

# Physical Layer Communications System Design Over-the-Air Using Adversarial Networks

Timothy J. O’Shea  
DeepSig Inc & Virginia Tech  
Arlington, VA  
toshea@deepsig.io

Tamoghna Roy  
DeepSig Inc.,  
Arlington, VA  
troy@deepsig.io

Nathan West  
DeepSig Inc.,  
Arlington, VA  
nwest@deepsig.io

Benjamin C. Hilburn  
DeepSig Inc.,  
Arlington, VA  
bhilburn@deepsig.io

**Abstract**—This paper presents a novel method for synthesizing new physical layer modulation and coding schemes for communications systems using a learning-based approach which does not require an analytic model of the impairments in the channel. It extends prior work published on the *channel autoencoder* to consider the case where the stochastic channel response is not known or can not be easily modeled in a closed form analytic expression. By adopting an adversarial approach for learning a channel response approximation and information encoding, we jointly learn a solution to both tasks applicable over a wide range of channel environments. We describe the operation of the proposed adversarial system, share results for its training and validation over-the-air, and discuss implications and future work in the area.

**Index Terms**—machine learning; deep learning; neural networks; autoencoders; generative adversarial networks; modulation; neural networks; software radio

## I. INTRODUCTION

The basic channel autoencoder [4], [9] architecture (shown in Figure 1) has recently received significant interest as a method for designing communications systems’ physical layer information encoding schemes. By jointly optimizing large multi-layer non-linear encoder and decoder networks with many degrees of expressive freedom, made possible by techniques collectively known as deep learning [3], it has been shown that effective new modulation schemes with an inherent error-correction capability can be readily learned for a variety of common channel models which is able to perform on par with maximum likelihood soft-decoding for small codes in terms of decoding error rates.

This is an exciting result in that rather than manually designing radio modulation schemes for analytic representation convenience (e.g., slicing on rectangular grid boundaries) and then measuring or demonstrating their optimal use, we can directly learn modulation and demodulation functions of high complexity which optimize the complete system for global performance metrics or loss functions (e.g., symbol error rate). This method can provide highly efficient solutions for analytic channel models of single-user, multi-user, and multi-antenna communications channels. To optimize for the effects of a real-world system, however - including the effects induced by digital conversion, analog RF hardware, and other sources of distortions and impairments - the analytic expression used within the channel model must accurately represent all of these

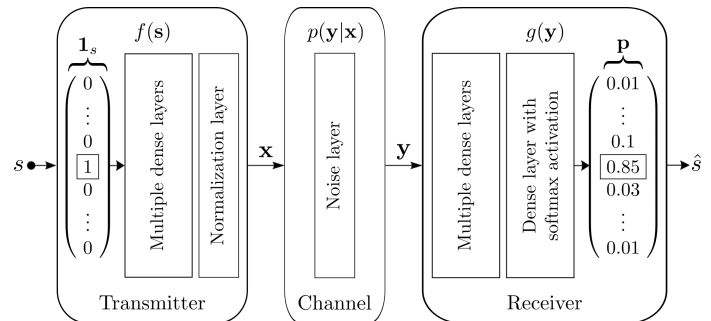


Fig. 1. A channel autoencoder system for learning physical layer encoding schemes optimized for a differentiable analytic channel model expression

effects, and it must do so in a differentiable way suitable for backpropagation.

Unfortunately, for many systems, it is difficult to accurately capture all of these effects in a closed-form analytic representation. Thus, they are often represented using simplified models which cannot fully express real-world complexities (e.g., device-to-device hardware variation, distortion, structured interference, etc). For this reason, it is highly desirable to learn a radio communication scheme directly from the sampled response of real physical hardware devices and channels rather than attempting to model and/or simplify the responses manually. One solution to this, which is implemented and explored in [6], is to perform gradient updates only on the receiver network after synthetic pre-training. However, due to the lack of a channel gradient expression, this approach provides no way to further update the transmitter or modulation scheme used. Differentiation through this wireless void is a difficult problem. An approach using reinforcement learning and sampling is proposed in [7] for related problems. Our approach, as presented in this paper, instead treats the problem as an adversarial function approximation problem.

In the next section we describe our proposed method for fully adapting the transmitter and receiver to a physical model-free channel response using a hardware-in-the-loop generative adversarial channel autoencoder network over-the-air.

## II. TECHNICAL APPROACH

Generative adversarial networks (GANs) [2] have recently been used effectively for a number of applications, including

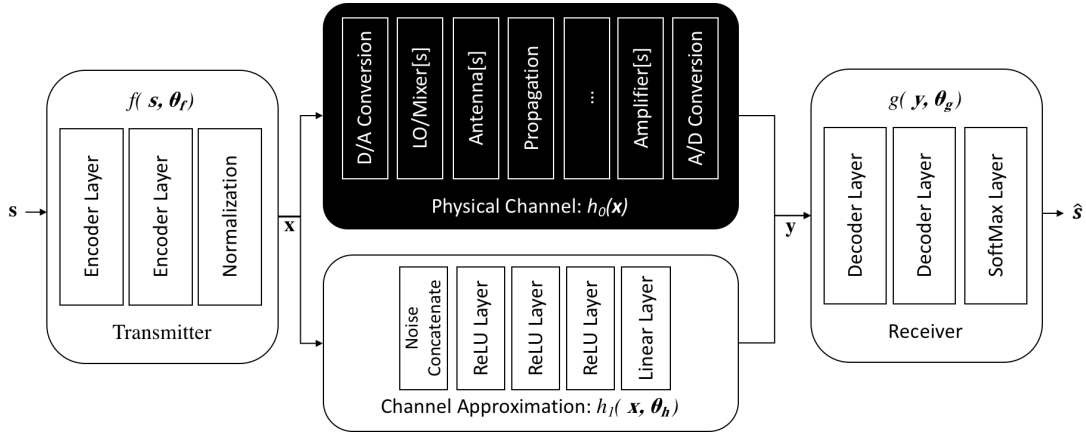


Fig. 2. A generative adversarial network for learning a communications system over a physical channel with no closed-form model or expression

the generation of false image examples which can confuse learned image recognition systems into mis-classifying objects, and which have demonstrated an unprecedented ability to generate realistic looking synthetic examples. These work by simultaneously training multiple adversarial objective functions, simultaneously or iteratively, which inform and improve each others' performance for some task. In the image generation case, the basic approach trains a discriminator whose objective is to minimize the error in classifying images as *real* or *fake*, and a second network whose objective is to minimize the error in generating images from random noise which classify as *real*. Since the original work in [2], numerous variations of this approach have significantly improved the performance of GANs [11], and helped to stabilize the notoriously unstable optimization problem, but they all generally rely on this core concept. Building upon this work, we combine our prior work on channel autoencoders with ideas inspired by GAN research to jointly optimize for the two tasks of:

- 1) approximating the response of the channel in any arbitrary communications system, and
- 2) learning an optimal encoding and decoding scheme which optimizes for some performance metric (e.g., low bit or symbol error rate)

The basic configuration of this approach, which we refer to as a Communications GAN (CommGAN), is shown in Figure 2. Here, as with the original channel autoencoder shown in Figure 1, we leverage two networks for encoding and decoding of information symbols ( $s$ ) which comprise our communications transmitter and receiver. The encoder network,  $f(s, \theta_f)$ , encodes codeword indices or bits into transmit waveform (sample values) or constellation (I/Q) values using a set of encoder weights  $\theta_f$  through a series of neural network layers. The encoder network comprises fully connected layers with a ReLU activation, which enables non-linear transformations, but could take on a number of different network architectures. The decoder network,  $g(y, \theta_g)$ , does the inverse - mapping the set of received samples (i.e., digitized voltage levels at

the receive antenna) into a set of pseudo-probabilities or likelihood levels for each bit or codeword which may have been transmitted. The decoder network also comprises a series of fully connected layers, again using ReLU for non-linear transforms, and a SoftMax output activation, but could also comprise any number of different network architectures. In contrast to our original work on the channel autoencoder, we do not employ a channel model such as an analytic expression for Additive White Gaussian Noise (AWGN) or Rayleigh fading, but instead introduce two forms of the channel  $h(x)$  to encompass modeling of any black-box channel transform:

- 1)  $h_0(x)$ : A real-world physical measurement of the response of a communications system comprising a transmitter, a receiver, and a channel.
- 2)  $h_1(x, \theta_h)$ : A non-linear multi-layer neural network which seeks to mimic the channel response of  $h_0$  synthetically.

---

#### Algorithm 1: CommGAN Training Routine

---

```

1  $\theta_f, \theta_h, \theta_g \leftarrow$  random initial weights
2 foreach epoch in epochs do
3   foreach step in steps0 do
4      $s_0 \leftarrow$  random symbol values
5      $y_0 = h_0(f(s_0, \theta_f))$ 
6      $\hat{y}_0 = h_1(f(s_0, \theta_f), \theta_h)$ 
7      $\theta_h \leftarrow$  Adam Update( $\mathcal{L}_0(y_0, \hat{y}_0)$ )
8   foreach step in steps1 do
9      $s_0 \leftarrow$  random symbol values
10     $\hat{s}_0 = g(h_1(f(s_0, \theta_f)), \theta_g)$ 
11     $\theta_f, \theta_g \leftarrow$  Adam Update( $\mathcal{L}_1(s_0, \hat{s}_0)$ )
12  Display metrics
13  Check stopping criteria

```

---

Throughout the training process, we use each of these channel expressions to iteratively arrive at an optimized solution. The simplified training approach used to train each of these networks is described in pseudo-code in Algorithm

1. In this work, we use a Mean-Squared Error (MSE) channel approximation loss ( $\mathcal{L}_0$ ), and a sparse categorical cross-entropy channel autoencoder loss ( $\mathcal{L}_1$ ), however numerous loss functions are possible.

As shown in Algorithm 1 we cycle between competing training objectives, updating weights for each network during the appropriate stage with manually tuned learning rates and relatively small networks for  $f$ ,  $g$ , and  $h$ , employing several fully connected ReLU layers for each. Tuning learning rates along with batch sizes in this fashion provides one relatively simplistic method for balancing the unstable learning rates of the two tasks. Improved methods do exist for improving the stability of GANs via the training algorithm, which we plan to leverage in future work (e.g. [5], [8], etc) but do not employ here at this point.

The physical channel  $h_0(x)$  is implemented using a Universal Software Radio Peripheral (USRP B210) [1] and a custom software stack developed at DeepSig. The ADC/DAC sample clocks of the transmit and receive hardware share a clock reference making the relative timing of the two predictable, but the devices have additional uncalibrated offsets, and impairments including filter group delay, analog component delay and response, antenna and amplifier distortion, and other over-the-air effects. We operate the radios in the 900 MHz ISM band using omni-directional whip antennas in a relatively benign indoor laboratory environment at a sample rate of 1 MSamp/sec. We loosely calibrate the transmit and receive times based on a rising edge pulse to within about 1 sample (1 microsecond), where unknown fractional timing error and channel effects exist between the transmit and receive samples. For purposes of the encoder and decoder networks, we map information to 16 discrete symbols (i.e., 4 bits per symbol). We use 3 samples per symbol on the transmitter, and in each stage we send the three samples from one symbol and consider a single receive sample at the receiver, which occurs somewhere within the unaligned 3-sample time window (i.e., there is an uncalibrated symbol offset between the transmitter and receiver).

### III. RESULTS

We found that while using MSE as the channel loss, normalization and noise insertion remain important aspects of network training, just as they were in the fully simulated channel autoencoder implementation. Effects such as clipping (i.e., 'saturation') in physical analog-to-digital converters are helpful to capture in normalization to improve training convergence. Including an average power constraint, like those used in simulation, seem to mitigate un-helpful gradients - for instance, solutions which simply continually increase the transmit power.

The training process completes in less than five minutes using an NVIDIA Titan V GPU. Figure 3 presents the loss curves of the channel approximation accuracy and the autoencoder symbol cross-entropy during the training process. Here, both functions reinforce the other, iteratively building better models

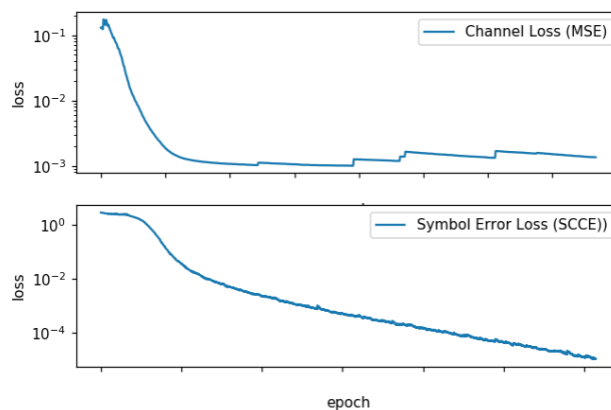


Fig. 3. CommGAN loss curves during the training process

of the channel and learning more optimal transmit signal representations as time and the number of epochs increase.

The resulting learned modulation scheme is a non-standard 16-QAM mode which achieves a symbol error rate of around 0.00714, measured over-the-air. Figure 4 plots the encoding scheme's learned I/Q samples after training, showing the three complex transmitted samples and the single complex received sample, taken at some fractional time within the 3-sample time window, with an over-the-air channel. We can see that symbols 0, 1, and 2 learn to transmit similar constellations with varying scaling and rotation, which interpolate to form a clean non-standard 16-QAM receive constellation, shown in the bottom right, with a different rotation relative to the transmitted constellations.

### IV. CONCLUSION

We have shown in this paper that by using an adversarial approach we can learn function approximations for arbitrary communications channels, and that by jointly learning a channel function approximation and an encoder/decoder scheme we can learn effective communication systems which achieve robust performance without needing a closed-form channel model or implementation, even with a simplified MSE loss function used for channel response approximation. Previously, over-the-air channel autoencoders required pre-training based on a closed-form model designed to match the expected deployment scenario and could not back-propagate through the black-box void of the radio channel, and thus only optimized the receiver side of the network [6]. Using an adversarial approach, we have shown that such a system can be learned directly on unknown and uncharacterized physical channels, and that the function approximation for this channel is sufficient to back-propagate and adapt both encoder and decoder networks.

Much future work remains for this approach, such as providing a more in depth analysis of the system performance, addressing training on a stream of information rather than a single symbol (also raised in [6]), investigating how such a system would perform under changing environmental conditions

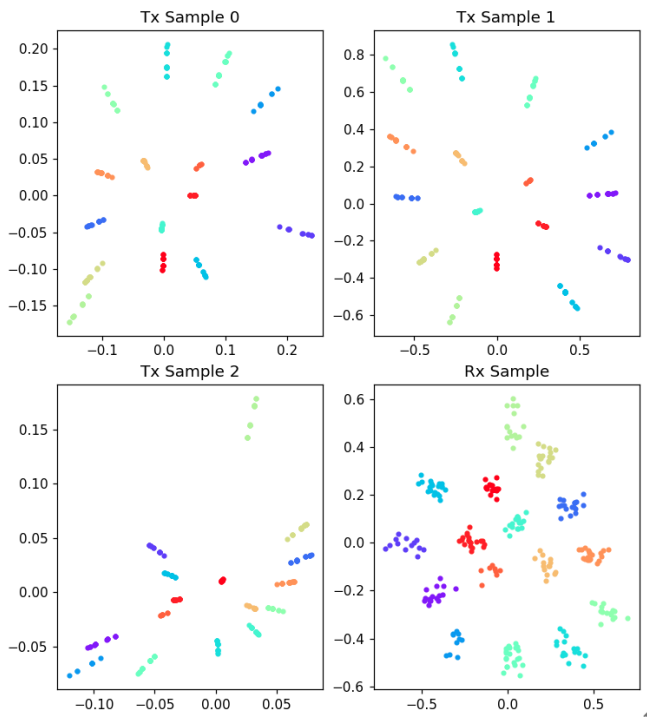


Fig. 4. Learned CommGAN scheme over the 900 MHz ISM band using a USRP B210 at 1 MSamp/sec, encoding 4 bits into one receive symbol from 3 unaligned over-the-air transmit symbols

with online over-the-air adaptation, and how it might scale to larger symbol block-sizes on the order of those used by modern LTE cellular standards. Finally, more recent work in approximating channel functions as stochastic functions using variational networks [10] and training using cross-entropy style GAN loss functions has shown significantly improved performance over the simplified MSE method used in this paper in representing complex random distributions. Applying this class of variational channel autoencoder GAN training rather than the simplified MSE-based training illustrated here has enormous potential to further increase performance and remove the need for the manual insertion of additional stochastic effects.

Learning directly from complex systems with high degrees of freedom, such as radio hardware and the propagation effects of physical channels, continues to be a challenging problem for modern communications systems with numerous sources of linear and non-linear impairments which can be difficult to accurately capture with simplified analytic models. This paper presented an alternative approach which addresses the complexities present in physical radio systems using approximation networks learned directly from data and experience. While this class of approach has yet to reach maturity, it represents a rapidly moving target, and one which we are working on maturing rapidly within our OmniPHY prototype communications system. Ultimately, this ability to optimize a system's performance holistically, over many joint effects, distortions and transformations, using principally real world

system measurement data in situ, is a fundamental requirement for the next major leap in the performance of communications systems, allowing for efficient adaptation, exploitation and resilience of a wide range of communications channels and effects.

#### REFERENCES

- [1] M. Ettus, *Universal software radio peripheral*, 2009.
- [2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [4] T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention," in *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, IEEE, vol. 2016, 2016, pp. 223–228.
- [5] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.
- [6] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning-based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, 2017.
- [7] W. Grathwohl, D. Choi, Y. Wu, G. Roeder, and D. Duvenaud, "Backpropagation through the void: Optimizing control variates for black-box gradient estimation," *arXiv preprint arXiv:1711.00123*, 2017.
- [8] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.
- [9] T. J. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, 2017.
- [10] T. J. O'Shea, T. Roy, and N. West, "Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks," *arXiv preprint arXiv:1805.06350*, 2018.
- [11] A. Hindupur, *The gan zoo*, Available at <https://github.com/hindupuravinash/the-gan-zoo>.