2022 ACM/IEEE 4th Workshop on Machine Learning for CAD (MLCAD) | 978-1-6654-9729-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/MLCAD55463.2022.9900088

RxGAN: Modeling High-Speed Receiver through Generative Adversarial Networks

Priyank Kashyap
North Carolina State University
Raleigh, USA
pkashya2@ncsu.edu

Archit Gajjar
North Carolina State University
Raleigh, USA
amgajjar@ncsu.edu

Yongjin Choi
Hewlett Packard Enterprise
Colorado Spring, USA
yongjin.choi@hpe.com

Chau-Wai Wong
North Carolina State University
Raleigh, USA
chauwai.wong@ncsu.edu

Dror Baron
North Carolina State University
Raleigh, USA
barondror@ncsu.edu

Tianfu Wu
North Carolina State University
Raleigh, USA
tianfu_wu@ncsu.edu

Chris Cheng
Hewlett Packard Enterprise
Milpitas, USA
chris.cheng@hpe.com

Paul Franzon
North Carolina State University
Raleigh, USA
paulf@ncsu.edu

ABSTRACT

Creating models for modern high-speed receivers using circuit-level simulations is costly, as it requires computationally expensive simulations and upwards of months to finalize a model. Added to this is that many models do not necessarily agree with the final hardware they are supposed to emulate. Further, these models are complex due to the presence of various filters, such as a decision feedback equalizer (DFE) and continuous-time linear equalizer (CTLE), which enable the correct operation of the receiver. Other data-driven approaches tackle receiver modeling through multiple models to account for as many configurations as possible. This work proposes a data-driven approach using generative adversarial training to model a real-world receiver with varying DFE and CTLE configurations while handling different channel conditions and bitstreams. The approach is highly accurate as the eye height and width are within 1.59% and 1.12% of the ground truth. The horizontal and vertical bathtub curves match the ground truth and correlate to the ground truth bathtub curves.

CCS CONCEPTS

• **Hardware → Modeling and parameter extraction**; **Electromagnetic interference and compatibility**; **Analog and mixed-signal circuits**.

KEYWORDS

SerDes, receiver, behavior modeling, adaptive, generative, measurement, GAN, DFE, IBIS-AMI

MLCAD '22, September 12–13, 2022, Snowbird, UT, USA

ACM ISBN 978-1-4503-9486-4/22/09…$15.00
https://doi.org/10.1145/3551901.3556480

ACM Reference Format:
Priyank Kashyap, Archit Gajjar, Yongjin Choi, Chau-Wai Wong, Dror Baron, Tianfu Wu, Chris Cheng, and Paul Franzon. 2022. RxGAN: Modeling High-Speed Receiver through Generative Adversarial Networks. In *Proceedings of the 2022 ACM/IEEE Workshop on Machine Learning for CAD (MLCAD '22), September 12–13, 2022, Snowbird, UT, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3551901.3556480

1 INTRODUCTION

As transmission speed increases, the need for accurate link simulation is essential in today's devices. Signal and power integrity plays a crucial role for these devices in determining whether the device in question can successfully sample the data or not. Throughout the design process, engineers must contend with lengthy iterative simulations, which makes the entire process cumbersome. The transmitter's and receiver's complexity increases as these devices speed up. High-speed receivers have some combination of a CTLE, feed-forward equalizer (FFE), and DFE filter to compensate for channel impairments and distortions. The presence of these filters helps cancel undesired effects such as inter-symbol interference and equalize the impact of the channel.

Today's designs consist of multiple intellectual property (IP) blocks from various vendors. Vendors provide IBIS algorithmic modeling interface (IBIS-AMI) models to system designers to share details about their transmitter and receiver blocks while protecting the underlying IP. However, IBIS-AMI has two significant drawbacks: the engineering time required along with the computation needed and correlation to actual devices. The first problem is cumbersome as the process requires iterative detailed circuit-level simulations, which can take months. Further, simulations with IBIS-AMI models do not necessarily reflect what the devices do in a real-world setting, making the whole process fruitless [2].

Much work has gone into applying machine learning and other data-driven approaches to solve the abovementioned issues. However, most of the work estimates the final system parameters, obscuring away the modeling details from the engineer while using

domain knowledge to reduce the problem space [6, 11, 15]. Previous work that attempts to recover a complete understanding of the receiver performance through an eye diagram requires multiple models for each tap configuration scenario [9, 10]. Further, previous work has focused on varying the values of a single filter, whether the CTLE or DFE [3, 7, 9]. We propose a conditional generative adversarial network (cGAN) to model a receiver with varying CTLE and DFE configurations through a single model for an actual device by targeting its bit-error-rate (BER) plot. By doing so, the model can predict unseen conditions and is capable of interpolating intermediate configurations and handling unseen bitstreams. More importantly, there is only a single computation cost associated with an entirely data-driven approach: training the model, which is under 2 minutes per iteration or 2 hours for the entire training.

The contributions of this work are as follows:

- To the best of our knowledge, we are the first to use actual device measurements to show that a cGAN is capable of modeling a serializer/deserializer (SerDes) receiver.
- The DFE and CTLE values are jointly incorporated as tuneable parameters from which the cGAN is able to determine the correct operation of the receiver.
- We show the ability to window the time-series data prior to transforming it into an image. Such a transformation enables us to frame the problem as an image-to-image translation task.

To that extent, the generated BER contour plot matches the underlying characteristics: eye height and width with a root-mean-squared error (RMSE) of 0.015. Additionally, the bathtub curves extracted from the BER plots match the ground truths with a Pearson correlation coefficient (PCC) of 0.993.

The rest of the paper is as follows. Section II contains a brief discussion of prior work and the necessary background. Section III presents the proposed method, and Section IV discusses the problem we analyzed and the dataset generation process. Section V discusses the evaluation metric and presents the experimental results. Finally, Section VI concludes the paper and presents a brief overview of future work.

2 BACKGROUND

2.1 Prior Work

Utilizing ML for modeling a receiver is a very active area of research as it solves multiple prevalent issues in the industry. Researchers aim to solve two issues: decreasing the simulation time to perform transient simulations and working around the complexity of developing IBIS-AMI models. Prior work falls into two major categories to accomplish the above tasks: one that predicts eye-diagram characteristics, and the other can predict waveforms and eye diagrams.

Kashyap et al. [6] do not explicitly model the receiver but determine the impact of channel parameters on the final eye diagram. Specifically, they perform cross-correlation amongst the different variables to remove redundant ones and streamline the learning for an artificial neural network. Trinchero and Canavero [15] use support vector machine (SVM) regression to predict the eye characteristics given specific channel parameters. Lu et al. [11] extend the work by demonstrating how a deep neural network (DNN) can model a high-speed channel and its performance gains over SVM regression for predicting eye characteristics. These studies though valuable, are limited in terms of the results as predicting eye characteristics only reveals information about 4 distinct points in the eye-opening and not much more. The mentioned studies reduce the problem's dimension using domain knowledge or dimensionality reduction and thus do not tackle the full-scale problem.

Other works look to model the receiver and its various components to predict the transient waveform. Choi and Cheng demonstrate a system identification (SID) approach for a high-speed serial link and handle three cases: a backplane channel, a redriver, and an active optical cable [3]. Each case creates individual models as each case presents different parameters. Li et al. further the work on the SID models by including the CTLE within a receiver using nonlinear SID models and using PCC to determine which time steps contribute toward inter-symbol interference (ISI) and disregard other time steps [9, 10]. The latter work reconstructs an eye diagram from the time series and demonstrates low error rates to the ground truth eye characteristics by simulating millions of bits. Though SID models have a significant advantage in faster training than other methods, they suffer because each filter or channel configuration requires an independent model, thus taking the same training time as a deep learning (DL) model.

DL-models, especially recurrent neural networks (RNNs), promise to effectively model the receiver's transient. Nguyen et al. use an RNN with long short-term memory (LSTM) cells to predict the receiver behavior and further the work by modeling the impact of non linearities introduced by the DFE [12, 13]. However, the studies do not go beyond a single pulse response to model the receiver with DFE taps and are forecasting future waveforms given a portion of a waveform from a SPICE simulator. Kashyap et al. show how GANs can predict receiver performance using an eye diagram; however, they limit their work to a fixed tap setting [8].

A significant drawback with the current approaches is that they are eye diagram based. As devices speed up, engineers refer to BER plots and bathtub curves as a better metric to evaluate a high-speed link, allowing them access to unique cases absent in an eye diagram. Kashyap et al. shows how a cGAN can recover a BER contour plot for varying DFE taps in a simulation environment [7].

2.2 Eye Diagrams and Bit-Error Rate (BER) Plots

Generally, engineers use eye diagrams, consisting of all possible transitions overlaid on top of each other, to provide insight into which transitions may present an issue or how jitter in the serial link plays a role in determining the eye-opening. The open area of the eye diagram is the region where one can sample the information without any issues. In an eye diagram, the horizontal axis represents the time, and the vertical axis represents the amplitude. Engineers often want a large open eye, indicating low amplitude and timing noise.

Though sufficient to relay information, it is impossible to capture high bit-error rates, the number of bits that arrive at the receiver in error per unit time, as low as 1×10^{-15} or 1 error in 10^{15} bits using a sampling oscilloscope. As a result, many engineers use bathtub curves and their 2D representations, BER contours plots, to measure the performance of the link [1].

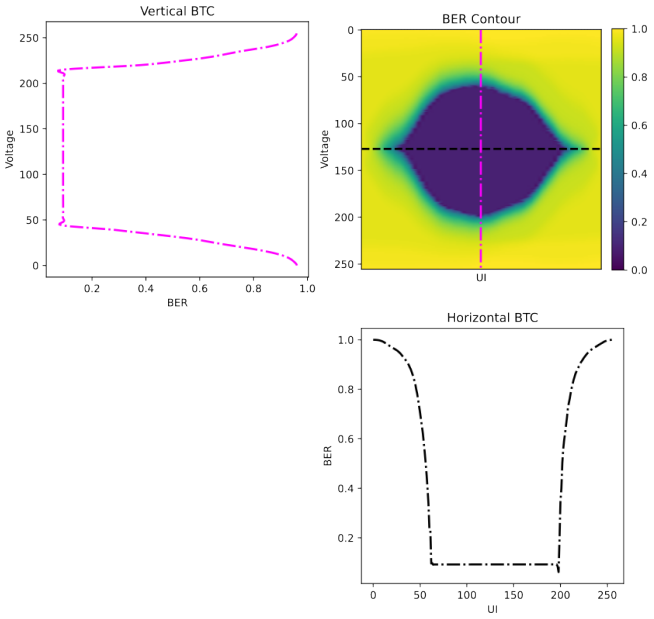


Figure 1: BER contour plot and corresponding horizontal and vertical bathtub curves.

Unlike eye diagram measurements, which require a sampling scope, we obtain bathtub curves and their corresponding BER contour plots by testing a device with a bit error rate tester (BERT). The underlying sampling strategy used by a BERT enables the BER contour plots to reveal edge cases, such as closed eyes that are otherwise not possible by a sampling scope, thus demonstrating its inherent advantage over the eye diagram [1]. In Figure 1, we show a BER contour plot and how the horizontal and vertical bathtub curves shown in black and pink, respectively, are embedded in the contour plot. The dark region at the center is where one can sample the data effectively with low errors, whereas as we move out towards the ends, the BERs increase as indicated by the green hue, and finally, the errors become large as indicated by the yellow hue.

2.3 Generative Adversarial Network (GAN)

Generative adversarial networks are crucial in synthetic data generation, image translation, and anomaly detection. A GAN typically consists of two modules: a generator that learns the distribution of the dataset by generating samples similar to the ground truth and a discriminator that distinguishes whether a sample presented to it is from the dataset. Together the two modules play a min-max game, where they try to outperform each other. Mathematically, the training loss function is as follows:

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

where x represents samples in the dataset and z is a random noise vector from which the generator learns. The first term in the equation is the ability of the discriminator to distinguish samples over the present in the dataset. The second term represents the ability of the generator to fool the discriminator over all possible z values.

cGANs are a variation of GANs where the generator's output is conditioned based on some known input, thereby making the problem a supervised learning problem. Prior work has found that passing the conditioning information to the discriminator increases the quality of the generated outputs [5]. With this in mind, Equation 1 changes to handle a conditioning variable y as follows:

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

Specific to image translation tasks, having an additional ℓ_1-loss term which compares the ground truth BER plot to the generators' improves the image quality. We weight the loss term by some factor λ, which is a hyperparameter. Thus, the final equation to train the cGAN is:

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

2.4 Gramian Angular Field (GAF)

Inspired by the prior work [4, 16] that encodes 1D time-series data as 2D images and subsequently uses them for various tasks such as classification and image translation, we do the same. In this work, we convert the receiver input waveform measurement to a 2D image representation, called a Gramian angular field (GAF) [16], and use GANs to make the task an image-to-image domain translation task. We use the GAF transform as it introduces new features not present in the original time series representations [4]. The GAF can effectively exploit the temporal correlation of the time-series data, whereas other transformations, such as Markov transition field (MTF), capture the transition dynamics instead [16]. Moreover, the ability to model a receiver requires knowing the temporal relationship between different time steps to model the impacts of the channel and ISI.

To implement a GAF, we rescale the time-series data such that the newly scaled values lie in the $[-1, 1]$ range. With the scaled time series, we express it in the polar coordinate system by taking the arccosine at each time step. At this point, one can construct a Gramian angular sum field (GASF) by taking the cosine of the trigonometric sum of each of the angles with each other. This process generates a $m \times m$ Gramian matrix, where m represents the number of time steps used. The full Gramian matrix is shown below:

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

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

3 METHOD

The Pix2Pix network [5], one of the prominent uses of a cGAN, serves as our starting point. We use a similar U-Net-based generator as the starting point but make modifications so that the model's output is conditional on the receiver input waveforms and the tap configuration. The encoder network in the U-Net learns the impact of the receiver waveform, whereas the decoder learns how to reconstruct the BER contour plot from the low dimensional

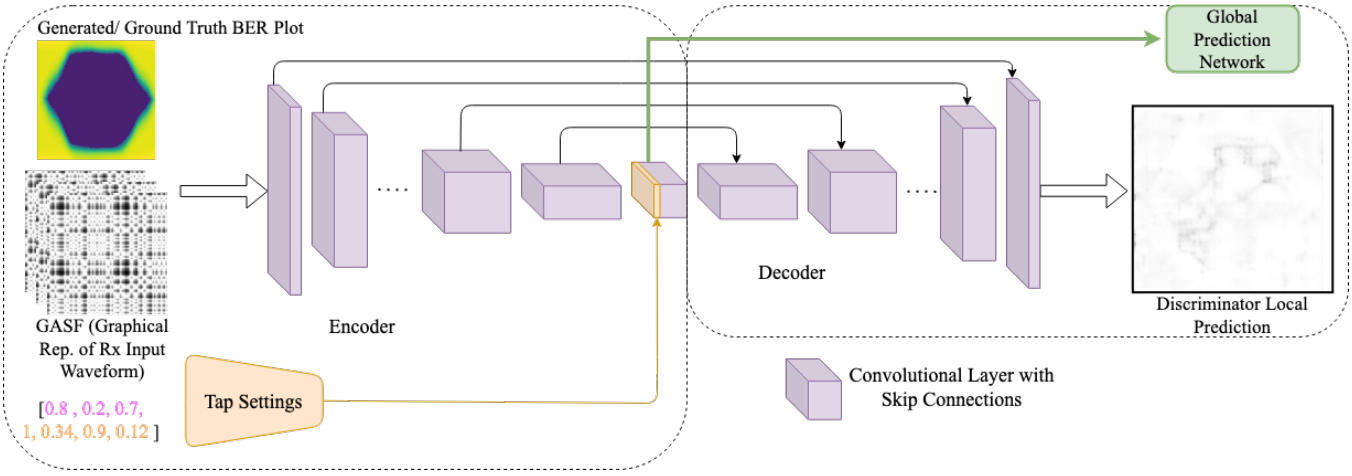


Figure 2: Discriminator architecture used for training all implementations. The discriminator takes in the concatenation of the GASF, taps values (DFE and CTLE) and either the synthetic or ground-truth BER plot and outputs a local pixel map along with a global prediction.

output of the encoder. The encoder model consists of convolution layers to downsample the image, whereas the decoder consists of ConvTranspose layers to upsample to the desired resolution. As the output BER plots depend on the tap setting of the DFE and CTLE, the model needs to be able to learn their impact. To accomplish this, we add a fully connected network whose output combines with that of the U-Net encoder. The decoder uses this combined representation as an input for reconstructing the BER contour plot. In this work, the U-Net encoder's output is a 300-element vector, and that of the tap network is a 40-element vector.

Figure 2 shows the discriminator model that we use in our training. Unlike the PatchGAN discriminator used in the Pix2Pix case, we opt to use a U-Net discriminator [14], which has a similar architecture to the generator described above. By doing so, the model predicts both the individual pixel level and an image level. These two prediction levels enable the discriminator to focus on global features and local details [14]. The global prediction gives a simple binary classification to indicate whether the combination of inputs is real or fake, whereas the local prediction is a binary prediction on each pixel location. The global and local predictions result from the input waveform, tap conditions, and the generated or synthetic BER contour plot. The pink tap values correspond to the CTLE tap, and orange corresponds to the DFE tap configurations. There is a network at the bottleneck to make a global prediction, and the decoder's output is the local pixel prediction.

We further illustrate how having a U-Net discriminator enables the generated BER plots to improve as training progresses in Figure 3. The generator uses the discriminator's local and global predictions as feedback to modify its predictions for the next iteration. However, we do not implement a hinge loss as a part of the training and utilize the regular loss functions because the results thus are sufficient for the task at hand.

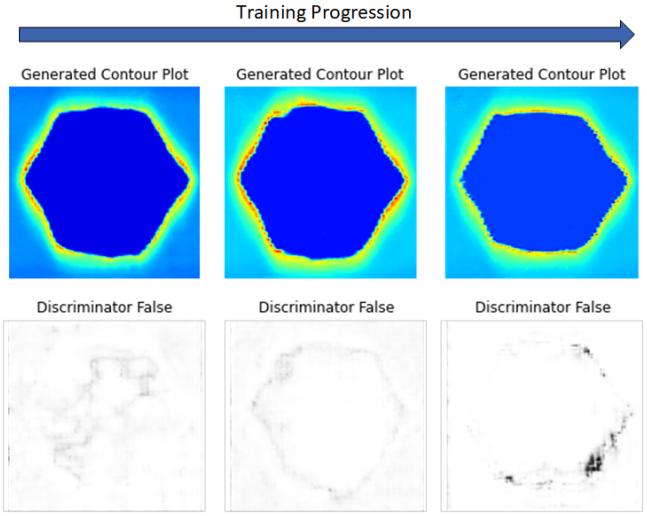


Figure 3: Evolution of the discriminator's local pixel prediction through the training process showing the discriminator's prediction over the output image as training progresses showing the different regions that the discriminator believes to be fake and how the generator uses that to minimize the differences.

4 DATASET CREATION

4.1 Data Collection

We connect our transmitter to a channel emulator for the measurements to configure different channel conditions with varying losses. Then the transmitter sends a 15 Gbps, pseudorandom bit stream (PRBS)-15, and a scope (20 ps sampling resolution) captures the

received waveform at the input to the receiver. With the preconfigured CTLE–DFE tap setting, and other receiver tuning taps fixed, the receiver's sampler sweeps the threshold voltage/sampling clock phase and generates the BER contour.

4.2 Data Preprocessing

We then randomly break up the original bitstream for each channel condition for as many tap configurations that exist per channel. After we split the receiver waveforms for each tap-BER present in the dataset, we transform them into a GASF, one of the GAN inputs. We limit the GASF generation to 256 timesteps because larger sizes become harder to fit into memory and increase the GAN training times. Due to the timestep limit, we further break each time series into multiple windows with overlap between consecutive windows. Thus, we can handle more extensive time series sequences and maintain the temporal relationship between different windows– Figure 2 further illustrates the different channels of the time series being input into the model.

The BER contour plots captured by the tool are of a resolution of 330×330, which we resize to 256×256 image used by the GAN as a target. To make the difference between different BER levels distinct, we take $\log_{10}$ of the BER plot and replace the open eye area with a positive value to allow for a gradual transition to areas where the eye-opening is not feasible. The last step of the preprocessing is to scale the data between $[0, 1]$ to help with GAN converging to the correct solution. Similarly, we scale each of the tap values so that they are within the same interval.

5 EVALUATION RESULTS AND COMPARISON

5.1 Evaluation Metric

Previous work that recovers an eye diagram or BER contour validates its results by comparing the ground truth with eye-opening characteristics or the horizontal bathtub curves at a single location [7, 8]. However, our evaluation metric combines the two to validate whether the eye-opening is correct and whether generated contour plots are consistent. To confirm the results, we take a slice of the BER contour plot at the center to get the horizontal and vertical bathtub curves to evaluate along. Thus, final metric is as follows:

$$Err = 0.5 \times RMSE_{CHR} + 0.5 \times RMSE_{ALL}, \quad (5)$$

where the $RMSE_{CHR}$ is the root mean squared error for both the eye height and width around the eye-opening and the $RMSE_{ALL}$ is the error for both the bathtub curves. As we cannot put the generated results into the measurement scope to evaluate them, we create an independent neural network trained on only the ground truth BER contours and their corresponding eye characteristics, referred to as a metric network. Then to show the similarity between the ground truth and generated BER contour plots, we use the metric network on the generated plot and determine how close its predicted eye characteristics are to that of the ground truths.

5.2 Results

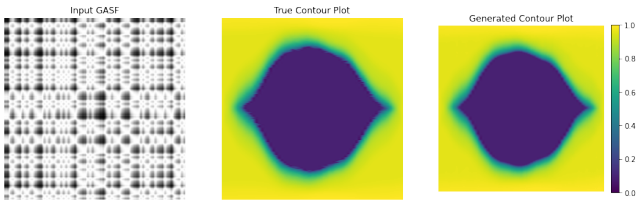
Figure 4 shows two sets of result from our test set for a high loss channel (Figure 4a) and a low loss channel (Figure 4b). The first column in Figure 4 shows the first windowed receiver input represented as a GASF, the second column shows the ground-truth BER plots and the final column shows the GAN generated contour plot. Simply looking at the ground truth and generated plot, it is difficult to tell the images apart. On closer inspection, though, the regions around the eye-opening differ due to the image resizing to the desired resolution. However, despite the difference, the overall features for the images are present in the correct locations.

For the same cases, Figure 5 shows the corresponding bathtub curves. The bathtub curves of the generated plot and the ground truth curve show a high correlation, as the PCC of the two curves is 0.993 across the test set. Additionally, the RMSE between the two curves over the entire testing dataset is 0.032, indicating a good fit for the generated plots.

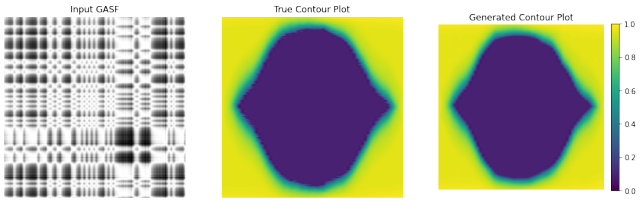
As mentioned previously, we use a neural network, the metric network, to evaluate specific eye characteristics, namely, eye height and width. Table 1 show the NN's ability to predict the characteristics mentioned above on the ground truth BER plots and the generated ones. The first row shows the metric network's prediction on the ground truth as the mean percent error and indicates a good fit for the model. The subsequent row shows the metric network's prediction on the generated plots. We observe that the eye characteristics of the generated plots using the metric network are

Table 1: Real vs generated contour plot statistics

	Eye Height % Error	Eye Width % Error
NN	0.70	0.85
Generated NN	1.59	1.12

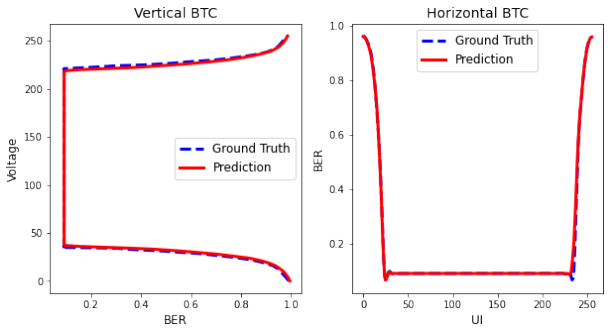


(a) High Loss Channel

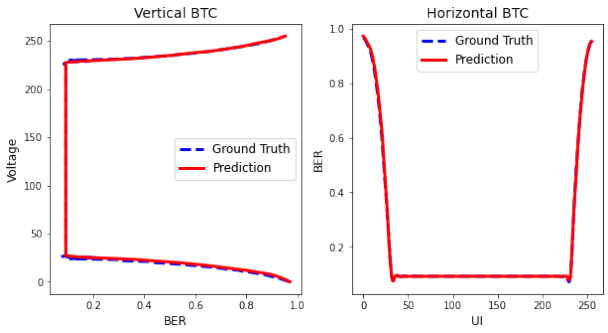


(b) Low Loss Channel

Figure 4: Results using the cGAN with the left indicating the GASF input, middle representing the ground-truth and right representing the cGAN-generated BER contour plot. The consistency between plots reveals that the cGAN is capable of handling different channel conditions, bitstreams and taps conditions.



(a) High Loss Channel



(b) Low Loss Channel

Figure 5: Horizontal and vertical bathtub curves from the BER contour plots shown in Figure 4. The accurately predicted bathtub curves demonstrate that the cGAN recovers BER plots where the underlying data is significant.

close to the ground truth and do not significantly increase errors. Furthermore, using the metric discussed in Section 5.1, we find the combined RMSE score on the test set to be 0.039, which indicates a good fit over the eye-opening as well as the bathtub curves along the center.

6 CONCLUSION

This paper presents a data-driven approach to modeling a high-speed SerDes receiver with the help of a cGAN. We demonstrate the model's ability to handle different bitstreams, channel conditions, and DFE and CTLE tap configurations. We show that the generated BER contour plots match the ground-truth BER plots visually, and both the vertical and horizontal bathtub curves at the center of each axis show a high correlation. Moreover, we analyze the eye characteristics with respect to the ground-truth BER plots and show that they are within 1.59% and 1.12% of the ground-truth eye height and width, respectively. The impact of signaling issues such as crosstalk and jitter can be explored in future work. Another avenue of exploration is related to whether an existing model can handle cases for different modulation schemes, namely, PAM-4 and NRZ.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. CNS 16-2137283 - Center for Advanced Electronics through Machine Learning (CAEML) and its industry members. We would like to give special thanks to Sumon Dey for his discussion on model optimizations.

REFERENCES

[1] 2019. Bridging the gap between BER and eye diagrams. https://download.tek.com/document/65W_26019_0_Letter.pdf

[2] H. Ahn, S. Baek, I. Madrigal, H. Zhang, J. Lai, M. Sapozhnikov, G. Zhang, A. Wong, and C. Borrelli. 2016. Novel Methodology of IBIS-AMI hardware correlation using trend and distribution analysis for high-speed SerDes system. (2016).

[3] Y. Choi and C. Cheng. 2017. High-speed link analysis with system identification approach. In *2017 IEEE International Symposium on Electromagnetic Compatibility Signal/Power Integrity (EMCSI)*. 741–744. https://doi.org/10.1109/ISEMC.2017.8077965

[4] N. Hatami, Y. Gavet, and J. Debayle. 2018. Classification of time-series images using deep convolutional neural networks. In *Tenth International Conference on Machine Vision (ICMV)*, Antanas Verikas, Petia Radeva, Dmitry Nikolaev, and Jianhong Zhou (Eds.), Vol. 10696. International Society for Optics and Photonics, SPIE, 242–249. https://doi.org/10.1117/12.2309486

[5] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5967–5976. https://doi.org/10.1109/CVPR.2017.632

[6] M. Kashyap, K. Keshavan, and A. Varma. 2017. A novel use of deep learning to optimize solution space exploration for signal integrity analysis. In *2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*. 1–3. https://doi.org/10.1109/EPEPS.2017.8329701

[7] P. Kashyap, Y. Choi, S. Dey, D. Baron, C.W. Wong, T. Wu, C. Cheng, and P. D. Franzon. 2022. Modeling of adaptive receiver performance using generative adversarial networks. In *2022 IEEE 72nd Electronic Components and Technology Conference (ECTC)*. 1958–1963. https://doi.org/10.1109/ECTC51906.2022.00307

[8] P. Kashyap, W. S. Pitts, D. Baron, C-W. Wong, and T. Wu P. D. Franzon. 2021. High speed receiver modeling using generative adversarial networks. In *2021 IEEE 30th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*. 1–3. https://doi.org/10.1109/EPEPS51341.2021.9609124

[9] B. Li, P. Franzen, y. Choi, and C. Cheng. 2018. Receiver behavior modeling based on system identification. In *2018 IEEE 27th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*. 299–301. https://doi.org/10.1109/EPEPS.2018.8534310

[10] B. Li, B. Jiao, M. Huang, R. Mayder, and P. Franzon. 2019. Improved system identification modeling for high-speed receiver. In *IEEE 28th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*. 1–3. https://doi.org/10.1109/EPEPS47316.2019.193212

[11] T. Lu, J. Sun, K. Wu, and Z. Yang. 2018. High-speed channel modeling with machine learning methods for signal integrity analysis. *IEEE Transactions on Electromagnetic Compatibility* 60, 6 (2018), 1957–1964. https://doi.org/10.1109/TEMC.2017.2784833

[12] T. Nguyen, T. Lu, J. Sun, Q. Le, K. We, and J. Schutt-Aine. 2018. Transient simulation for high-speed channels with recurrent neural network. In *2018 IEEE 27th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*. 303–305. https://doi.org/10.1109/EPEPS.2018.8534232

[13] T. Nguyen and J. Schutt-Aine. 2020. 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)*. 1–3. https://doi.org/10.1109/EPEPS48591.2020.9231402

[14] E. Schonfeld, B. Schiele, and A. Khoreva. 2020. A U-Net based discriminator for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8207–8216.

[15] R. Trinchero and F. G. Canavero. 2018. Modeling of eye diagram height in high-speed links via support vector machine. In *2018 IEEE 22nd Workshop on Signal and Power Integrity (SPI)*. 1–4. https://doi.org/10.1109/SaPIW.2018.8401679

[16] Z. Wang and T. Oates. 2015. Imaging time-series to improve classification and imputation. In *International Joint Conference on Artificial Intelligence (IJCAI)*. 3939–3945.