaped natural language

Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

processing (GPT, BERT) [3] and computer vision (DALL-E, CLIP) [4]. A single such model can be applied to many tasks, adapts with only light fine-tuning, and often shows abilities no one trained it for explicitly.

This paper examines how the foundation model paradigm can be applied in the wireless domain. Rather than training separate models for spectrum sensing, modulation classification, interference detection, and resource allocation, we can pre-train a single foundation model on massive unlabeled wireless datasets: raw I/Q samples, channel state information, constellation diagrams, and then adapt it to diverse downstream tasks with minimal additional training [5].

Wireless networks now generate petabytes of unlabeled data daily [6], providing the raw material for self-supervised pre-training. Transformer architectures, originally designed for language, have proven remarkably effective for the sequential and spatial patterns in wireless signals. Model compression techniques make even large models deployable on resource-constrained edge devices. And emerging 6G standards explicitly call for AI-native network architectures that foundation models are uniquely positioned to enable.

This paper provides a comprehensive roadmap for this emerging field. We explain the fundamentals of foundation models through a wireless lens, detail the development lifecycle from pre-training to deployment, present case studies of working systems (including our own NMformer [7] and DenoMAE [8] frameworks), analyze the technical challenges that remain, and chart research directions toward truly intelligent, autonomous wireless networks. By bridging advances in machine learning with the pressing demands of modern wireless systems, we aim to catalyze a new era of spectrum management, where networks don't just react to their environment but understand it.

This paper is a survey of foundation models for wireless communication with constellation-diagram analysis as its anchor modality and running case study, and its organization reflects that structure deliberately. Sections 2 and 3 develop the foundation-model paradigm, pre-training objectives, data modalities, and adaptation strategies, at the level of the wireless domain as a whole, because these mechanisms are shared across representations: the masking, contrastive, and generative objectives developed for spectrogram inputs transfer directly to constellation-based models, and several surveyed systems share encoders or training objectives with our own frameworks. Section 4 surveys representative systems across modalities and closes with a comparative synthesis that situates constellation-based approaches within this landscape. Section 5 examines the challenges that remain before foundation models can be deployed widely in wireless systems. Section 6 then grounds the general framework in two constellation-centric case studies, NMformer and DenoMAE. Section 7 outlines research directions and potential solutions and Section 8 concludes the paper.

The literature discussed here was curated as follows. We searched databases: arXiv, IEEE Xplore, Google Scholar, using combinations of the terms, "wireless foundation model", "self-supervised", "contrastive learning", "modulation classification", "constellation diagram", "channel representation", over the window 2015–2025, supplemented by backward and forward citation chasing from the retrieved papers. A work was included if it (a) pre-trains a model on wireless signal data with a self-supervised, contrastive, or generative objective; (b) directly informs the adaptation strategies of Section 3; or (c) provides necessary comparative context for constellation-based modulation classification.

2. What Are Foundation Models

Machine learning for wireless systems is usually built one task at a time. A task is fixed first, say spectrum sensing, modulation classification, or beamforming, and then labeled data is gathered for it, a network is trained, and the result is deployed. The next task repeats the cycle with its own dataset and its own model, and when conditions change, a new modulation scheme, a shift in SNR, or a move to a different deployment site, the trained model usually degrades and has to be rebuilt from scratch. Foundation models take a different route. A single large model is pre-trained on large quantities of unlabeled wireless data, such as raw I/Q samples, constellation diagrams, and channel impulse responses across several bands, with no single task in view. What it learns is the structure of the signals themselves: how modulation appears in phase and amplitude, how a channel distorts what was transmitted, how interference shows up in the spectrum, and how patterns develop over time and space.

That pre-trained model then serves as the starting point for everything downstream. For a specific task we do not start over; we adapt the existing model, fine-tuning it on a modest amount of task-specific labeled data. In practice this often matches or beats a conventionally trained model while using far less labeled data and far less training time.

Foundation models share a few properties that make them attractive for wireless work:

- **Transferable representations.** Pre-training on varied unlabeled data yields features that carry over between tasks. A model exposed to constellation diagrams from many modulation schemes picks up what symbol

clustering, phase coherence, and added noise look like, and those same features often help on tasks the model was never trained for.

- **Sample efficiency.** Labeled data has always been the bottleneck in wireless ML, since ground-truth labels for modulation types, interference sources, or channel conditions come from costly measurement campaigns and expert annotation. Self-supervised pre-training sidesteps most of this: the model learns from plentiful unlabeled over-the-air signals and then needs only a small labeled set to specialize.
- **Cross-domain generalization.** Wireless systems span very different environments, from sub-6 GHz cellular to mmWave 5G, Wi-Fi, satellite links, and IoT networks. A model pre-trained across this range tends to carry over between them: having learned the basics of signal propagation, it can move from, say, indoor Wi-Fi to outdoor cellular with little retraining, because what it learned is largely independent of any one environment.
- **Emergent behavior.** The more surprising property is that these models do things they were never explicitly trained to do. Zero-shot prediction lets a model handle a task with no labeled examples by drawing on what it already knows about the domain, and few-shot learning lets it adapt to a new scenario from a handful of examples. Both tend to appear only once pre-training is done at sufficient scale on diverse data.

Why transformers. Several architectures have been tried, but transformers have become the default choice for foundation models, in wireless as much as elsewhere. Their self-attention mechanism suits the long-range dependencies that wireless signals contain. Treating a constellation diagram as a sequence of patches, or an I/Q stream as a sequence of samples, attention lets the model weigh which components interact, how a symbol relates to its neighbors in time, and which frequencies matter for a given decision.

They also scale well. Where a recurrent network processes a sequence step by step, a transformer handles the whole input in parallel, which makes training on large wireless datasets practical. The architecture is also convenient to adapt: one can freeze most of the layers and train only a small task-specific head, or add lightweight adapter modules that specialize the model without disturbing what it already learned.

3. Stages of Developing Foundation Models

Building a wireless foundation model does not look much like a conventional ML pipeline. Instead of the usual train, validate, and deploy loop, the process splits into two stages: a long pre-training stage that builds a broad model of wireless signals, and a shorter adaptation stage that turns that model toward a particular application. The two stages serve different purposes, so it helps to look at each on its own.

3.1. From Pre-Training to Adaptation

The idea at the center of foundation models is to separate learning general representations from learning anything task-specific. Most of the work goes into the first part, using large amounts of unlabeled data to build a general picture of a domain, so that the second part, solving any one problem, becomes cheap.

Pre-training is where the large network, usually a Transformer, is trained on a large unlabeled corpus with a self-supervised objective. For wireless, that corpus is made up of raw in-phase and quadrature (I/Q) samples, channel state information (CSI), channel impulse response (CIR) measurements, or visual renderings of these. The point is not to solve a task but to produce feature representations, or embeddings, that reflect the structure and statistics of wireless signals, learned without any human-annotated labels.

Fine-tuning is the second stage, where the pre-trained model is adapted to a specific downstream task, for instance automatic modulation classification (AMC), beam prediction, user localization, or line-of-sight (LoS) detection. Because it starts from an already-capable model, this stage runs on a much smaller labeled dataset and costs far less in compute and data than training a fresh model for each task.

The reason this split matters so much in wireless is the labeled-data bottleneck. Gathering large, varied, and correctly labeled signal datasets is hard, slow, and expensive, and the problem only grows with the number of modulation schemes, the way channels change over time, and the need to cover a wide SNR range. Pre-training offers a way around it: unlabeled over-the-air signals are plentiful, and a model trained on enough of them can be specialized for demanding tasks from only a small labeled set. The Large Wireless Model (LWM) is a good example, pre-trained on large and diverse wireless data to learn channel embeddings that transfer well across downstream tasks even when labeled examples are scarce [9].

This marks a real change in how wireless signal processing is done. Automatic modulation classification, for one, long depended on hand-crafted features, where experts used domain knowledge to pull out informative quantities such as higher-order statistics (cumulants), cyclostationary features, or likelihood functions. Such methods work in the right conditions but tend to be fragile, sensitive to noise, and dependent on accurate prior knowledge

of the channel, which limits how far they generalize. Early deep learning, especially with Convolutional Neural Networks (CNNs), took over the feature-extraction step but still needed large labeled datasets and supervised training from scratch [10]. An important move in the same direction was casting the physical layer itself as a learning problem, training transmitter and receiver together as an end-to-end autoencoder [11]; this showed that learned representations could replace hand-built signal-processing blocks and set the stage for the data-driven designs that foundation models now carry further. Self-supervised pre-training continues this line by learning useful representations straight from the signals, which removes the need to design features by hand. Models such as LWM [12], the Sparse pretrained Radio Transformer (SpaRTran) [13], and CSI2Vec [14] are built to act as general feature extractors, and their embeddings can be reused across many channel-related tasks. The research question changes accordingly: rather than engineering features for one task, the effort moves to choosing pre-training objectives and assembling datasets that produce the most broadly useful signal representations.

3.2. Stage 1: Self-Supervised Pre-Training Strategies

Pre-training is where most of the work happens, and where the model picks up the structure of wireless signals. Two choices shape the result more than anything else: how the signal is presented to the model, and which self-supervised objective is used to train it. Both are domain-specific, and both strongly affect how good the learned representations turn out to be.

3.2.1. Data Modalities and Representation for Pre-Training

The first decision is how to present the signal. Each modality exposes a different side of it, so the right choice depends on what the model will eventually be used for.

- **Raw I/Q Sequences:** The most basic representation is the signal itself: the time series of complex baseband samples, with its in-phase (I) and quadrature (Q) components. Training on raw I/Q lets the model work with the native temporal structure, including amplitude, phase, and frequency, and avoids whatever information is lost when the signal is converted to another domain. RIS-MAE [15] and IQFM [16] both take this route, building their representations straight from the time-domain waveform.
- **2D Visual Representations:** A common alternative is to turn the one-dimensional time series into a two-dimensional, image-like format [17]. The appeal is practical: it lets one reuse mature computer-vision architectures such as CNNs and Vision Transformers (ViTs), which are already well tuned for spatial features.
  - **Constellation Diagrams:** For modulation analysis in particular, the constellation diagram is hard to beat. It is a 2D scatter plot of the received I/Q samples, and its geometry, how the points cluster, how far the symbols scatter, the overall shape, says a great deal about which modulation scheme produced it. It also shows channel impairments directly, since noise, phase offset, and I/Q imbalance all distort the plot in visible ways, which makes it a natural input for vision-based models. DenoMAE2.0 [18], for example, is pre-trained on large unlabeled sets of constellation diagrams to learn representations for AMC.
  - **Other Visual Formats:** Constellation diagrams are not the only option. Spectrograms (time-frequency diagrams) [19] and bispectrum diagrams [20] are also used, and they capture things a constellation diagram does not, such as how frequency content evolves over time, which can complement its phase and amplitude information.
- **Multi-Modal Channel Information:** Some tasks call for a fuller picture of the channel than any single view provides, and that picture can be assembled from several data sources that line up naturally with one another. This is the motivation behind multi-modal foundation models [8], which learn a joint representation across different data types.
  - **CSI and CIR Fusion:** CSI-CLIP [21] is a clear example. It treats Channel State Information (CSI), the channel's frequency response, and the Channel Impulse Response (CIR), its time-domain counterpart, as a naturally aligned pair, and uses contrastive learning to match each CSI with its corresponding CIR. Learning this correspondence gives the model a fuller sense of the channel, much as pairing images with text does in vision-language models like CLIP [22], and it helps on downstream tasks such as localization and beam management.
  - **Fusing Network and Spectrum Data:** Other work combines spectrum-sensing data with network traffic, so a model can relate what happens at the physical layer to higher-layer behavior and decide accordingly. Cross-modal attention can go further and fuse quite different sources, such as images and action sequences, for robotics and control systems that operate in a wireless environment [23].

3.2.2. Reconstructive Learning: Masked Signal Modeling (MSM)

Masked Signal Modeling (MSM) [24] is one of the most effective and scalable self-supervised objectives for this setting. The idea comes from Masked Autoencoders (MAEs) [25] in vision and masked language models such as BERT [3] in NLP: hide part of the input and train the model to fill it back in.

Concretely, a large fraction of the input is masked, say a set of patches in a constellation diagram or contiguous stretches of an I/Q sequence, and the model has to predict what was removed. Simple as it sounds, the task cannot be solved by pattern matching alone; to fill in the gaps the model has to represent the dependencies and the rules that govern how the signal is put together.

Several MSM frameworks have been built specifically for wireless:

- **WiMAE (Wireless Masked Autoencoder):** WiMAE [26] is a transformer pre-trained on a large, realistic dataset of multi-antenna channels. It masks a high fraction of the input (around 60%), which forces the model to recover a lot from little visible context, and it uses an asymmetric encoder-decoder in which the heavy encoder sees only the unmasked patches. That design keeps pre-training cheap and pushes the encoder toward abstract features rather than low-level memorization; a light decoder then reconstructs the masked regions from those features.
- **DenoMAE2.0:** DenoMAE2.0 [18] adapts the standard MAE to the noise and imperfection of real wireless signals, especially constellation diagrams. It adds a second training objective alongside patch reconstruction: a local patch classification task in which each unmasked patch is labeled by its position in the diagram and the model learns to predict that position. With both objectives active, the model has to capture global coherence from reconstruction and position-aware local detail from classification at the same time. That combination pays off on noisy constellations, where both the overall shape and the exact location of the symbol clusters matter for getting the classification right.

There is more going on here than filling in gaps. In reconstructing a signal, the model ends up learning the physical and statistical constraints behind it. A received signal is not arbitrary; its structure follows from communication theory and from the physics of how electromagnetic waves propagate, and reconstruction pushes the model to internalize those rules. To restore a missing patch of a 16-QAM constellation, for instance, the model cannot just interpolate neighboring pixels. It has to learn that valid symbols sit on a particular grid, the modulation alphabet, and it has to account for the noise and impairments that scatter those symbols away from their ideal positions.

3.2.3. Contrastive Learning for Robust Signal Embeddings

Contrastive learning offers a different route. Rather than reconstruct the input, it shapes an embedding space in which similar signals land close together and dissimilar ones end up far apart. The encoder is trained to agree across two "views" of the same sample, a positive pair, while disagreeing with views drawn from other samples, the negative pairs.

In practice it has three parts:

- **Data Augmentation:** A stochastic data augmentation module that generates two correlated views of a given signal, x_i and x_j, which form a positive pair. These augmentations should transform the signal in ways that preserve its semantic identity.
- **Encoder Network:** A neural network that extracts representation vectors (embeddings) from the augmented data samples.
- **Contrastive Loss Function:** A loss function, such as InfoNCE, that pulls the representations of positive pairs (h_i, h_j) together in the embedding space while pushing them apart from the representations of all other samples in the batch (negative pairs).

Over training, this makes the representations invariant to whatever augmentations were applied: the model keeps the features that define a signal's identity and discards nuisance variation such as noise or small channel fluctuations.

Several contrastive frameworks have been adapted to wireless signals:

- **FLA-CL (Feature-Level Augmented Contrastive Learning):** FLA-CL [27] augments not only the input but also the high-dimensional feature space. The extra augmentation produces more varied and harder positive and negative pairs, which in turn yields more discriminative representations and better downstream classification.
- **Domain-Invariant RF Fingerprinting:** Contrastive learning is also well suited to domain shift, a recurring problem in Radio Frequency (RF) fingerprinting where channel conditions drift between training and test time.

CrossRF [28] treats signals from the same transmission, captured at different times or under slightly different channels, as positive pairs, and signals from different transmissions as negative pairs. Training this way encourages features that ignore channel effects, giving a device fingerprint that holds up across environments.

- **ContraWiMAE:** ContraWiMAE [26] shows what happens when the two objectives are combined. Starting from WiMAE's pre-trained weights, it adds a contrastive head, builds positive pairs by adding white Gaussian noise (AWGN) to a channel realization, and contrasts them against samples from other realizations. The reconstruction objective keeps the detailed structure of the channel while the contrastive one supplies the invariant, discriminative features needed for classification. The result is a more linearly separable embedding and better downstream performance than either objective gives on its own.

3.2.4. Generative Pre-Training and Augmentation

A third family, generative models, learns the probability distribution of the training data rather than a discriminative embedding. The main examples are Generative Adversarial Networks (GANs) [29] and Variational Autoencoders (VAEs) [30]. A GAN pits two networks against each other: a generator that turns random noise into synthetic samples, and a discriminator that tries to tell real samples from generated ones. As training goes on, the generator gets better at producing samples the discriminator can no longer distinguish from real data.

Table 1 compares the three families of pre-training objectives discussed in this section in terms of their core mechanism, their main strengths and limitations for wireless signals, and representative models.

Table 1. Comparison of Self-Supervised Pre-training Objectives for Wireless Foundation Models.

Objective Type	Core Mechanism	Key Strengths for Wireless	Potential Limitations	Example Models
Reconstructive (MSM)	Randomly mask portions of the input signal (e.g., constellation patches, I/Q segments) and train the model to reconstruct the missing parts.	• Excellent for learning holistic spatial/temporal structure • Highly scalable and computationally efficient (e.g., asymmetric encoder) • Implicitly learns signal physics and statistical constraints	May underperform in tasks requiring fine-grained discriminative features compared to contrastive methods.	WiMAE [26], DenoMAE2.0 [18], RIS-MAE [15]
Contrastive	Learn an embedding space where augmented versions of the same signal (positive pairs) are close, and different signals (negative pairs) are far apart.	• Learns representations that are invariant to nuisance variations • Excels at producing linearly separable features for classification • Highly effective for domain adaptation and handling domain shift	Performance is highly dependent on the quality and relevance of the data augmentations used.	FLA-CL [27], ContraWiMAE [26], CSI-CLIP [21]
Generative	Learn the underlying probability distribution of the training data to generate new, synthetic samples (e.g., using GANs).	• Unlocks data augmentation for rare signal types or conditions • Enables large-scale, data-driven simulation and testing • Can be made physics-informed to ensure scientific validity	• Training can be unstable (GANs) • Ensuring physical plausibility of generated wireless signals is a significant challenge	GANs for OFDM Simulation, Physics-based Generative Models [31,32]

For wireless signal analysis, a GAN trained on real constellation diagrams can produce new, synthetic ones, which helps in two main ways:

- **Data Augmentation:** For unusual or proprietary modulation schemes, or for signals under rare channel conditions, collecting enough real data is often impractical. A GAN can supply as many realistic examples as needed, which helps fill out a dataset, balance classes that are underrepresented, and make downstream classifiers more robust.

Transfer learning can be applied to the GAN itself so that it still produces good constellation images from a very small seed set, which also helps avoid familiar failure modes like mode collapse.

- **Simulation and Interference Testing:** Generative models can also synthesize full, realistic waveforms, such as Orthogonal Frequency-Division Multiplexing (OFDM) signals. That makes it possible to build data-driven simulation environments for system-level evaluation and interference testing that are often more realistic than purely statistical models and cheaper than large measurement campaigns.

3.3. Stage 2: Adaptation and Fine-Tuning for Downstream Tasks

The most straightforward adaptation methods involve either updating all the model's parameters or just a small fraction of them.

Full Fine-Tuning: Here all or most of the pre-trained weights are unfrozen and training continues, by backpropagation, on the labeled downstream set. The learning rate is usually one or two orders of magnitude smaller than in pre-training, so the model specializes its existing features without overwriting what it already knows. One place this works especially well in wireless is domain adaptation. A model can be pre-trained on high-SNR signals, where the modulation features are clean, and then fine-tuned on a smaller set of low-SNR signals; carrying knowledge from the clean domain into the noisy one improves AMC under realistic conditions far more than training on the noisy data alone.

Linear Probing: Linear probing takes the opposite stance. The pre-trained model is frozen and used only as an encoder; a single linear classifier is attached on top, and only that head is trained on the downstream labels. It is cheap to run and doubles as a diagnostic, because how well a linear head does is a direct read on how separable the pre-trained representations already are. High accuracy from a linear probe means the pre-training objective has organized the embedding space so the classes are largely separated to begin with. Wireless foundation models have posted strong linear-probing results on tasks like localization, which speaks to the quality of their self-supervised features.

Parameter-Efficient Fine-Tuning (PEFT): PEFT methods cut the compute and storage cost of full fine-tuning by freezing most of the pre-trained parameters and adding only a few task-specific ones. The best-known is Low-Rank Adaptation (LoRA) [33], which rests on the observation that the weight update during adaptation has low intrinsic rank. The change matrix ΔW can therefore be approximated as a product of two smaller matrices, $A \in \mathbb{R}^{r\times d}$ and $B \in \mathbb{R}^{d\times r}$ with $r \ll d$, so the adapted weights become $W' = W_0 + BA$ and the number of trainable parameters drops from d^2 to $2dr$. In practice LoRA adds parallel adapter branches to chosen layers, usually the self-attention blocks, a down-projection A followed by an up-projection B, while the original weights W_0 stay frozen.

In wireless this makes multi-task deployment cheap: one foundation model can serve AMC across standards, beam prediction across frequency bands, or localization across environments, with only a small set of adapter weights stored per task. LoRA is not a free lunch, though; it can trail full fine-tuning in harder settings such as federated learning with non-IID data, which is what motivated variants like SLoRA that use data-driven initialization to close the gap.

One reasonable worry about any unified model is whether serving many tasks at once causes catastrophic interference. The setup described here mostly avoids that. With a frozen shared backbone and one small adapter per task, the task-specific parameters are kept separate, so adding or updating a task cannot hurt the others, and the backbone's pre-trained representations stay intact. Interference only really arises when tasks share a single trainable head, where conflicting gradients can cause negative transfer; the usual fixes are gradient balancing and loss weighting, or routing tasks to separate adapters. The paradigm thus trades joint multi-task optimization for modular, additive specialization, which is the safer choice for the mixed and shifting workloads wireless systems have to handle.

Table 2 summarizes each adaptation strategy in terms of trainable parameters, data requirements, computational cost, and the case study.

Table 2. Overview of Adaptation Strategies for Wireless Foundation Models.

Strategy	Trainable Parameters	Data Requirement	Computational Cost	Key Advantage	Ideal Wireless Use Case
Full Fine-Tuning	All/Most	Medium to High	High	Achieves maximum task-specific performance; deep specialization.	Adapting a model to a single, critical downstream task, such as improving AMC in very low-SNR conditions.

Table 2. *Cont.*

Strategy	Trainable Parameters	Data Requirement	Computational Cost	Key Advantage	Ideal Wireless Use Case
Linear Probing	Very Low (Classifier only)	Low to Medium	Very Low	Fast and efficient evaluation of pre-trained representation quality.	As a diagnostic tool to assess the linear separability of learned features from a new pre-training method.
PEFT (e.g., LoRA)	Very Low (∼0.1–1%)	Low to Medium	Low (for fine-tuning)	Highly efficient multi-task deployment; preserves pre-trained knowledge.	Adapting one large channel model to many tasks (e.g., beam prediction for different base stations, AMC for different standards).
Meta-Learning	All (during meta-training)	Very Low (per task)	High (for meta-training)	Enables rapid adaptation to new, unseen tasks from very few examples.	Few-shot AMC [34] in cognitive radio to identify novel or non-standard modulation schemes on the fly.

4. Foundation Models in Wireless Communication

A number of foundation models have already been built for wireless communication, aimed at tasks like spectrum awareness, interference mitigation, and modulation recognition. We go through the most notable ones below and what each one contributes.

4.1. DSINet

DSINet [4] is a multitask deep network for spectrum sensing that handles signal classification, spectrum-hole detection, and modulation classification at once. Instead of training a separate model per task, it reads spectrum usage jointly across time, frequency, power, and code, which helps spectrum awareness in 5G and 6G networks. Across all three tasks it reports higher accuracy than single-task baselines while staying considerably smaller and faster, with a 65.5% reduction in model size and a 230% gain in computational efficiency, enough to make it practical to deploy.

4.2. Transformer-Empowered 6G Intelligent Networks

Wang et al. [35] apply transformer architectures to a range of wireless problems. The design copes well with the high-dimensional data of massive MIMO systems and holds up across varied channel conditions. Covering channel estimation, resource allocation, and semantic communication, it targets ultra-reliable low-latency communication (URLLC) and transfer across domains, which the authors put forward as a basis for intelligent 6G systems.

4.3. Large Wireless Model (LWM)

LWM [12] is a task-agnostic model that produces general, contextualized channel embeddings. Its transformer backbone is pre-trained self-supervised on large-scale wireless-channel datasets, so it picks up the patterns of propagation and network dynamics and carries over to new environments even when task-specific data are scarce. The same embeddings serve channel prediction, interference management, and spectrum analysis, which is a step toward a single adaptable model for next-generation networks.

4.4. Enhanced WaveNet for RF Signal Separation

The enhanced WaveNet [12] reworks the original WaveNet for radio-frequency (RF) signal separation. Learnable dilation parameters in the convolutional layers let the model adjust its receptive field to the signal at hand, and new data-augmentation schemes help it pull apart overlapping sources in a dense spectrum. It raises the Signal-to-Interference-plus-Noise Ratio (SINR) by 58.82% at a Bit Error Rate (BER) of 10^{-3}, well above the usual benchmarks, and took first place in the MIT RF Signal Separation Challenge.

4.5. NextGen Communications Copilot

NextGen Communications Copilot extends Copilot [36] with domain-specific foundation models to act as a conversational assistant for wireless communication. It draws on context extraction, user feedback, and domain databases to answer questions, and is meant for synthesizing communication protocols, troubleshooting on the fly,

and suggesting design choices for wireless systems. On these specialized tasks it scores higher than a general-purpose assistant such as ChatGPT, as measured by BLEU and BERTScore.

4.6. 6G-Transformers

6G-Transformers [37] use a Vision Transformer (ViT) backbone for spectrogram learning in next-generation networks. Pre-training follows Masked Spectrogram Modeling (MSM), a self-supervised scheme that learns representations from large unlabeled datasets. On Channel State Information (CSI)-based human activity sensing and on spectrogram segmentation, the pre-trained ViT matches supervised training and, notably, beats larger models trained from scratch. The authors also point to uses in cross-layer optimization, environmental awareness, and distributed learning, and to applications such as holographic communication and digital twins in 6G.

4.7. Supporting Research

Beyond these systems, a broader body of work fills in the picture. Data-driven channel modeling has been studied as a way to tune future communication systems, and other work looks at noise uncertainty in OFDM signals and at the particular difficulties of IoT networks. Taken together, this research shows foundation models gaining ground in wireless, but it also keeps the open problems in view: computational overhead, data scarcity, and fitting these models alongside legacy systems.

5. Challenges and Key Problems

Foundation models change how wireless problems get solved, since one model can cover many tasks and adjust to changing conditions. That generality comes with its own set of problems, though, and the ones below have to be worked out before the approach can be deployed widely.

- **Data Challenges:** Wireless systems need large, varied datasets to cover the range of conditions they actually meet: different channels, interference patterns, and mobility. Gathering that data is slow and expensive, and for many tasks, spectrum sensing or interference classification among them, there are few or no labels to train on in the first place [38]. Domain adaptation is a related difficulty, since a model trained in one environment often does poorly in another when the hardware, location, or operating conditions differ. We address this with several mechanisms rather than one: (i) self-supervised pre-training on heterogeneous, multi-band data, which pushes the encoder toward domain-invariant structure instead of environment-specific shortcuts; (ii) contrastive objectives that enforce invariance to channel and device variation; (iii) parameter-efficient per-domain adaptation, where a frozen backbone is specialized with small LoRA [33] adapters at little storage cost; and (iv) high-to-low-SNR transfer fine-tuning, which recovers accuracy in degraded regimes without retraining from scratch.
- **Sim-to-Real and Hardware Impairments:** A major deployment obstacle is the gap between the clean, simulated or controlled data used in pre-training and what a signal actually looks like over the air. Real transceivers add impairments that idealized simulators rarely reproduce in full: power-amplifier nonlinearity, IQ imbalance, phase noise, carrier-frequency and sampling-clock offsets, and finite-resolution quantization, all of which bend the constellation geometry that vision-based models rely on. On top of that, deployed systems face adversarial conditions like jamming, whose energy and detection signatures recent surveys have catalogued in detail [39]. Self-supervised pre-training narrows the gap, because the encoder can learn from plentiful unlabeled over-the-air signals rather than synthetic data alone; whatever distribution shift is left is then handled by light domain-adaptation fine-tuning on a small labeled real-world set, and by physics-informed augmentation that injects realistic hardware effects during pre-training.
- **Model Complexity:** Foundation models are large, and that size costs compute at both training and inference time. The problem is sharpest on resource-constrained edge devices, which carry much of the load in wireless applications, and in latency-sensitive tasks like beamforming and scheduling, where meeting real-time deadlines takes a lot of optimization. The cost splits into two parts that behave differently. Storage and adaptation are mostly handled by PEFT: LoRA-style adapters add only about 0.1–1% trainable parameters per task, so many downstream behaviors can live on one device by swapping small adapter weights over a frozen backbone. Inference is the harder part, since it is dominated by the shared backbone and sets the real-time ceiling: adaptive modulation can tolerate tens of milliseconds of latency, but closed-loop beamforming has a sub-millisecond budget, below what an unoptimized transformer encoder currently reaches on edge hardware. Closing that gap depends on inference-time methods (quantization, structured pruning, knowledge distillation,

and hardware-aware NAS), and a careful on-device latency–energy study of the compressed models is left as explicit future work.

- **Interpretability and Explainability:** The black-box character of these models is a real problem in safety-critical wireless settings, where decisions need to be auditable, both for debugging and for meeting regulatory requirements. If a resource-allocation or interference-management decision cannot be explained, it is hard to trust and hard to get approved in a regulated industry. Our own frameworks offer some post-hoc transparency: the ViT self-attention maps in NMformer show which parts of the constellation drive a classification, and the per-patch reconstruction errors in DenoMAE indicate where the denoiser is least confident. Even so, we treat genuinely interpretable wireless foundation models, attention analysis combined with physics-based constraints, as an open problem to settle before these models are trusted with regulated, safety-critical decisions.
- **Real-Time Adaptation:** Wireless channels and user mobility change quickly, so a model has to keep up in real time, without stopping to retrain, which is hard to do well. Tasks such as adaptive power control and real-time beamforming push this further, demanding latencies that current foundation models struggle to hit.
- **Standardization and Integration:** To be adopted, foundation models have to fit the existing standards, 3GPP and IEEE among them, and reaching interoperability without breaking compliance is not straightforward. Fitting them into legacy systems is a further bottleneck, given the hardware and software constraints already in place.
- **Energy Efficiency:** Wireless edge devices usually run under tight power budgets, which makes heavy models awkward to deploy, and training and running foundation models draws enough energy to raise real sustainability concerns.
- **Ethical and Regulatory Issues:** Bias in the training data can lead to uneven performance across regions and demographics, which matters a lot for systems meant to work globally. And because the decision process is opaque, getting regulatory approval for AI-driven wireless solutions is correspondingly harder.
- **Task Specialization:** Foundation models are built to generalize, yet many wireless tasks, channel estimation, modulation classification, resource allocation, still want a specialized solution. Getting the balance between broad generalization and task-specific accuracy right is an ongoing research question.
- **Cost of Implementation:** Training foundation models for new wireless use cases, such as those expected in 6G, is very expensive, and putting them into service often calls for hardware upgrades that existing infrastructure may not be able to accommodate.

6. Case Study: Reproducible Foundation Models

Foundation models like BERT, GPT, and T5 are now central tools in AI [3,40], but reproducing them is hard: the technical and resource demands often stand in the way of independent validation. This case study looks at those demands through our own work on transformer-based denoising during pre-training, and suggests some practical ways to keep results reproducible without giving up performance. We focus on NMformer and DenoMAE for a simple reason, that we have their full training pipelines, data, and configurations on hand and can therefore report the experimental details in full.

6.1. Methodology

6.1.1. NMformer

NMformer (Figure 1) is a Vision Transformer (ViT) model for automatic modulation classification (AMC). Rather than feed it raw modulated signals, we first turn each signal into an RGB constellation diagram, which is the form a ViT can work with. The image is split into patches, and self-attention over those patches extracts the features that distinguish one modulation from another; a multi-layer perceptron (MLP) head then maps the result to a modulation class. Training happens in two stages. We first build a base classifier on a large set of modulation images, then fine-tune on a much smaller set to match specific SNR conditions, freezing every transformer layer except the final linear one so that adaptation needs very little extra data. Because the ViT attends across the whole image, the model captures global structure in the constellation, which helps it classify under a range of channel conditions.

Table 3 reports how NMformer does across SNRs. The base classifier is accurate on the SNRs it was trained on but drops off on noise levels it has not seen. Fine-tuning on only a few samples changes this, especially at low SNR, where it clearly beats the base model. The fine-tuned classifier is more accurate on both in- and out-of-distribution data, and the gains show up in precision, recall, and F1 alike. In short, it generalizes past its training distribution well enough to be useful on real modulation data.

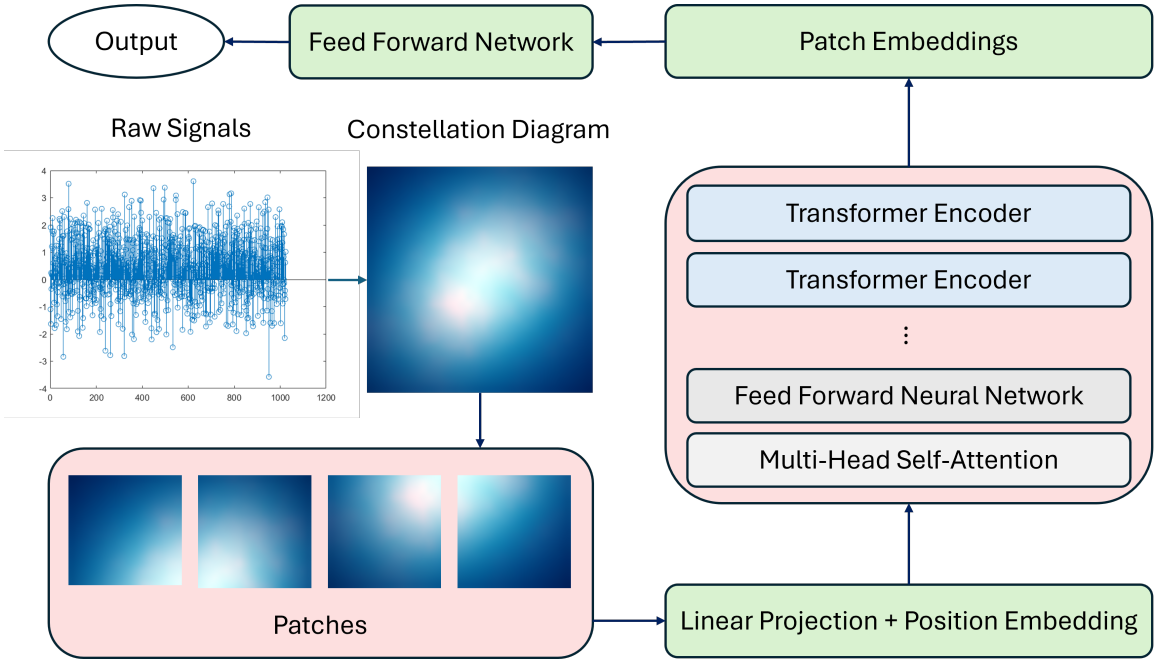

Figure 1. NMformer system sequential pipeline: starting with generating constellation diagrams from raw signals and culminating in modulation classification using a ViT architecture.

Table 3. Performance of the classifier on different settings.

Model		**Base**		**Fine-Tuned**	
Sample		106,800		3000	
		Accuracy (%)		**Accuracy (%)**	
Distribution Type		**In**	**Out**	**In**	**Out**
SNRs	0 dB, 4 dB, 10 dB	71.10	-	-	-
	0.5 dB, 1.5 dB, 4.5 dB	-	67.48	71.60	70.51
	5.5 dB, 7.5 dB, 9.5 dB	-	72.12	75.77	71.87

6.1.2. DenoMAE

DenoMAE (Figure 2) is a multimodal autoencoder that denoises modulation signals within a masked-autoencoder framework. Its distinguishing choice is to treat noise as an explicit input modality, which lets the model learn across modalities and hold up better as noise varies. The architecture is an encoder-decoder: the input signals, again as constellation diagrams, have 75% of their patches masked during pre-training, a shared transformer encoder processes the visible patches, and a separate decoder per modality reconstructs the missing ones. Training minimizes the mean squared error (MSE) of the reconstruction across all modalities, which forces a shared latent representation that transfers between them. For AMC, we reuse the pre-trained encoder and fine-tune it, which needs far fewer labeled samples, and the result is cleaner signals and steady accuracy across signal-to-noise ratios.

DenoMAE denoises and classifies well across SNRs (Figure 3). It holds up in heavy noise, reaching 77.50% accuracy at -10 dB, and improves steadily as the SNR rises, up to 84.30% at 10 dB, which reflects the quality of its pre-trained representations. Against non-pre-trained baselines the margin is as large as 22.1% in the hardest conditions. Taken together, these numbers show that the masked multimodal objective does denoise the constellation diagrams and lets the model classify accurately from very little labeled data.

A word on the scope of these experiments. The results above track performance mainly against SNR under additive noise, which isolates denoising and classification cleanly but does not yet cover the full range of real channel impairments. One reason to expect the approach to extend is that the constellation diagram already encodes several of those effects: multipath fading shows up as cluster dispersion and rotation, carrier-frequency offset and Doppler as coherent rotation and smearing, and co-channel interference as extra spurious clusters. The masked-reconstruction objective that learns to remove additive noise should be able to learn to normalize these as well, and generating such conditions through physics-informed data is a natural next step.

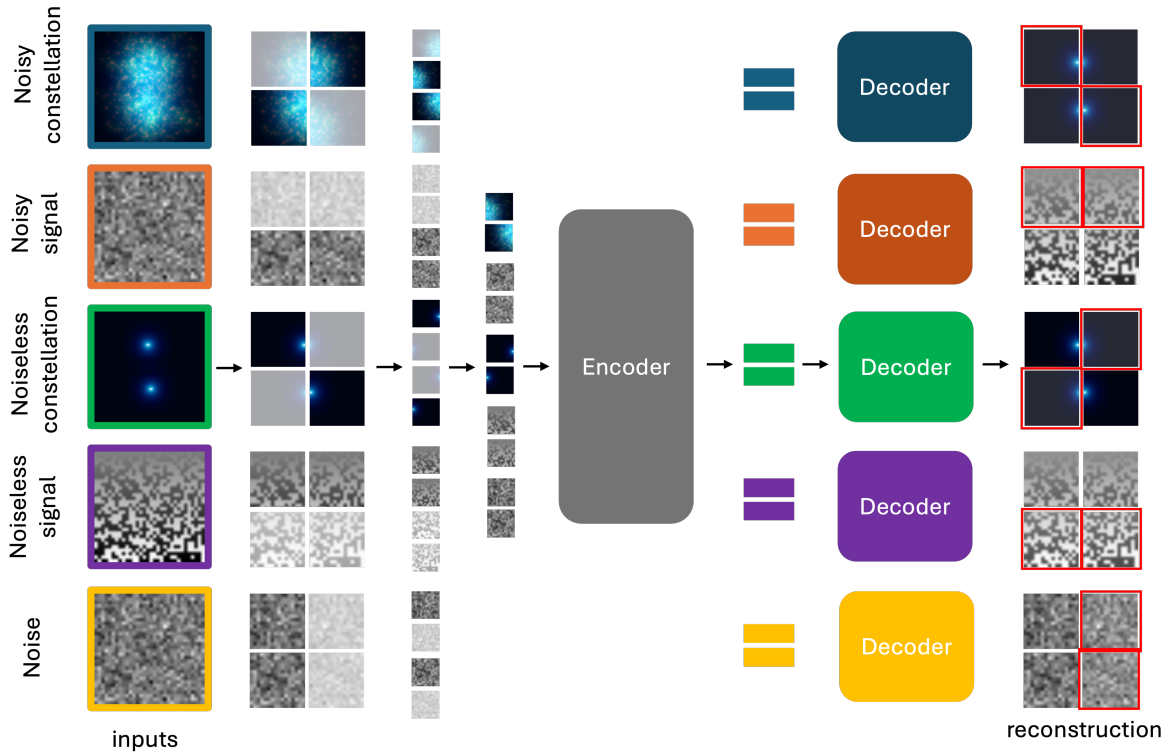

Figure 2. DenoMAE Pretraining Strategy: We apply a random 75% masking (not to scale in the illustration) across all input modalities. The remaining 25% of visible patches are processed by a shared encoder, while each modality utilizes a dedicated decoder to reconstruct its masked patches. Only the encoder is reused for fine-tuning the downstream tasks.

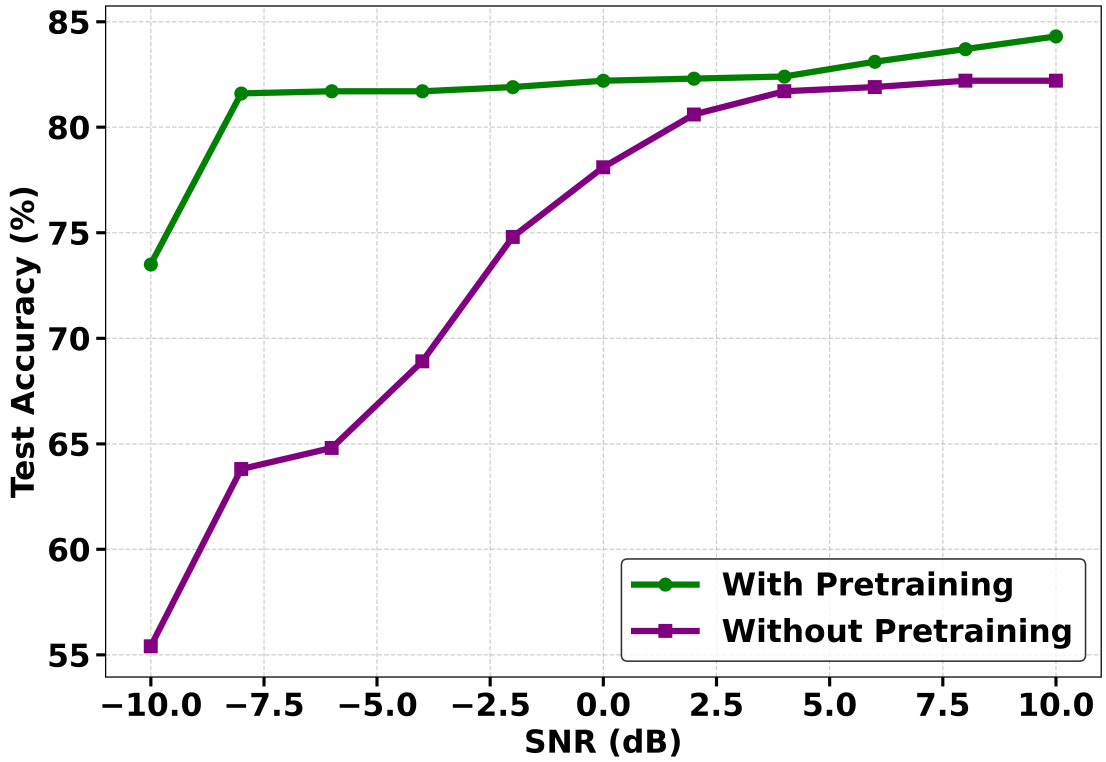

Figure 3. DenoMAE fine-tuned classification accuracy for constellation diagrams at different SNRs.

Table 4 summarizes the experimental configuration of both frameworks; complete implementations are publicly available.

Table 4. Experimental configuration of the case-study frameworks, NMformer and DenoMAE.

Item	NMformer	DenoMAE
Data	Synthetic signals to RGB constellation diagrams (exponential-decay enhancement); SNR: 0–10 dB	Same constellation pipeline, 5 paired modalities (noisy/clean image, noisy/clean signal, noise); SNR: -10 to 10 dB
Classes	10 modulation formats (4/8ASK, 4/16PAM, CPFSK, DQPSK, GFSK, GMSK, OOK, OQPSK), class-balanced	Same 10 formats, class-balanced
Samples & splits	106,800 base/3000 fine-tune; val. 3000; in- and out-of-distribution SNR test sets	10,000 unlabeled pre-train; 1000/100/100 train/val/test for fine-tune
Backbone	ViT: 224^2 input, 16^2 patches, dim. 768, 12 layers, 12 heads; 85.8 M params	Shared 12-layer encoder + 4-layer decoders (pre-train only), dim. 768, 75% masking; 85.8 M params
Training	Adam, LR 5×10^{-5}, batch 128; fine-tune updates last linear layer only; no augmentation	AdamW pre-train (LR 10^{-4}, batch 64); Adam fine-tune (LR 10^{-4}, batch 32), MLP head; no augmentation
Inference	14.5 ms/sample	14.5 ms/sample

7. Research Directions and Potential Solutions

Foundation models raise a number of open problems in wireless systems, and closing them will take focused work. Below we lay out the directions that look most promising, along with the kinds of solutions each might involve.

- Lightweight Architecture Development: Making foundation models small and efficient enough for wireless use is one of the more pressing directions. Two threads stand out:
  - Model Compression Techniques: Pruning, quantization, and knowledge distillation, tuned for wireless tasks specifically, can shrink a model substantially while keeping its spectrum-sensing and interference-detection accuracy.
  - Neural Architecture Search (NAS): Searching automatically for architectures that balance accuracy against cost, which can surface designs suited to wireless-specific tasks without losing what foundation models provide.
- Edge-Native Foundation Models: Running these models at the edge takes more than shrinking them; it changes how they are designed:
  - Distributed Learning Frameworks: Architectures that split a model across network layers, so computation can be shared between edge devices and central nodes.
  - Hardware-Aware Optimization: Designs that account for the hardware limits of wireless devices from the start, optimizing for energy as well as accuracy.
- Data Generation and Augmentation: Data scarcity calls for ways to create or stretch what little data exists:
  - Physics-Informed Synthetic Data: Simulators that build in physical channel models, so the signals they generate are realistic enough to train on.
  - Self-Training Methods: Self-supervised techniques that make better use of unlabeled wireless data and reduce the dependence on hand-labeled sets.
- Real-Time Adaptation Mechanisms: Keeping up with environments that change quickly:
  - Online Learning Frameworks: Updating a model continuously from streaming data without letting it go unstable or lose accuracy.
  - Meta-Learning Approaches: Architectures that adapt to a new wireless scenario from very little extra training.
- Interpretable Wireless Foundation Models: Making the models easier to inspect and trust:

  - Attention Mechanism Analysis: Attention mechanisms built to be readable, not just accurate, for signal-processing tasks.
  - Physics-Based Constraints: Folding domain knowledge and physical constraints into the architecture, which helps both interpretability and reliability.

- Standardization and Integration Solutions: Lowering the barrier to adoption in real systems:
  - Interface Standardization: Agreed APIs and protocols for plugging foundation models into existing wireless infrastructure.
  - Compliance Frameworks: Verification and validation methods that show a model meets regulatory requirements.

- Energy-Efficient Computing Paradigms: Bringing power consumption down:
  - Adaptive Computation: Scaling how much computation the model does to the difficulty of the task and the resources on hand.
  - Energy-Aware Training: Training procedures that optimize for energy alongside accuracy instead of treating it as an afterthought.

- Cross-Domain Generalization: Making models hold up across different wireless settings:
  - Domain-Invariant Features: Representations that capture the underlying wireless structure rather than the particulars of one deployment.
  - Transfer Learning Mechanisms: Efficient transfer techniques for adapting quickly to a new environment.

Together these directions sketch a route from the current limitations toward foundation models that can handle messy, fast-changing wireless environments under real deployment constraints.

Getting there will take academia, industry, and standards bodies working together, and the work will need to track the wireless technologies and standards that are actually emerging so that it stays relevant.

If it comes together, the payoff is networks that can run largely on their own, use their resources efficiently, and adapt as conditions shift, which is what the next generation of wireless applications will demand.

8. Conclusions

Foundation models give wireless communication a genuinely different way to work: one model, pre-trained once, that can be reused across tasks, adapted with little data, and fed several kinds of signal at once. Those traits are what make them a good fit for spectrum awareness, interference mitigation, and resource allocation. The obstacles are just as clear, and we have tried to be honest about them: data is scarce, the models are expensive to run, and adapting fast enough for real-time use is still hard. The most useful next steps are lightweight and edge-native architectures and more energy-efficient computing, so the models can run where wireless systems actually live, together with the interpretability and regulatory compliance that any regulated industry will want before it trusts them. None of these problems is trivial, but none looks fundamental either, and working through them is what will let foundation models support the intelligent, largely autonomous networks that 5G and beyond will need.

Author Contributions

M.R. and A.F. contributed equally to this work. Conceptualization, M.R., A.F. and R.G.R.; validation, M.R., A.F. and R.G.R.; formal analysis, M.R., A.F. and R.G.R.; writing, M.R., A.F. and R.G.R.; review and editing, M.R., A.F., R.G.R., N.M., Y.-D.Y. and H.W.; supervision, N.M., Y.-D.Y. and H.W.; project administration, H.W.; funding acquisition, H.W. All authors have read and agreed to the published version of the manuscript.

Funding

This work is funded by the U.S. Department of Energy, funding number DE-FE0032196.

Data Availability Statement

Not applicable.

Conflicts of Interest

As the Editor-in-Chief and an Associate Editor of the journal, respectively, Y.-D.Y. and H.W. had no involvement in the peer review of this paper and had no access to information regarding its peer-review process. Full responsibility for the editorial process of this paper was delegated to another editor of the journal.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used Claude (Anthropic), ChatGPT (OpenAI) and Gemini (Google) to assist with literature search, proofreading, grammar correction, paraphrasing and improving the clarity of the writing. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Vishwakarma, A.K.; Chaurasia, S.; Kumar, K.; et al. Internet of things technology, research, and challenges: A survey. *Multimed. Tools Appl.* **2025**, *84*, 8455–8490. https://doi.org/10.1007/s11042-024-19278-6.
2. Rawat, D.B.; Song, M.; Shetty, S. An Overview of Cognitive Radio Networks. In *Dynamic Spectrum Access for Wireless Networks*; Springer International Publishing: Cham, Switzerland, 2015; pp. 1–11. https://doi.org/10.1007/978-3-319-15299-8_1.
3. Devlin, J.; Chang, M.W.; Lee, K.; et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186. https://doi.org/10.18653/v1/n19-1423.
4. Kotaru, M. Adapting Foundation Models for Information Synthesis of Wireless Communication Specifications. *arXiv* **2023**, arXiv:2308.04033v2.
5. Adesina, D.; Bassey, J.; Qian, L. Robust Deep Radio Frequency Spectrum Learning for Future Wireless Communications Systems. *IEEE Access* **2020**, *8*, 148528–148540. https://doi.org/10.1109/access.2020.3015939.
6. Fontaine, J.; Shahid, A.; De Poorter, E. Towards a Wireless Physical-Layer Foundation Model: Challenges and Strategies. *arXiv* **2024**, arXiv:2403.12065v2.
7. Faysal, A.; Rostami, M.; Roshan, R.G.; et al. NMformer: A Transformer for Noisy Modulation Classification in Wireless Communication. In Proceedings of the 2024 33rd Wireless and Optical Communications Conference (WOCC), Hsinchu, Taiwan, 25–26 October 2024; pp. 103–108. https://doi.org/10.1109/wocc61718.2024.10786062.
8. Faysal, A.; Boushine, T.; Rostami, M.; et al. DenoMAE: A Multimodal Autoencoder for Denoising Modulation Signals. *IEEE Commun. Lett.* **2025**, *29*, 1659–1663. https://doi.org/10.1109/lcomm.2025.3570602.
9. Wolf, D.; Payer, T.; Lisson, C.S.; et al. Self-supervised pre-training with contrastive and masked autoencoder methods for dealing with small datasets in deep learning for medical imaging. *Sci. Rep.* **2023**, *13*, 20260. https://doi.org/10.1038/s41598-023-46433-0.
10. Tamizhelakkiya, K.; Gauni, S.; Chandhar, P. Modulation classification analysis of CNN model for wireless communication systems. *AIMS Electron. Electr. Eng.* **2023**, *7*, 337–353. https://doi.org/10.3934/electreng.2023018.
11. O'Shea, T.; Hoydis, J. An Introduction to Deep Learning for the Physical Layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. https://doi.org/10.1109/tccn.2017.2758370.
12. Alikhani, S.; Charan, G.; Alkhateeb, A. Large Wireless Model (LWM): A Foundation Model for Wireless Channels. *arXiv* **2024**, arXiv:2411.08872.
13. Ott, J.; Stahlke, M.; Feigl, T.; et al. Simplicity is Key: An Unsupervised Pretraining Approach for Sparse Radio Channels. *arXiv* **2025**, arXiv:2505.13055.
14. Palhares, V.; Taner, S.; Studer, C. CSI2Vec: Towards a Universal CSI Feature Representation for Positioning and Channel Charting. *arXiv* **2025**, arXiv:2506.05237.
15. Liu, Y.; Liu, M.; Xie, W.; et al. RIS-MAE: A Self-Supervised Modulation Classification Method Based on Raw IQ Signals and Masked Autoencoder. *arXiv* **2025**, arXiv:2508.00274.
16. Mashaal, O.; Abou-Zeid, H. IQFM—A Wireless Foundational Model for I/Q Streams in AI-Native 6G. *arXiv* **2025**, arXiv:2506.06718.
17. Li, Z.; Li, S.; Yan, X. Time series as images: Vision transformer for irregularly sampled time series. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 49187–49204.
18. Faysal, A.; Rostami, M.; Boushine, T.; et al. DenoMAE2.0: Improving Denoising Masked Autoencoders by Classifying Local Patches. *arXiv* **2025**, arXiv:2502.18202.
19. Xie, L.; Li, C.; Zhang, X.; et al. TRLS: A Time Series Representation Learning Framework via Spectrogram for Medical Signal Processing. *arXiv* **2024**, arXiv:2401.05431.
20. Parker, P.A.; Holan, S.H.; Ravishanker, N. Nonlinear time series classification using bispectrum-based deep convolutional neural networks. *Appl. Stoch. Model. Bus. Ind.* **2020**, *36*, 877–890. https://doi.org/10.1002/asmb.2536.

21. Jiang, J.; Yu, W.; Li, Y.; et al. A MIMO Wireless Channel Foundation Model via CIR-CSI Consistency. *arXiv* **2025**, arXiv:2502.11965.
22. Radford, A.; Kim, J.W.; Hallacy, C.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8748–8763.
23. Behzad, K.; Zandi, R.; Motamedi, E.; et al. RoboMNIST: A Multimodal Dataset for Multi-Robot Activity Recognition Using WiFi Sensing, Video, and Audio. *arXiv* **2025**, arXiv:2408.16703v2.
24. Chong, D.; Wang, H.; Zhou, P.; et al. Masked Spectrogram Prediction For Self-Supervised Audio Pre-Training. *arXiv* **2022**, arXiv:2204.12768.
25. He, K.; Chen, X.; Xie, S.; et al. Masked Autoencoders Are Scalable Vision Learners. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 15979–15988. https://doi.org/10.1109/cvpr52688.2022.01553.
26. Guler, B.; Geraci, G.; Jafarkhani, H. A Multi-Task Foundation Model for Wireless Channel Representation Using Contrastive and Masked Autoencoder Learning. *arXiv* **2025**, arXiv:2505.09160v2.
27. Mu, S.; Chen, S.; Zu, Y.; et al. Contrastive Learning with Feature-Level Augmentation for Wireless Signal Representation. *Electronics* **2025**, *14*, 2728. https://doi.org/10.3390/electronics14132728.
28. Tiras, F.E.; Altinoluk, H.S. CrossRF: A Domain-Invariant Deep Learning Approach for RF Fingerprinting. *arXiv* **2025**, arXiv:2505.18200.
29. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; et al. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661.
30. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2013**, arXiv:1312.6114.
31. Böck, B.; Oeldemann, A.; Mayer, T.; et al. Physics-Informed Generative Modeling of Wireless Channels. *arXiv* **2025**, arXiv:2502.10137.
32. Gupta, C.; Das, R.K.; Barik, R.K.; et al. GANCE: Generative Adversarial Network Assisted Channel Estimation for Unmanned Aerial Vehicles Empowered 5G and beyond Wireless Networks. *IEEE Access* **2025**, *13*, 198–213. https://doi.org/10.1109/access.2024.3522847.
33. Hu, E.J.; Shen, Y.; Wallis, P.; et al. LoRA: Low-rank adaptation of large language models. In Proceedings of the International Conference on Learning Representations, Virtual, 25–29 April 2022; Volume 1; p. 3.
34. Zhang, H.; Zhou, F.; Wu, Q.; et al. FSOS-AMC: Few-Shot Open-Set Learning for Automatic Modulation Classification. In Proceedings of the 2024 16th International Conference on Wireless Communications and Signal Processing (WCSP), Hefei, China, 24–26 October 2024; pp. 746–751. https://doi.org/10.1109/wcsp62071.2024.10826902.
35. Wang, Y.; Gao, Z.; Zheng, D.; et al. Transformer-Empowered 6G Intelligent Networks: From Massive MIMO Processing to Semantic Communication. *IEEE Wirel. Commun.* **2023**, *30*, 127–135. https://doi.org/10.1109/mwc.008.2200157.
36. Stratton, J. An Introduction to Microsoft Copilot. In *Copilot for Microsoft 365*; Apress: Berkeley, CA, USA, 2024; pp. 19–35. https://doi.org/10.1007/979-8-8688-0447-2_2.
37. Aboulfotouh, A.; Eshaghbeigi, A.; Abou-Zeid, H. Building 6G Radio Foundation Models with Transformer Architectures. *arXiv* **2024**, arXiv:2411.09996v1.
38. Chin, W.L. On the Noise Uncertainty for the Energy Detection of OFDM Signals. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7593–7602. https://doi.org/10.1109/tvt.2019.2920142.
39. Herrera-Loera, C.; Del-Valle-Soto, C.; Valdivia, L.J.; et al. A Systematic Review and Energy-Centric Taxonomy of Jamming Attacks and Countermeasures in Wireless Sensor Networks. *Sensors* **2026**, *26*, 579. https://doi.org/10.3390/s26020579.
40. Raffel, C.; Shazeer, N.; Roberts, A.; et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.