On the Feasibility of Modeling OFDM Communication Signals with Unsupervised Generative Adversarial Networks

Jack Sklar and Adam Wunderlich

High-quality recordings of radio frequency (RF) emissions from commercial communication hardware in realistic environments are often needed to develop and assess spectrum-sharing technologies and practices, e.g., for training and testing spectrum sensing algorithms and for interference testing. Unfortunately, the time-consuming, expensive nature of such data collections together with data-sharing restrictions pose significant challenges that limit dataset availability. Furthermore, developing accurate models of real-world RF emissions from first principles is often very difficult because system parameters and implementation details are at best only partially known, and complex system dynamics are difficult to characterize. Hence, there is a need for flexible, data-driven methods that can leverage existing datasets to synthesize additional similar waveforms. One promising machine learning approach is unsupervised deep generative modeling with generative adversarial networks (GANs). To date, GANs for RF communication signals have not been studied thoroughly. In this paper, we present the first in-depth investigation of generated signal fidelity for GANs trained with baseband orthogonal frequency-division multiplexing (OFDM) signals, where each subcarrier is digitally modulated with quadrature amplitude modulation (QAM). Building on prior GAN methods, we propose two novel GAN models and evaluate their performance using simulated datasets with known ground truth. Specifically, we investigate model performance with respect to increasing dataset complexity over a range of OFDM parameters and conditions, including fading channels. The findings presented here inform the feasibility of use-cases and provide a foundation for further investigations into deep generative models for RF communication signals.

Index Terms—generative adversarial network (GAN), time-series, machine learning, RF datasets

I. INTRODUCTION

To aid the development of spectrum-sharing technologies, practices, and policies [1], [2], it is necessary to characterize emissions in the band of interest, assess spectrum sensing performance, and evaluate interference between heterogeneous systems [2]–[6]. One issue complicating such studies is the time and expense required to collect high-quality radio frequency (RF) recordings of real-world emissions; see [7]–[9] for examples. Moreover, there are few publicly available libraries of RF recordings from real-world systems.

Although early-stage development can be executed with simulated data, it is very challenging to develop accurate models of RF emissions for real-world modern communication systems from first principles. Specifically, in typical measurement scenarios with real hardware, system parameters and implementation details are at best only partially known, and complex system dynamics are difficult to characterize. For example, it is difficult to accurately model out-of-band emissions from real radio frequency (RF) hardware, power control and scheduling dynamics, traffic loading, and wireless propagation [3].

Due to the dearth of high-quality RF recordings from real-world communication systems, and considering the aforementioned challenges of modeling realistic emissions from “black-box” systems, there is a need for flexible, data-driven alternatives that can leverage existing datasets to synthesize additional waveforms. Unfortunately, there is currently no established state-of-the-art modeling approach for this problem, and experimental testing often relies on idealistic, synthetic waveforms, e.g., [10]–[12]. One way to characterize a dataset of RF recordings with unknown properties is through the use of unsupervised deep generative modeling.

Deep generative models utilize a deep neural network to produce samples from a high-dimensional target distribution defined by a training set. In recent years, deep generative models, and most notably, generative adversarial networks (GANs), have drawn a great deal of attention in the machine learning community and have progressed rapidly, e.g., see [13]–[17] for reviews. Namely, GANs and other deep generative models have been used to successfully synthesize and process realistic images, speech, and video. Because deep generative models can be trained in an unsupervised manner, they are a potentially flexible and powerful approach for characterizing real-world RF emissions with unknown, complex attributes.

To date, most work on deep generative models has focused on computer vision applications with images [13]–[17], while time-series have received less attention. Nonetheless, there have been several papers proposing generative models for time-series, primarily for audio applications. Examples of non-GAN deep generative models for audio time-series include WaveNet [18], an autoregressive model, and MusicVAE [19], which uses a variational autoencoder (VAE).

Several time-series GAN models aim to leverage prior work on GANs for images by training the generator to produce an image-domain time-frequency representation, such as a spectrogram, which is then mapped into a time-series. Examples of models employing this approach include SpecGAN [20], MelGAN [21], TSGAN [22], GANSynth [23], and TiFGAN [24]. Additional approaches are compared by Nistal et al. [25].

Alternatively, there have been some papers proposing GANs that directly model time-series. One class of methods includes architectures based on recurrent neural networks (RNNs), such as Long-Short Term Memory (LSTM), e.g., [26], [27]. Methods based on convolutional neural networks (CNNs), include WaveGAN [20], which employs a flattened version of
the popular DCGAN model [28], and QuantGAN [29], which uses temporal convolutional networks (TCNs) [30].

In the field of wireless communications, there have been some recent efforts to apply GANs. Examples include GANs for data augmentation for signal classification [31], [32], wireless channel modeling [33], anomaly detection [34], and adversarial attacks on communication systems [35], [36]. Due to their focus on use-case performance, e.g., signal classification accuracy, none of the aforementioned works present comprehensive evaluations of generated signal fidelity, nor do they compare their models to other approaches. Furthermore, these prior works do not investigate how their models perform with respect to increasing dataset complexity over synthetic data with known ground-truth, we investigate model effectiveness and associated performance limitations.

In this paper, we present the first in-depth investigation of generated signal fidelity for GANs applied to unsupervised modeling of baseband in-phase and quadrature (I/Q) orthogonal frequency-division multiplexing (OFDM) signals, where each subcarrier is digitally modulated with quadrature amplitude modulation (QAM) [37], [38]. OFDM is a commonly-used modulation and encoding scheme for digital transmission that is used, for example, in cellular networks and wireless local area networks (WLANs) [39], [40]. Namely, using synthetic data with known ground-truth, we investigate model performance with respect to increasing dataset complexity over a range of OFDM parameters and conditions, including fading channels. The findings presented here inform the feasibility of use-cases for GANs in RF communication system testing and modeling and provide a foundation for further investigations.

II. BACKGROUND: GENERATIVE ADVERSARIAL NETWORKS

First, we summarize key features of GAN models and Wasserstein loss with gradient penalty, the loss function that we use for training our models. Familiarly with the fundamentals of deep neural networks is assumed. For general background on deep learning, see the textbook by Goodfellow et al. [41].

A. Generative Adversarial Networks

A GAN consists of two neural networks, a generator, \( G \), and a discriminator, \( D \). The generator is trained to generate samples from a target data distribution, and the discriminator attempts to distinguish between generated and target data samples. Specifically, the generator learns to map latent vectors, \( z \), drawn from a multidimensional Gaussian distribution, \( p_Z(z) \), to the generator distribution \( p_G \), and the discriminator maps sample data, \( x \), to \( D(x) \), the probability that a sample belongs to the target distribution, \( p_d \).

In the original GAN formulation [42], the generator and discriminator compete against each other in the form of a zero-sum game. If the discriminator is trained to optimality before each generator update, the original GAN loss is equivalent to minimizing the Jensen-Shannon divergence between the target and generated distributions [42]. It was soon discovered that the original GAN model can suffer from training instabilities, such as diverging loss and mode-collapse. Subsequently, a large number of papers were published that attempted to address these shortcomings [13]–[17].

B. Wasserstein Loss with Gradient Penalty

A popular method that has been found to help stabilize GAN training is the Wasserstein loss function used by Arjovsky et al. [43]. The Wasserstein GAN (WGAN) aims to minimize the Wasserstein distance between the generated distribution and the target data distribution, where the Wasserstein “earthmover’s” distance can be interpreted as the minimum cost of transporting mass to transform one distribution into another. Due to the intractability of computing the Wasserstein distance directly, Arjovsky et al. instead proposed to train the generator to minimize the proxy loss

\[
L_W = \max_{p \in D} E_{x \sim p_d}[D(x)] - E_{x \sim p_G}[D(G(z))],
\]

where \( D \) is the set of 1-Lipshitz functions and \( E[\cdot] \) denotes expected value. When the discriminator is trained to optimality, i.e., the maximum is attained above, then minimizing the value function in (1) with respect to the generator parameters is equivalent to minimizing the Wasserstein distance between \( p_G \) and \( p_d \) [43], [44]. Arjovsky et al. enforced the Lipshitz constraint via weight clipping following every training update.

To further stabilize the WGAN model, Gulrajani et al. [44] proposed to add a gradient penalty term to the WGAN loss function instead of weight clipping. The resulting loss, denoted WGAN-GP, minimizes the objective

\[
L = E_{\tilde{x} \sim p_d}[D(\tilde{x})] - E_{x \sim p_G}[D(x)] + \lambda E[(\|\nabla_{\tilde{x}}D(\tilde{x})\|^2 - 1)^2],
\]

where \( \tilde{x} = \epsilon x + (1 - \epsilon) \tilde{x} \) with \( \tilde{x} \sim p_G \), \( x \sim p_d \), and \( \epsilon \sim U[0, 1] \), i.e., \( \epsilon \) is drawn from a uniform distribution over the unit interval. Note that \( \tilde{x} \) is a random linear interpolation between a real data sample, \( x \), and a generated data sample, \( \tilde{x} \).

Gulrajani et al. [44] make several implementation recommendations for WGAN-GP. First, since the WGAN-GP objective penalizes the gradient of the discriminator for each batch independently, the use of batch normalization is not recommended. Second, like Arjovsky et al. [43], Gulrajani et al. use an imbalanced discriminator-generator update rule, where the discriminator weights are updated five times for each generator update. Third, they recommend using \( \lambda = 10 \) for the default gradient penalty weight. Last, Gulrajani et al. recommend the ADAM optimizer [45] for discriminator and generator training with default hyperparameter settings \( \alpha = 10^{-4}, \beta_1 = 0, \beta_2 = 0.9 \) for the learning rate and moment decay rates, respectively.

III. SYNTHETIC OFDM DATASETS

In this work, we use synthetic (simulated) datasets of OFDM waveforms, which offer several advantages for early-stage investigations into GAN models. First, since high-quality recordings of real-world OFDM-based communication waveforms

1Python code implementing our models and experiments is available at https://github.com/usnistgov/OFDM-GAN. Also, training data and experimental results are available at https://doi.org/10.18434/mds2-2428
are not readily available, and since acquiring such recordings requires significant laboratory effort, the ease of creating unlimited amounts of synthetic data is well-suited to model development. Second, synthetic data provides control over multiple OFDM parameters, including OFDM symbol length, cyclic prefix, pilot symbols, and resource allocation size, i.e., the number of occupied subcarriers in an OFDM symbol. Last, because real-world communication system recordings involve complicated signaling protocols and suffer from non-ideal physical effects, implementing software-based channel equalization and demodulation is difficult. By contrast, synthetic data enables straightforward symbol demodulation and performance evaluation.

Synthetic datasets are constructed by first simulating a different random sequence of bits for each OFDM waveform, where 0 and 1 occur with equal probability. For M-ary QAM, each group of \( k = \log_2(M) \) bits is mapped using Gray encoding to QAM symbols \([37]\). Each block of QAM symbols is then mapped onto a specified collection of OFDM subcarriers, which are modulated into a baseband, I/Q, time-domain waveform by applying an inverse fast Fourier transform (FFT), producing the multicarrier OFDM symbol \([37]\).

Every synthetic waveform consists of a sequence of six OFDM symbols, each with a cyclic prefix equal to 25% of the OFDM symbol length. The OFDM symbol length, i.e., the number of subcarriers, is set to be 128, 256, or 512, yielding full time-series lengths of 960, 1920, or 3840, respectively. The above choices are motivated by the specifications for downlink long-term evolution (LTE) \([46, 47]\). Namely, symbol lengths of 128, 256, and 512 correspond to LTE channel bandwidths of 1.4 MHz, 3 MHz, and 5 MHz, respectively \([47]\). The cyclic prefix size corresponds to the so-called “extended” cyclic prefix option in LTE, for which there are six OFDM symbols per 0.5 ms “slot” \([46, 47]\).

Later, in Section VII-A we present three experiments. In the first experiment, we consider three different settings for the proportion of occupied OFDM subcarriers, i.e., the resource allocation. Namely, we set the proportion of occupied subcarriers equal to 25%, 50%, or 75% of the maximum allowed for downlink LTE \([46]\) Table 1). In each case, the block of occupied subcarriers is centered in frequency, and the zero frequency (DC) subcarrier is not used. For OFDM symbol lengths of 128, 256, and 512, the maximum number of occupied subcarriers, excluding the DC subcarrier, is taken to be 75, 150, and 300, respectively \([46]\) Table 1). Below, we refer to the three allocation sizes of 25%, 50%, or 75% as small, medium and large allocations, respectively.

To simulate the effect of thermal noise, additive white Gaussian noise (AWGN) is added to the OFDM signals. In each experiment, the AWGN level is set such that the error vector magnitude (EVM), as defined in Section VI is a specified level.

Figure 1 shows an example synthetic OFDM waveform together with a corresponding estimate of the power spectral density (PSD). Here, the OFDM symbol length is 256, 16-QAM is used on each occupied subcarrier, EVM = \(-25\) dB, and the allocation size is medium (50% occupancy). The dip in the PSD at zero frequency arises from the fact that the DC subcarrier is not used.

Our third experiment, presented below in Section VII-C applies 3GPP fading channel models \([48, 49]\) to the synthetic OFDM data. In addition to application of the random channel model, we modify synthetic data generation for this experiment by inserting a pilot symbol on the 4th OFDM symbol to enable estimation of channel frequency response and equalization coefficients \([38]\). Here, the pilot symbol is taken to be the first Zadoff-Chu base sequence, as defined for the demodulation reference signal (DMRS) in uplink LTE \([39]\) Sec. 5.5]. Complex-valued Zadoff-Chu sequences are commonly used for channel estimation because they have constant power in time and frequency \([49]\).

IV. NOVEL GENERATIVE MODEL ARCHITECTURES

Here we present two novel GAN models for OFDM signals that build on prior convolutional GANs for time-series. Specifically, we propose a direct time-series model and another indirect model based on an image-domain time-frequency representation. Key considerations in our designs include conceptual simplicity and scalability, i.e., the ability of the model to scale to longer and more complex signals, in terms of both computation and performance.

For many practical applications, the OFDM symbol length is known \(a \text{ priori}, \) e.g., it is determined by the channel bandwidth for 4G LTE. For this reason, the two models presented below...
are tailored to OFDM data in the sense that they utilize prior knowledge of the OFDM symbol length. However, all other aspects of the OFDM waveforms, e.g., the QAM symbol constellation, cyclic prefix size, OFDM symbol boundaries, and channel distortions are not assumed to be known.

### A. Progressively-Scaled-Kernel GAN (PSK-GAN)

Our direct time-series model, called progressively-scaled-kernel GAN (PSK-GAN), uses 1-D convolutional layers and aims to model temporal dynamics by progressively scaling kernel lengths with model depth. Specifically, PSK-GAN employs kernels with lengths progressively scaled up or down by a factor equal to the convolution stride with generator and discriminator model depth, respectively. The motivation behind progressively scaling kernel lengths is to increase the receptive field while avoiding kernels with lengths longer than their inputs. Progressively scaling kernel lengths has the effect of scaling the kernel resolution with feature map resolution. The concept of progressively scaling kernel sizes with layer depth is motivated by WaveNet [18], which employs dilated convolutions to achieve a similar outcome.

To avoid the generation of so-called “checkerboard artifacts,” which manifest as spikes in the power spectrum, convolutional layer kernel lengths are set to be integer multiples of the stride length, as recommended by Odena et al. [50].

Based on empirical testing, we set the maximum kernel length equal to the OFDM symbol length and the minimum allowable kernel length to 4.

Tables I and II outline the PSK-GAN architectures for the generator and discriminator, respectively. In these tables, the filter dimensions for convolutional layers correspond to kernel length, number of input channels, and number of output channels, respectively. Here, \( f = 1, 2, \) or 4 for OFDM symbol sizes of 128, 256, and 512, respectively. Also, \( n \) is the batch size.

Similarly, the filter dimensions for the dense layers correspond to input length and output length respectively. To yield time-series lengths that are compatible with convolutional layers with strides of 4, target OFDM signals are zero-padded to the nearest power of 2.

### B. Short-Time Fourier Transform GAN (STFT-GAN)

As mentioned in Section I, many time-series GANs train the generator to produce an image-domain time-frequency representation that is then mapped into a time-series. Motivated by these approaches, we propose a 2-D convolutional model, called short-time Fourier transform GAN (STFT-GAN), that is trained on a complex-valued short-time Fourier transform (STFT) representation of the OFDM time-series. Similar GANs based on STFT representations have also been used for audio generation, e.g., [23], [25]. Our model differs from these prior works in two ways. Namely, our model uses different network architectures and it directly uses the complex-valued STFT without additional processing. Related approaches that apply additional processing to the STFT, e.g., [23], [25], were not found to be advantageous in preliminary tests with synthetic OFDM datasets.

The STFT (a.k.a. windowed Fourier transform) is computed by dividing the time-series into overlapping segments of equal length, applying a window function, and then calculating the discrete Fourier transform (DFT) on each segment [51], [52]. We use a Hann window and 75% segment overlap. Under these conditions, the constant-overlap-add (COLA) constraint is satisfied [51], and the STFT is invertible, i.e., no information is lost.

OFDM target waveforms are first zero-padded to the nearest power of 2 before conversion to an STFT representation. We set the STFT window length equal to the OFDM symbol length, since empirical tests indicated that this choice gives superior results. The STFT values are rescaled to the range \([-1, 1]\) and shifted such that the zero-frequency component is at the center of each DFT window.

The architecture of STFT-GAN is based on the DCGAN architecture [28], with modifications made to accommodate the non-square shape of the STFT. Specifically, STFT-GAN is composed of four 2-D convolutional layers with \(4 \times 4\) kernels for both the generator and discriminator; see Tables III and IV. In the tables, \( f = 1, 2, \) or 4 corresponds to waveforms with OFDM symbol lengths of 128, 256, and 512, respectively and \( n \) is the batch size.
TABLE III
STFT-GAN GENERATOR ARCHITECTURE \([f = 1, 2, 4]\)

| Operation        | Filter Shape | Output Shape |
|------------------|--------------|--------------|
| \(z \sim \text{Uniform}(-1, 1)\) | (4, 4, 128, 2) | (n, \(1004, 8f, 2\)) |
| Dense            | (100, 16384, 5f) | (n, \(16384f, 2\)) |
| Reshape          | (n, \(1024, 8f, 2\)) | (n, \(1024, 8f, 2\)) |
| ReLU             | (4, 4, 1024, 512) | (n, \(512, 16f, 4\)) |
| Transpose Conv2-D (stride=2) | (4, 4, 512, 256) | (n, \(256, 32f, 8\)) |
| ReLU             | (4, 4, 256, 128) | (n, \(128, 64f, 16\)) |
| Transpose Conv2-D (stride=2) | (4, 4, 256, 128) | (n, \(128, 64f, 16\)) |
| ReLU             | (4, 4, 128, 2) | (n, \(2, 128f, 33\)) |
| Tanh             | (n, \(16384f, 1\)) | (n, \(1, 1\)) |

TABLE IV
STFT-GAN DISCRIMINATOR ARCHITECTURE \([f = 1, 2, 4]\)

| Operation        | Filter Size | Output Shape |
|------------------|-------------|--------------|
| \(x \sim f(z)\)  | (4, 4, 2, 128) | (n, \(2, 128f, 33\)) |
| Conv2-D (stride=2) | (4, 4, 2, 128) | (n, \(2, 128f, 33\)) |
| LReLU (\(\alpha = 0.2\)) | (4, 4, 128, 256) | (n, \(256, 32f, 8\)) |
| Conv2-D (stride=2) | (4, 4, 128, 256) | (n, \(256, 32f, 8\)) |
| LReLU (\(\alpha = 0.2\)) | (4, 4, 256, 128) | (n, \(128, 64f, 16\)) |
| Conv2-D (stride=2) | (4, 4, 256, 128) | (n, \(128, 64f, 16\)) |
| LReLU (\(\alpha = 0.2\)) | (4, 4, 256, 512) | (n, \(512, 16f, 4\)) |
| Conv2-D (stride=2) | (4, 4, 256, 512) | (n, \(512, 16f, 4\)) |
| LReLU (\(\alpha = 0.2\)) | (4, 4, 512, 1024) | (n, \(512, 8f, 2\)) |
| Reshape          | (n, \(16384f, 1\)) | (n, \(1, 1\)) |
| Dense            | (n, \(16384f, 1\)) | (n, \(1, 1\)) |

V. TRAINING PROTOCOL

Both PSK-GAN and STFT-GAN are trained with WGAN-GP loss described in Section II-B. Like the WGAN-GP training protocol, PSK-GAN and STFT-GAN are trained using the ADAM optimizer for discriminator and generator with hyperparameter settings of \(\alpha = 10^{-4}\), \(\beta_1 = 0\), and \(\beta_2 = 0.9\) for the learning rate and moment decay rates, respectively. In a departure from the WGAN-GP protocol, we use a 1:1 update ratio between the discriminator and generator, modified from the original 5:1 ratio. This choice was found to yield better convergence on our target datasets. We train each model with a target dataset of size \(2^{16} = 65,536\), for 500 epochs with a batch size of 128.

Following common practice with GANs, the training data is scaled to the range \([-1, 1]\), which corresponds to the range of the tanh output activation of the generator. Specifically, for STFT-GAN, all target distributions are scaled using feature-based min-max scaling that scales the minimum and maximum values of each time-step to \([-1, 1]\). By contrast, for PSK-GAN, the target distribution is scaled with min-max scaling using global minimum and maximum values. Global min-max scaling is used for PSK-GAN since it was found to yield better results. All outputs from the generator are rescaled back to the original range using the applicable inverse transformation.

VI. EVALUATION METHODS

Evaluations of GANs often focus on subjective assessments of perceptual quality or quantitative metrics that require a suitable feature space, defined, e.g., by a pre-trained model on a standard dataset \([53, 54]\). Since OFDM waveforms are not directly human interpretable, e.g., see Figure 1, it is not possible to assess OFDM signal fidelity on a perceptual basis. Moreover, there are no standard pre-trained classification models for time-series, so general purpose quantitative GAN evaluation measures that require a suitable feature space are not easily applied. Therefore, we focus our evaluations on OFDM-specific signal attributes. Specifically, we evaluate the quality of the power spectral density (PSD), the QAM constellation, and the cyclic prefix.

All evaluations are conducted with test sets that are \(1/4\) the size of the training set. Test sets have size \(2^4 = 16,384\). Test sets of target waveforms are created independently of the training set, and test sets of generated waveforms are created at the completion of training.

To avoid parametric assumptions, we estimate the PSD by applying the multitaper method \([55, 56]\), a versatile nonparametric approach, to the full duration of each waveform. The number of frequency bins is the next power of 2 greater than or equal to the waveform length. To obtain a representative PSD estimate across the test set, we take the median value in each frequency bin; see Figure 1 for an example median PSD estimate.

Let the median PSDs for the target and generated distributions be denoted as \(P_t(f_d)\) and \(P_g(f_d)\), respectively, where \(f_d \in [-0.5, 0.5]\) is normalized digital frequency with units of cycles per sample. To assess the accuracy of \(P_g\) relative to \(P_t\), we use the “geodesic distance” for power spectra proposed by Georgiou \([57, 58]\), defined as

\[
d_g(P_g, P_t) = \sqrt{\sum_{f_d} \left( \log \frac{P_g(f_d)}{P_t(f_d)} \right)^2 df_d - \left( \sum_{f_d} \log \frac{P_g(f_d)}{P_t(f_d)} df_d \right)^2}.
\]

The above quantity can be interpreted as the length of a geodesic connecting points on the manifold of power spectral densities \([57]\). Denoting the discrete-valued PSDs as \(P_g[k]\) and \(P_t[k]\), and approximating the integrals with summations, we obtain the discrete form

\[
d_g(P_g, P_t) \approx \sum_k \left( \frac{\log P_g[k]}{P_t[k]} \right)^2 \Delta f_d - \sum_k \log \frac{P_g[k]}{P_t[k]} \Delta f_d^2,
\]

where the summations are taken over the index for the normalized digital frequency grid with step-size \(\Delta f_d\). Since the choice of logarithm above is arbitrary, we choose to implement the above formula with a natural logarithm.

To quantitatively evaluate the quality of the QAM constellation, we use EVM, which measures the root mean square (RMS) deviation of measured symbols from the ideal signal constellation \([59, 60]\). Namely, we use the commonly-used definition \([59]\)

\[
\text{EVM} = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} |S_{\text{meas},i} - S_{\text{ideal},i}|^2 / \sum_{i=1}^{N_t} |S_{\text{ideal},i}|^2},
\]
where $N_s$ is the number of symbols in a random symbol sequence, $M$ is the number of unique symbols in the constellation, $S_{\text{meas},i}$ is the $i$th measured symbol, and $S_{\text{ideal},i}$ is the ideal constellation point for the $i$th symbol. Above it is assumed that the number of symbols in the sample, $N_s$, is large enough to ensure that all possible symbols and transitions are observed [59]. To define added noise levels and to assess signal fidelity, we use the above definition of EVM expressed in decibels (dB), i.e., $20 \log_{10} \text{EVM}$. We estimate EVM for a single waveform using the set of all QAM symbols in the waveform, and then find the median EVM value across all waveforms in a test set.

We supplement EVM assessment with qualitative evaluation of constellation diagrams, a commonly-used method to visualize digitally modulated signals. A conventional constellation diagram is a scatter plot in I/Q space of the sequence of measured symbols. Since the total number of QAM symbols in a test set is very large, we plot 2-D histograms instead of scatter plots. Specifically, we plot constellation diagrams using 2-D histograms with 150 × 150 bins, evenly spaced over the region $[-1.5, 1.5] \times [-1.5, 1.5]$. An example 16-QAM constellation diagram for a test set with EVM = $-25$ dB is shown in Figure 2.

The presence and quality of a cyclic prefix at the beginning of each OFDM symbol is evaluated as follows. First, for each waveform in the target and generated test sets, we find the cross-correlation function of each cyclic-prefix with the waveform, where all cyclic prefix sections are removed. The location and strength of the cross-correlation maximum indicates the accuracy of the cyclic prefix in the generated waveforms. We obtain an aggregate metric for cross-correlation strength by finding the median of the maximum cross-correlation values across the generated and target test sets, denoted $R_{\text{gen}}$ and $R_{\text{target}}$ respectively, and then computing the relative error, expressed as a percentage, i.e.,

$$\text{RelErr}_{\text{CP}}\% = \frac{|R_{\text{gen}} - R_{\text{target}}|}{|R_{\text{target}}|} \times 100. \quad (6)$$

As mentioned earlier in Section III, our third experiment, presented in Section VII-C, applies fading channel models to the synthetic OFDM data. To extract QAM symbols and evaluate the signal constellation, we first equalize the OFDM data on each subcarrier to correct for channel effects, a standard step prior to decoding received OFDM signals. Namely, the channel frequency response for each block of 6 OFDM symbols is estimated across occupied subcarriers by taking the demodulated pilot symbol located at the 4th OFDM symbol location and dividing it by the known pilot sequence. The QAM symbol on each subcarrier is then equalized by dividing by the corresponding estimated channel frequency response coefficient [38].

A commonly-used metric to characterize channels with frequency-selective fading is the coherence bandwidth, defined as the half-width at half-maximum of the channel’s time-frequency correlation function [38]. Because the coherence bandwidth characterizes the correlations between fades at different frequencies, it is particularly relevant for schemes like OFDM that transmit on multiple subcarriers [38]. For this reason, we use coherence bandwidth as a performance metric when we investigate fading channels in Section VII-C.

To estimate coherence bandwidth, we compute the autocorrelation function of the channel frequency response estimated with each pilot symbol, and find the half-width at half-max. We then plot histograms of the estimated values across the whole test set.

VII. EXPERIMENTS

Below, we present the results of three experiments: a data complexity experiment, a modulation-order experiment, and a fading channel experiment.

A. Data Complexity Experiment

The objective of the data complexity experiment is to evaluate how well our GAN models perform as the target OFDM dataset becomes increasingly complex. Namely, we change two experimental factors: OFDM symbol length and the resource allocation size (proportion of occupied subcarriers). For details on how these factors are implemented, see Section III. We use three settings for the OFDM symbol length, 128, 256, and 512, and three settings for the allocation size, denoted small, medium, and large, resulting in a total of nine test configurations. All target datasets use 16-QAM digital modulation on the occupied OFDM subcarriers and AWGN is added such that EVM = $-25$ dB. This EVM value was selected because it corresponds to a noticeable level of noise that results in essentially no bit errors, i.e., a strong communication link [61]. We compare the two GAN models presented in Section IV, PSK-GAN and STFT-GAN, to an implementation of WaveGAN [20], described in the Appendix. WaveGAN was chosen as a baseline model for comparison, since it is a state-of-the art direct time-series GAN.

To assess training variability, all models were trained on each dataset three times, with different neural network weight initialization and different batch randomization. On our computational hardware, the training time for each model was approximately 12 hours. Therefore, due to the high computational cost associated with training multiple model instances, it was necessary to limit the number of repetitions. For this reason, three repetitions of each configuration was selected to gain limited insights into training variability.
Figure 3 summarizes the results of the data complexity experiment for the PSD, EVM, and cyclic prefix evaluations, respectively. In these plots, the results for each model repetition are shown with circles. Also, lines connecting average values across the three repetitions are shown to aid visual interpretation. Error bars are omitted from the plots since the dominant source of uncertainty is training variability as reflected by the spread of the three repetitions, and the uncertainties for individual model results are too small to be visible.

The PSD results in Figure 3 (Top) show that STFT-GAN consistently had the smallest PSD distance relative to the target distribution and was fairly consistent across test conditions. On the other hand, PSK-GAN and WaveGAN displayed a wider range of performance, with much larger PSD distances in many cases.

The cyclic prefix results are shown in Figure 3 (Bottom). WaveGAN performed fairly well for the 128 and 256 symbol lengths, but was much worse for the 512 symbol length, with relative errors above 10%. On the other hand, PSK-GAN and STFT-GAN had relatively uniform performance across all conditions, with PSK-GAN displaying slightly better relative errors below 10%.

B. Modulation-Order Experiment

The goal of the modulation-order experiment is to assess how STFT-GAN performs with different QAM modulation orders, which are dynamically adjusted in LTE and wireless local area networks (WLANs) to adapt the data transmission rate to the propagation channel. The plots in Figure 3 show clear qualitative differences in performance between the models. Specifically, for all conditions, WaveGAN struggled to learn the full 16-QAM constellation. PSK-GAN successfully learned the 16-QAM constellation in the simpler conditions with smaller allocations and shorter symbol length, but performed worse as the symbol length and allocation size increased. On the other hand, STFT-GAN learned the 16-QAM symbol constellation well under all conditions.

The EVM results in Figure 3 (Middle) display clear trends in performance. Specifically, the direct time-series models, PSK-GAN and WaveGAN, show worsening performance as both the OFDM symbol size and allocation size increase, with PSK-GAN edging out WaveGAN, especially for small allocation sizes. By contrast, STFT-GAN markedly outperformed the direct time-series models, and did not display a degradation in performance as the symbol size and allocation size increased.

Overall, based on the superior performance of STFT-GAN as evidenced by the PSD and EVM metrics, as well as the constellation diagrams, we conclude that STFT-GAN is the most effective of the three models at adapting to increasing OFDM dataset complexity. For this reason, the next two experiments focus specifically on STFT-GAN.
Figure 4 shows estimated median power spectral densities (PSDs) for the data complexity experiment with 256 OFDM symbol length and medium allocation size. The PSD for the target distribution is shown in red and the PSD for the generated distribution is shown in blue. Left: WaveGAN, Middle: PSK-GAN, Right: STFT-GAN.

Figure 5 shows constellation diagrams for the data complexity experiment. Left: WaveGAN, Middle: PSK-GAN, Right: STFT-GAN.

Figure 6 shows median cross-correlation magnitude of all cyclic prefixes and OFDM symbols in the 256-medium condition for STFT-GAN.

Figure 7 shows constellation diagrams for the modulation order experiment. Left: Target distribution constellations for 4-QAM (upper left), 16-QAM, (upper right), 32-QAM (lower left), and 64-QAM (lower right). Right: Generated distribution constellations.

C. Fading Channel Experiment

The objective of the fading channel experiment is to evaluate the ability of STFT-GAN to learn waveform variations due to frequency-selective fading that arises from multipath RF propagation. Specifically, we apply stochastic N-tap Rayleigh fading channel models [38] to the target distribution OFDM waveforms. We use three channel models specified in the Third-Generation Partnership Project (3GPP) cellular standard [48, Annex B.2]: EPA-5Hz, EVA-70Hz, and ETU-300Hz. The first three letters specify a delay profile and the frequency is...
To be sensitive to frequency-dependent channel variations, the largest OFDM symbol length of $512$ is used. In addition, the allocation size is medium, the QAM modulation order is $M = 16$. To partially offset the nonuniform level of distortion from the different channel models, AWGN is added such that the EVM is $-30$ dB, $-40$ dB, and $-50$ dB before application of the EPA-5Hz, EVA-70Hz, and ETU-300Hz channel models, respectively.

Because the 3GPP channel models are specified in terms of physical quantities, their implementation requires specification of a physical sampling rate. We use a sampling rate of 7.68 MS/s, which corresponds to the 3GPP specification for a 5 MHz LTE downlink channel, where the OFDM symbol length is $512$.

Figure 8 presents histograms of estimated coherence bandwidth for the fading channel experiment. In each case, these plots show excellent agreement between the distributions of estimated coherence bandwidth for the generated and target data distributions. Moreover, for all cases, the PSD distances ranged between .075 to 0.1, indicating strong median PSD accuracy.

After channel equalization, the EVMs of the target and generated distributions were $-26.8$ dB and $-12.1$ dB, $-18.1$ dB and $-11.3$ dB, and $-7.8$ dB and $-10.1$ dB, for the EPA-5Hz, EVA-70Hz, and ETU-300Hz channel models, respectively. These results are consistent with the findings of the data complexity experiment, indicating that there is a limit to the constellation fidelity for lower target EVMs. Thus, while STFT-GAN failed to achieve target constellation fidelity, it successfully learned the target PSD and channel effects quantified by the coherence bandwidth.

VIII. DISCUSSION AND CONCLUSIONS

The findings presented here with synthetic data are anticipated to provide a foundation for future investigations into generative models trained with real-world recordings of RF time-series and other types of communications signals. By focusing on experiments with simulated data, we were able to gain important early-stage insights into model effectiveness that would not have otherwise been possible. Namely, the ease of creating unlimited amounts of synthetic data and the ability to control OFDM parameters allowed us to investigate the ability of models to handle increasing dataset complexity. Moreover, because synthetic data enables straightforward symbol demodulation, it was possible to directly evaluate QAM symbol constellation fidelity.

In the data complexity experiment, we found that a GAN based on an image-domain time-frequency representation, STFT-GAN, demonstrated superior performance to two GANs that directly model time-series, PSK-GAN and WaveGAN. Namely, STFT-GAN had better PSD fidelity while maintaining lower EVM across increasingly complex OFDM scenarios. By contrast, both direct time-series models did not perform as well when the OFDM symbol length and allocation size increased. Because both direct time-series models had worsening performance for longer OFDM symbol lengths and larger proportions of occupied subcarriers, they were not found to be viable candidates for further study. Hence, the remaining two experiments focused exclusively on STFT-GAN.

Perhaps not surprisingly, the modulation order experiment demonstrated that increasing the QAM modulation order with the same target EVM results in a decreased ability to learn the symbol constellation. Specifically, with a target EVM of $-25$ dB, STFT-GAN did not learn the $M = 64$ and $M = 32$ constellations as well as the $M = 16$ and $M = 4$ constellations. This observation is likely due to the fact that when the signal power is held constant, the density of constellation symbols in I/Q space increases with modulation order. Future studies on GANs with higher-order modulation schemes may need to overcome this limitation.

The fading channel experiment showed that STFT-GAN accurately learned the expected distribution of estimated coherence bandwidth for each channel type. Moreover, STFT-GAN learned generated distributions with median PSDs that closely matched the target distributions. Thus, this experiment demonstrated that GANs are capable of learning signal distortions due to stochastic fading channels.

In all three experiments, STFT-GAN achieved excellent PSD accuracy, including regions outside the set of active subcarriers. Moreover, it reliably learned the time-domain cyclic prefix location. On the other hand, while STFT-GAN learned the QAM signal constellation in many cases, it failed to achieve target EVMs. These findings indicate which use-cases are feasible for unsupervised generative models as well as where additional advances are needed. Namely, STFT-GAN is likely not a good candidate for end-to-end communication.
A state-of-the-art GAN that directly models time-series with CNNs is WaveGAN \cite{20}. In Section VII-A we compare an implementation of WaveGAN, described below, against other GAN models for OFDM data. The original WaveGAN model produces single-channel audio waveforms of length 16384, using an architecture based on a 1-D flattened version of the popular DCGAN model for images \cite{28}. WaveGAN attempts to widen its convolutional receptive field by modifying the DCGAN 5x5 kernels with 2x2 strides to 1-D kernels of length 25 with strides of 4 \cite{20}. Also, in contrast to DCGAN, WaveGAN has no normalization layers, adds a fully connected layer to both models, and is trained with WGAN-GP loss.

In addition, WaveGAN adds a novel layer to the discriminator, called phase shuffle, which performs a random circular shift of a convolutional layer’s output activation. Donahue et al. \cite{20} found optimal results when the random circular shift is between $-2$ and 2 time-steps. The aim of the phase shuffle operation is to make the model insensitive to the input waveform’s phase, preventing so-called “checkerboard” artifacts which manifest as spikes in the power spectrum \cite{50}.

Our slightly modified implementation of WaveGAN is shown in Tables \ref{tab:GeneTable} and \ref{tab:DiscTable}, where \(f = 1, 2, 4\) for OFDM symbol lengths of 128, 256, and 512, respectively, and \(n\) is the batch size. Like the original WaveGAN model, our implementation uses five convolutional layers for the generator and discriminator, but the dense layer is modified to support the lengths of our synthetic OFDM waveforms. We attempted to implement phase shuffle for layers for which it was compatible, i.e., layers with output dimension more than 4. However, we observed training instabilities with phase shuffle, and therefore do not include it in our implementation.

Following the original WaveGAN training implementation \cite{20}, we train the model with a 5:1 update ratio between the generator and discriminator and the Adam optimizer with \(\beta_1 = 0.5\) and \(\beta_2 = 0.9\). Otherwise, we follow the same training protocol described in Section VII. Namely, the learning rate is \(\alpha = 10^{-4}\) for the generator and discriminator, the batch size is 128, and the model is trained for 500 epochs. Also, all target distributions are scaled between [-1, 1] using feature-based min-max scaling.

### References

\[1\] C. B. Papadias, T. Ratnarajah, and D. T. Stocik, Eds., *Spectrum Sharing: The Next Frontier in Wireless Networks*. Hoboken, NJ: John Wiley & Sons, 2020.

\[2\] A. M. Voicu, L. Simić, and M. Petrova, “Survey of spectrum sharing for inter-technology coexistence,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1112–1144, 2019.

\[3\] D. Kuester, Y. Ma, D. Gu, A. Wunderlich, J. Coder, and J. Mruk, “Measurement challenges for spectrum sensing in communication networks,” in *15th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 2021.

\[4\] M. Danneberg, R. Datta, A. Festag, and G. Fettweis, “Experimental testbed for 5G cognitive radio access in 4G LTE cellular systems,” in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE, 2014, pp. 321–324.

\[5\] R. Caromi, J. Mink, C. Wedderburn, M. Souryal, N. El Ouni et al., “3.5 GHz ESC sensor test apparatus using field-measured waveforms,” in *Proc. WinComm 2018*. Wireless Innovation Forum, 2018.

\[6\] W. Young, D. McGillivray, A. Wunderlich, M. Krangle, J. Sklar, A. Sanders et al., “AWS-3 LTE Impacts on Aeronautical Mobile Telemetry,” National Institute of Standards and Technology, Tech. Rep. 2140, Feb 2021.

\[7\] P. Hale, J. Jargon, P. Jeavons, M. Lofquist, M. Souryal, and A. Wunderlich, “3.5 GHz radar waveform capture at Point Loma,” National Institute of Standards and Technology, Tech. Note 1954, May 2021.

\[8\] E. Nelson and D. McGillivray, “In-Situ Captures of AWS-1 LTE for Aeronautical Mobile Telemetry System Evaluation,” National Telecommunications and Information Administration, Tech. Rep. 21-553, March 2021.

\[9\] A. Sanders, K. Forsyth, R. Horansky, A. Kord, and D. McGillivray, “Laboratory Method for Recording AWS-3 LTE Waveforms,” National Institute of Standards and Technology, Tech. Rep. 2159, May 2021.

\[10\] F. H. Sanders, J. E. Carroll, G. A. Sanders, R. L. Sole, J. S. Devereux, and E. F. Drocella, “Procedures for Laboratory Testing of Environmental Sensing Capability Sensor Devices,” National Telecommunications and Information Administration, Tech. Rep. 18-527, Nov 2017.

\[11\] U.S. DOT Volpe Center, “Global Positioning System (GPS) Adjacent Band Compatibility Assessment: Final Report,” U.S. Department of Transportation.
of Transportation, Tech. Rep., April 2018. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/35535

[12] W. F. Young, A. D. Feldman, S. M. Genco, A. Kord, D. G. Kuester et al., “LTE Impacts on GPS,” National Institute of Standards and Technology, Tech. Rep. 1952, Feb 2017.

[13] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53–65, 2018.

[14] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, “How generative adversarial networks and their variants work: An overview,” ACM Computing Surveys (CSUR), vol. 52, no. 1, pp. 1–43, 2019.

[15] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, “Recent progress on generative adversarial networks (GANs): A survey,” IEEE Access, vol. 7, pp. 3232–3233, 2019.

[16] Z. Wang, Q. She, and T. E. Ward, “Generative adversarial networks and their variants work: An overview,” arXiv preprint arXiv:1906.01529, 2019.

[17] A. van den Oord, A. health, R. Korn, and P. Kretschmer, “Quant GANs: an intrinsic metric for power spectral density functions,” arXiv preprint math/0607026, July 2006.

[18] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer, “Quant GANs: Deep generation of financial time series,” in Proc. ICLR, 2019.

[19] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer, “Quant GANs: A Wavelet Tour of Signal Processing, 3rd ed. Academic Press, 2013.

[20] ———, “Spectral Audio Signal Processing,” in Proc. 28th European Signal Processing Conference (EUSIPCO), 2021, pp. 161–165.

[21] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[22] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[23] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[24] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[25] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[26] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[27] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[28] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[29] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[30] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[31] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[32] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[33] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[34] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in International Conference on Machine Learning. PMLR, 2018, pp. 4364–4373.

[35] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.

[36] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.

[37] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.

[38] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.

[39] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.

[40] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.

[41] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.

[42] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.

[43] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.

[44] ———, “Deep convolutional generative adversarial networks,” in Proc. ICLR, 2019.