Modeling of Adaptive Receiver Performance Using Generative Adversarial Networks

Priyank Kashyap*, Yongjin Choi†,Sumon Dey†, Dror Baron*, Chau-Wai Wong*,
Tianfu Wu*, Chris Cheng†, Paul D. Franzon*
* Electrical and Computer Engineering, North Carolina State University
†Hewlett Packard Enterprise

***Abstract*—As the development of IBIS Algorithmic Modeling Interface (IBIS-AMI) models gets complex and requires time-consuming simulations, a data-driven and domain-independent approach have tremendous value. This paper presents a data-driven approach to modeling a high-speed serializer/deserializer (SerDes) receiver through generative adversarial networks (GANs). In this work, the modeling considers multiple channels, random bitstreams, and varying decision feedback equalizer (DFE) tap values to predict an accurate bit-error rate (BER). We employ a discriminator structure that improves the training to generate a contour plot with a root-mean-squared error (RMSE) of 0.014 to the ground truth and strongly correlates to the underlying bathtub curves, indicating a good fit.**

***Index Terms*—SerDes, receiver, behavior modeling, adaptive, generative, GAN, DFE, IBIS-AMI**

I. Introduction

With increasing clock frequencies, signal integrity and power integrity play an essential role in deciding whether a device will function as desired. Engineers must consider problems that may arise due to signal integrity and power integrity from the device's conception to production. To do so, engineers utilize computationally expensive and time-consuming simulations at each step of the design phase. Many times designs consist of different intellectual property (IP) blocks from different vendors. The vendors need to obfuscate and conceal these design blocks to protect their IP; thus, the industry utilizes IBIS-AMI models to share relevant transmitter and receiver design properties and compatibility with different electronic design automation (EDA) tools.

As the frequency of operation goes up, the design complexity of the transmitter and receiver circuitry increases. Transmitters consist of finite impulse response (FIR) filters with multiple taps, and receivers consist of continuous time linear equalizer (CTLE) and DFE filters to alleviate the signal integrity issues arising from intersymbol interference (ISI). However, the development of IBIS-AMI models requires multiple design iterations to develop an accurate model of their IP block. These design iterations can be time-consuming for engineers since they require detailed circuit-level simulations. Hence, it is crucial to find alternative strategies to improve simulation times and effectively assist engineers.

A rich swath of prior work attempts to solve the earlier issues of iterative long simulation times. Both [1] and [2] use non-linear system identification models to predict the receiver behavior. However, [1] requires unique models for each set of receiver settings, whereas [2] predicts the time-series output before reconstructing an eye diagram. Further, [3] and [4] model the impact of variable tap settings to an effect. Li et al. use a deep neural network (DNN) to learn a receiver's CTLE adaptation process [3] and evaluate the results using eye characteristics. Nguyen and Schutt-Aine recovers a pulse waveform using an recurrent neural network (RNN) that demonstrates the impact of the DFE tap setting. The RNN learns the initial state through a DNN that takes the tap settings and learns a latent space [4]. Other works have utilized machine learning (ML) algorithms such as support vector machine (SVM), but they tend to be limited to regression tasks in terms of system performance characteristics [5].

GANs are gaining traction in different EDA flows for image-based tasks. LithoGAN is an example where the authors use GANs to go from an input mask to an output resist pattern [6]. WellGAN is another example of the design flow where a GAN converts the input layout pattern to one that has the well regions defined [7]. Closely related to the problem on hand, GAN models simulated the work struggles with recovering individual lines of the eye diagram and does not consist of the various tap conditions in the receiver [8].

The rest of the paper is organized as follows. Section II consists the necessary background and prior work. Section III describes the proposed method and the metrics utilized for evaluation. Section IV discusses the problem that was analyzed, and how the dataset was generated. Section V shows the experimental results. Section VI concludes the article and presents a brief overview of future work.

II. Background

A. Bit-Error Rate (BER) Contour Plots

In today's high-speed links, the device's success depends on its BER, the number of bits that arrive at the receiver in error per unit time. Eye diagrams tell the engineer about the performance of a communication link. An eye diagram contains all possible transitions overlaid on top of each other; thus, it provides insight into which transitions may present an issue or how jitter in the link plays a role in determining the eye-opening. Though eye diagrams are sufficient, it has become increasingly challenging to measure BER of 1×10^{-15} using a sampling oscilloscope, and so many have started to use bathtub curves and their 2D representations, BER contours plots, as a measure of performance.

Unlike the case of the eye diagram, a bit-error rate tester (BERT) creates the bathtub curve by generating and passing data through a device under test. BERT compares the transmitted and received data for errors across the unit interval. BER contour plots are a 2D extension of the bathtub curves and provide us a view of the eye closure as the BER increases and reveal cases that an eye diagram might not recover as a sampling scope relies on a sparse sampling strategy, making it difficult to catch a bit error. Fig. 1 shows the BER contour plot and its corresponding bathtub curve.

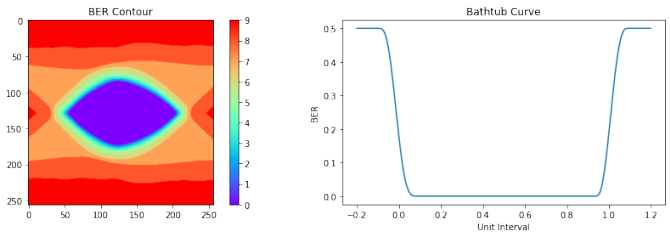


Fig. 1: BER contour plot and corresponding bathtub curve.

B. Generative Adversarial Network (GAN)

Generative networks have found tremendous use for tasks such as synthetic dataset generation, super-resolution, image translation, to name a few. The GANs themselves consist of two modules: the generator, *G*, which generates samples similar to the ground truth, and a discriminator, *D*, whose job is to determine if a sample presented to it is the ground truth or not. Mathematically, the generator attempts to learns a mapping from some noise vector, z, to a desired image, y, given parameters θ_g of the neural network, i.e. $G : z \rightarrow y$. The discriminator's function $D(y|\theta_d)$, returns the probability that the sample y, presented to it is from the dataset or the generator. Together the generator and discriminator play a min-max game where they try to outperform each other until they reach a state of equilibrium [9]. To that extent, the loss function is

$$\begin{aligned} L_{\text{GAN}}(G, D) = &\mathbb{E}_y \left[\log D(y)\right] \\ &+ \mathbb{E}_{x,z} \left[\log(1 - D(G(z|x)))\right]. \end{aligned} \tag{1}$$

Unlike regular GANs, the conditional GAN (cGAN) uses labeled data to enable the model to extend the latent space z and thus generate and discriminate images better. The generator learns a mapping given some input x, and noise vector z, to generate an output y. The discriminator trains to discern whether for the given input x, if y is from the dataset or the generator [10]. To that extent, the loss function in Eq. 1 is modified to give the loss function of the cGAN

$$\begin{aligned} L_{\text{cGAN}}(G, D) = &\mathbb{E}_{x,y} \left[\log D(y|x)\right] \\ &+ \mathbb{E}_{x,z} \left[\log(1 - D(G(x, z)|x))\right]. \end{aligned} \tag{2}$$

Additionally, prior work has shown that adding additional loss terms to the generator training improves the quality of the images generated by the model [10]. Thus the GAN's final objective is as follows:

$$L = L_{\text{cGAN}}(G, D) + \lambda \ell_1. \tag{3}$$

C. Gramian Angular Field (GAF)

SerDes modeling is inherently a time series problem; thus, to utilize GANs, which work well with images, we convert the time-series waveforms to a gramian angular field (GAF). Unlike other conversions, such as Markov Transition Field (MTF) and recurrence plots (RPs), which keep the transition dynamics and signal trajectories over an area, GAFs do not require additional tuning. The deep learning (DL) models can utilize previously unavailable features by converting the time series to an image. The GAF consists of the angular between time steps and uses it to exploit the temporal relationships, capturing channel effects such as inter-symbol interference.

To convert a waveform to a GAF, we scale the original data between $[-1, 1]$ or $[0, 1]$. The scaled time series is then expressed in the polar coordinate system by taking the arcosine of the value of the scaled time series at each time step. After converting to the polar coordinate system, we make the gramian matrix $m \times m$, where m is the length of the time sequence. The (i, j)th entry of the Gramian matrix is the trigonometric sum or difference between the ith and jth step to form the gramian angular sum field (GASF) or gramian angular difference field (GASF), respectively. The GASF is defined as follows:

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \dots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \dots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \dots & \cos(\phi_n + \phi_n) \end{bmatrix} \tag{4}$$

where ϕ_n is the nth time step encoded in the polar coordinate system.

III. Methodology

Similar to the Pix2Pix network, we use a U-Net-based generator as a starting point for the generator network [10]. The generator uses the waveform at the receiver's input as a GASF and the DFE tap settings to predict the appropriate BER contour plot. The network itself consists of three sub-networks, a time encoder network, a tap encoder network, and a decoder network. The time encoder learns a latent representation from the transformed waveform input and consists of convolution layers that downsample the input GASF. The tap encoder is a fully connected network that learns a latent space from the tap configurations. The decoder network takes in the combined latent representation from the two encoder models and attempts to recreate the correct BER contour plot. The decoder network has ConvTranspose layers to reconstruct the higher resolution BER plots and has skip connections from the time encoder network such that there is no information loss in the reconstruction process.

Unlike prior work, where a discriminator predicts patches to determine whether the sample presented is from the dataset or the generator, we employ a U-Net-based discriminator with two levels of prediction. By having two levels of prediction,

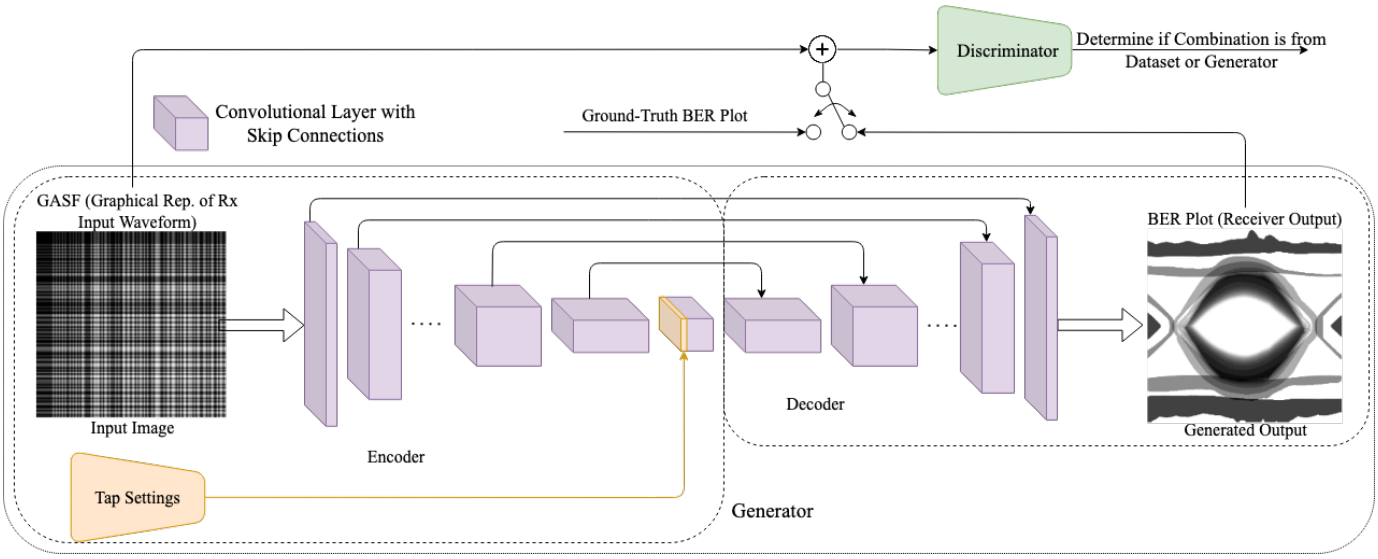


Fig. 2: GAN architecture used for training all implementations. The generator encoder and decoder with skip connections to forward information for the reconstruction of the eye diagram. The discriminator takes in the concatenation of the GASF and either the synthetic or ground-truth eye diagram.

the discriminator can focus on both global features and local details [11]. The global prediction from the network comes from the bottleneck in the U-Net, and it predicts whether the combination of the GASF, tap settings, and BER contour plot are from the dataset or the generator. The local prediction is a PatchGAN for the total image size and determines if each pixel value at each locationare from the dataset or the generator. However, unlike Schonfeld et al., we use regular GAN loss functions instead of hinge loss as the generated images are similar to the ground truth.

We use a separate neural network, the metric network, to validate the generator's results that produces a bathtub curve for a given BER contour plot. We employ this approach as it is not feasible to put the generated results back into a simulator to determine if they are significant. Additionally, metrics such as Frechet inception distance (FID) score, which compares image quality between a dataset and GAN generated images, do not reveal much. We train the metric network on the ground truth BER contour plots and bathtub curves and then use it to obtain the bathtub curves for the generated BER contour plots. The metric network constitutes multiple encoder modules from the generator and then has a fully connected layer and the output neurons corresponding to the number of steps in the bathtub curve.

IV. Dataset Creation

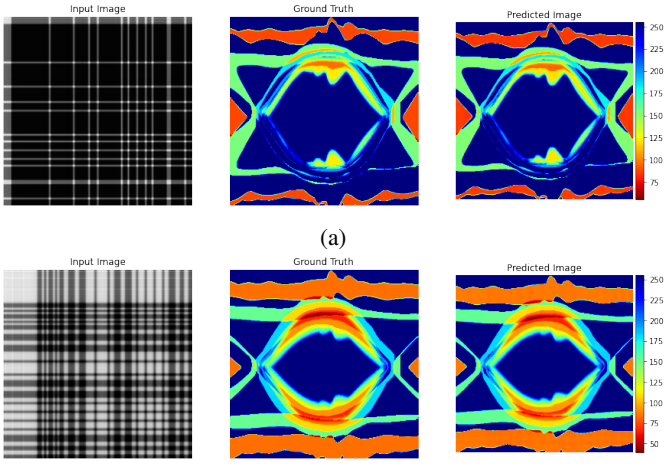
To create a representative transmission line, we create a parametric stripline model where the length of line, spacing between signal traces, the width of the signal traces, and the thickness of the traces are parameters. We use HFSS to capture the s-parameters of 8 different transmission lines and then use it as a link to simulate a SerDes running at $32\,\mathrm{Gb/s}$. Prior to the data collection, we determine the optimal tap settings for the DFE that yields the maximum eye-opening for each transmission line through Ansys Electronic Desktop. We then use Gaussian sampling around the optimal taps with a standard deviation equivalent to half of the optimal tap weight to collect the dataset.We capture the waveforms going into the receiver and the BER contour plots generated from the simulation.

The original waveform consists of 32000-time steps, sampled at every $0.5\,\mathrm{ps}$. We downsample the waveform by determining the appropriate Nyquist frequency, which we found $10\,\mathrm{ps}$. Thus, the preprocessing flow downsamples the original waveform to 3200-time steps before converting it to a GAF. The downsampling step is crucial as the GAF is an mxm matrix, making the GAF generation a significant bottleneck.

After we downsample the waveforms are during the GASF generation process, we transform the waveforms to a GASF, which is one of the inputs to the GAN. The BER contour plots captured by Ansys default to a resolution of 700×500, and the preprocessing flow digitizes and resizes them to a 256×256 image used by the GAN as a target. Lastly, the preprocessing flow normalizes the DFE tap settings $[-1, 1]$ to help the with model's convergence.

V. Experimental Results

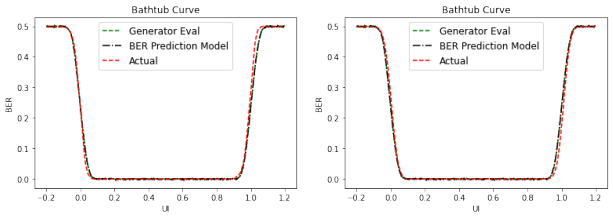
In Fig. 3, the first column contains the input waveform to the receiver, which we express as a GASF, the second column shows the ground-truth BER contour plot that we obtain from Ansys EDT, and the third column is the generated BER contour plot. Both Fig. 3 (a) and (b) are results based on the test set from our dataset. It is hard to determine whether any difference exists between the BER contour plots based on simple visual inspection. All the features are present in the correct locations, to the degree that we observe that transition between different BERs is correct.



(a) (b)

Fig. 3: Results using the cGAN with the left indicating the GASF input, middle representing the ground-truth and right representing the generated BER contour plot for two cases.

As with the eye diagram, we use a secondary network to evaluate the generated BER contour plots as there is no way to put the result back into the simulator. The metric network trains on the ground-truth BER contour and the corresponding bathtub curves. We then use the metric network to evaluate the cGAN's generated plots by looking at the resulting bathtub curve. When the metric network evaluates the generated contour plots, we observe a high correlation with the actual BER contour plots, as shown in Fig. 4. To further quantify the result, we calculate the RMSE between the generated and the ground truth contour plots for the dataset and find it to be 0.014, indicating a good fit.



(a) (b)

Fig. 4: Bathtub curve prediction on BER plot from ground truth, prediction and cGAN

VI. Conclusion

This paper presents a data-driven approach to modeling a high-speed SerDes receiver through GANs. The approach converts the time series to an intermediate representation from which the GAN performs a domain translation task based on the different DFE tap settings. We show that the model can interpolate between unseen tap values while generating images similar to the ground truth. Additionally, we demonstrate that the bathtub curves for the generated images and ground truths are correlated and have an RMSE of 0.014.

The impact of nonlinearities introduced by the CTLE in the receiver can be explored in future work. Additionally, the modeling aspect can include the impact of crosstalk and different signaling modes.

Acknowledgment

This research is supported in part by the NSF under Grants No. CNS 16-244770 (Center for Advanced Electronics through Machine Learning) and the industry members of the CAEML IUCRC.

References

[1] B. Li, P. Franzen, Y. Choi, and C. Cheng, "Receiver behavior modeling based on system identification," in *2018 IEEE 27th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2018, pp. 299–301.

[2] B. Li, B. Jiao, M. Huang, R. Mayder, and P. Franzon, "Improved system identification modeling for high-speed receiver," in *IEEE 28th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2019, pp. 1–3.

[3] B. Li, B. Jiao, C. .-H. Chou, R. Mayder, and P. Franzon, "CTLE adaptation using deep learning in high-speed serdes link," in *IEEE 70th Electronic Components and Technology Conference (ECTC)*, 2020, pp. 952–955.

[4] T. Nguyen and J. Schutt-Aine, "A tunable neural network based decision feed-back equalizer model for high-speed link simulation," in *IEEE 29th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2020, pp. 1–3.

[5] R. Trinchero and F. G. Canavero, "Modeling of eye diagram height in high-dpeed links via support vector machine," in *IEEE 22nd Workshop on Signal and Power Integrity (SPI)*, 2018, pp. 1–4.

[6] W. Ye, M. B. Alawieh, Y. Lin, and D. Z. Pan, "LithoGAN: End-to-end lithography modeling with generative adversarial networks," in *56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[7] B. Xu, Y. Lin, X. Tang, S. Li, L. Shen, N. Sun, and D. Z. Pan, "WellGAN: Generative-adversarial-network-guided well generation for analog/mixed-signal circuit layout," in *56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[8] P. Kashyap, W. S. Pitts, D. Baron, C.-W. Wong, and T. W. P. D. Franzon, "High speed receiver modeling using generative adversarial networks," in *2021 IEEE 30th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2021, pp. 1–3.

[9] 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*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https : / / proceedings . neurips . cc / paper / 2014 / file / 5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[10] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976.

[11] E. Schonfeld, B. Schiele, and A. Khoreva, "A U-Net based discriminator for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8207–8216.