

High Speed Receiver Modeling Using Generative Adversarial Networks

Priyank Kashyap, W. Shepherd Pitts, Dror Baron, Chau-Wai Wong, Tianfu Wu, Paul D. Franzon
Electrical and Computer Engineering, North Carolina State University

Raleigh, USA

Email: {pkashya2, wspitts2, barondror, chauwai.wong, tianfu_wu, paulf}@ncsu.edu

Abstract—This paper presents a generative approach to modeling a high-speed receiver with a time series input. The model is not built with domain knowledge but learned from a wide range of channel conditions and input bitstreams to generate an eye diagram. The generated eye diagrams are similar to the simulated eye diagrams for the same scenario. We also developed a neural network model to evaluate the generated eye diagram’s relevant characteristics, such as eye height and width. The generated eye diagrams are within 7% and 3% error to the ground-truth in eye height and eye width, respectively, based on our evaluation neural network.

Index Terms—eye diagram, IBIS-AMI, generative model, generative adversarial network, GAN, receiver

I. INTRODUCTION

In a high-speed Serializer/Deserializer (SerDes), signal integrity has become increasingly difficult. A modern SerDes consists of a transmitter, a receiver, and a channel to communicate. Receivers consist of continuous time linear equalizers (CTLEs) and decision feedback equalizers (DFEs) to mitigate the effects of the channel, such as intersymbol interference (ISI). It is typical in the industry to use IBIS Algorithmic Modeling Interface (IBIS-AMI) models to enable vendors to share transmitter and receiver behavior without disclosing their intellectual property. Developing such IBIS-AMI models requires detailed circuit-level simulations and multiple design iterations, during which engineers use optimization algorithms to ensure that systems meet the desired specifications.

To address the problem of multiple design iterations when using IBIS models, designers have adopted techniques that utilize machine learning to model the behavior of a high-speed receiver and its subcomponents [1]–[3]. In [1] and [2], modeling a receiver is tackled by working with the time-series information and recovering the time-series information at the receiver’s output. However, Li et al. use nonlinear system identification models to recover the time series and then reconstruct an eye diagram[1], whereas Nguyen and Schutt-Aine recover only a single pulse response using a recurrent neural network (RNN) [3] uses a support vector machine to predict eye height based on the system’s parameters.

Though generative adversarial networks (GANs) have been used for image generation tasks, they have found use in the design flow, especially in lithography tasks considered image-based [4]. In Ye et al., a conditional GAN (cGAN) takes a mask pattern as input and predicts the corresponding resist pattern [4].

We present a cGAN based approach to modeling a receiver with time-series information with the above advancements in mind. The method incorporates a variety of input bitstreams from various channels and models the behavior of a receiver with fixed CTLE and DFE tap weights. We show that by converting a time-series waveform to an intermediate representation, we can condition the GAN and recover the eye diagram with high accuracy to the simulated eye diagram.

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

II. CONDITIONAL GAN FOR EYE DIAGRAM GENERATION

Inspiration is taken from prior work using cGAN to perform image to image translation, where the model transforms input images from one domain to another [5]. GANs consist of two modules, the generator G and the discriminator D playing a zero-sum game. Unlike regular GANs, the cGAN’s generator updates itself to be capable of creating output, $\hat{y} = G(z|x)$, that resembles the ground truth based on the input condition, x , combined with a random noise vector, z . Then, the discriminator updates itself to improve its capability of correctly identifying if a given output is from a dataset or the generator, based on the same input condition, x . To that extent, the objective function of the cGAN is as follows:

$$L_{\text{cGAN}}(G, D) = \mathbb{E}_{x,y} [\log D(y|x)] + \mathbb{E}_{x,z} [\log(1 - D(G(z|x)|x))]. \quad (1)$$

The dropout mechanism introduces the randomness of z in this implementation, as in Isola et al. [5].

As GANs have been successfully applied to image synthesis problems [5], we consider receiver modeling as an image problem. First, we convert the raw time series feeding into the receiver as an image by transforming it to a Gramian Angular Sum Field (GASF) [6]. To convert the time series to a GASF, we scale the original data in $[-1, 1]$. After scaling the time series, it is expressed in the polar coordinate system by taking the *arccosine* of the value at each time step. Once the time series is expressed in the polar coordinate system, the (i, j) th entry of the Gramian sum matrix is constructed by taking the

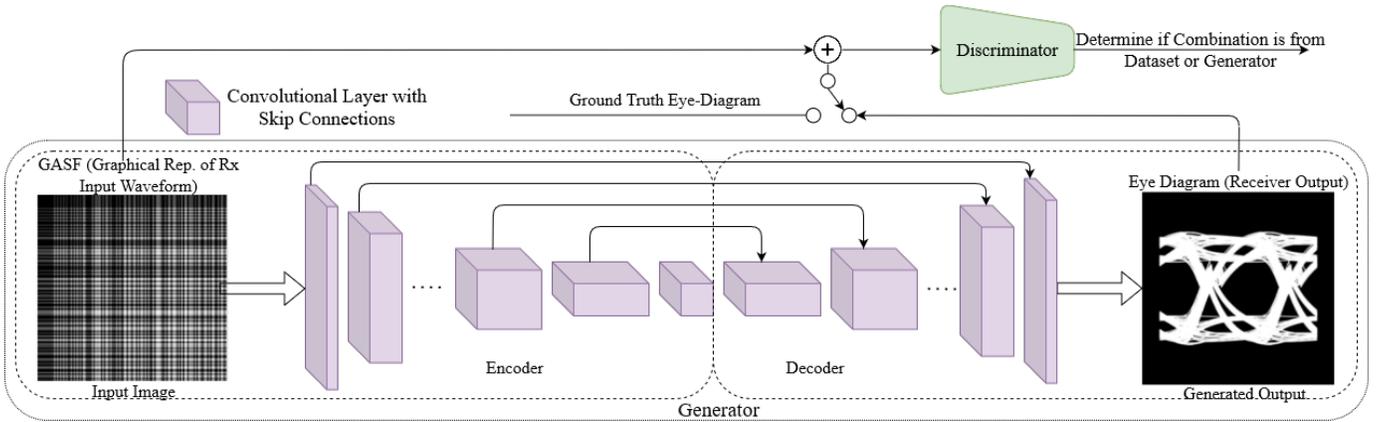


Fig. 1. GAN architecture used for training all implementations. The generator encoder and decoder with skip connections to forward information for the reconstruction of the eye diagram. The discriminator takes in the concatenation of the GASF and either the synthetic or ground-truth eye diagram.

trigonometric sum between cosine of the sum of the i th and the j th angular points. The GASF is then defined as:

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \dots & \cos(\phi_1 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \dots & \cos(\phi_n + \phi_n) \end{bmatrix} \quad (2)$$

where ϕ_n is the encoded angular value of the n th time step. Through the conversion process, the temporal dependency between time steps is preserved [6], which captures the ISI. Moreover, the time series problem is converted to an image problem and can use existing image algorithms.

With the GASF generated, the generator receives it and reconstructs an eye diagram based on the learned latent space. The generator is a U-Net-based network similar to that proposed by Isola et al. [5]. The model has two distinct parts, an encoder, and a decoder network. The encoder comprises convolution layers that downsample to lower dimensions to learn a latent space of the essential information about the eye diagram from the GASF. The decoder network generates the image of an eye diagram and consists of convolution transpose layers. There are skip connections between the encoder and the decoder to prevent information loss due to the bottleneck introduced by the encoder. In addition to the objective function in (1), the generator has an ℓ_1 loss function to enforce the sparsity with the eye diagram reconstruction.

We then describe the discriminator, which gets either the ground-truth or the synthetic images, and attempts to discern whether it is a ground-truth eye diagram or a synthetic eye diagram. Unlike making a binary prediction based on the entire image, the discriminator is a PatchGAN [5], which outputs different patches corresponding to regions of the image and helps recover the global features from the image. The cGAN model is detailed in Fig. 1.

A challenging task associated with generative image modeling is to quantify the resultant images. Metrics such as Fréchet inception distance (FID) determine how similar the actual and generated images are. In this case, we are concerned with the image similarity and the relevant eye characteristics. To achieve this, we use a pre-trained deep neural network (DNN) to evaluate the quality of the generated images in terms of

eye diagram statistics. The network consists of a series of convolutional and max-pooling layers, after which there is a flattening layer before an output layer. The output layer is a dense layer, where the numbers of neurons correspond to the desired eye characteristics to be used to evaluate the synthetic eye diagrams. This DNN aims to achieve comparable performance in terms of the eye characteristics given the ground-truth eye diagrams. The DNN is trained separately and is used to evaluate the generated eye-diagram images.

III. DATASET CREATION

The original dataset consists of time-series waveforms from Cadence Virtuoso using VerilogA models for a SerDes running at 5 Gb/s. We collect the waveforms coming off the transmission line and after the DFE in the receiver, with both waveforms sampled at 5 ps intervals. We collect the eye diagram statistics from Virtuoso, such as eye height and eye width, during the data collection. We then randomly split the waveforms for the duration of the experiments in 60, 20, and 20 splits for the training, validation, and testing sets, respectively. For our experiments, we collect a total of 2,000 different bitstreams and channel variations. Both models are trained on the same data and evaluated on the same test sets.

After collecting the dataset, we preprocess the waveforms to their corresponding GASF. For this purpose, we downsample the input waveforms to the receiver by selecting a sample at 50 ps intervals allowed by the Nyquist frequency. We then transform the downsampled waveform to create a GASF of size 256×256 . The waveforms captured at the receiver's output are processed to create eye diagrams by overlapping 2-bit periods (UIs) on top of each other for the entirety of the waveform. The preprocessing flow for the data takes 223 ms for one eye diagram generation and 30 ms for each GASF.

Figure 1 shows an input GASF used to condition the GAN and generate an eye diagram. Lastly, before we use the eye diagram statistics for the metric DNN, we rescale the values between $[0, 1]$ to improve the convergence of the network.

IV. EXPERIMENTAL RESULTS

Fig. 2 shows good training and validation performance. For evaluation purposes, we select the cGAN model during the

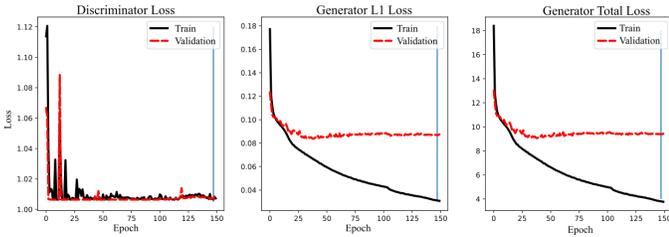


Fig. 2. The loss functions corresponding to the discriminator’s loss, generator’s ℓ_1 loss, and the total generator loss. The vertical line indicated that models used for evaluation were saved at the 148th epoch. Overall, the generator has good generalization between the training and validation sets.

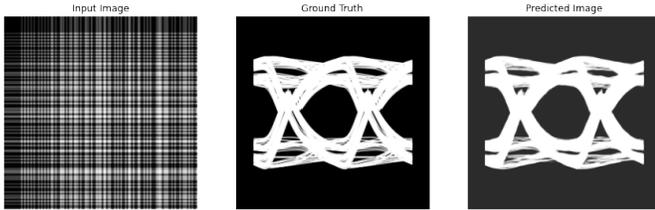


Fig. 3. The GASF representation of the time series, the corresponding ground-truth eye diagram, and the generator’s predicted eye diagram for a 210×150 image with zero paddings.

last training iteration, where the discriminator misclassifies the synthetic eye diagrams from the validation set to be from the dataset. The converged discriminator loss shows that the generator can fool the discriminator successfully, indicating that the generated eye diagrams are similar to the ground truth. As is evident in Fig. 3 and Fig. 4, the generator can accurately recover the complete eye diagram. More importantly, the eye openings of the generated images are similar to that of the ground truth based on visual inspection. However, the generator struggles to recover the individual lines in the generated images.

We quantified our results by evaluating the ground truth and synthetic eye diagrams through our metric-based DNN. In TABLE I, *NN* refers to the metric DNN’s performance on the ground-truth eye diagram and *Generated NN* refers to the metric DNN’s performance on the synthetic eye diagrams. The table shows that the DNN can predict the eye height and width within 2% error to the ground truth for both metrics. Moreover, on the synthetic images from the cGAN, it can be observed that the eye height and width are within 7% and 3% error to the ground truth, respectively.

To achieve this, our model takes 63s for one training iteration of the GAN for the 256×256 sized images, and images smaller than 256×256 are zero-padded to that size. Due to lower images being zero-padded, the network training time is unchanged. However, for higher resolution images at 512×512 , the training time is 113s due to the increase of the input image parameters. We observe that the metric DNN trained on that resolution has similar performance for the different resolution images, as seen in TABLE I. Notably, the inference time for the models at the discussed resolutions is under 1s.

V. CONCLUSION

This paper presents a method for creating a generative model for a SerDes receiver with relatively little data. We

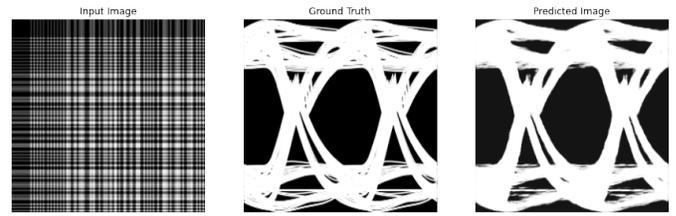


Fig. 4. The GASF representation of the time series, the corresponding ground-truth eye diagram, and the generator’s predicted eye diagram for a full 256×256 image.

TABLE I
REAL VS SYNTHETIC EYE-DIAGRAM STATISTICS

	Eye Height		Eye Width	
	mV	% Error	ps	% Error
Ground Truth	114.9	-	140.5	-
NN	116.6	1.5	138.1	1.7
Generated NN	122.1	6.3	137.2	2.3

achieve this by representing the time series data in the two-dimensional form of a GASF, which makes the modeling a domain-transfer task. The training data is diverse in terms of the bitstreams and channel conditions used. Moreover, we have shown that the images are similar based on visual inspection, and to further quantify the results, we have introduced a neural network to determine the eye characteristics. The eye height and width predictions are within 7% and 3% error, respectively, based on the neural network model. Through our proposed data-driven approach, the overhead required by engineers to design an IBIS-AMI model is reduced, and the time that is taken to run simulations is comparable to other data-driven receiver models.

In future work, the impact of nonlinearities introduced in the receiver by the CTLE and DFE will be factored into the modeling methodology.

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. We would like to give special thanks to Chris Cheng, Yongjin Choi, and Sumon Dey from HPE for their support.

REFERENCES

- [1] 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.
- [2] 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.
- [3] R. Trinchero and F. G. Canavero, “Modeling of eye diagram height in high-speed links via support vector machine,” in *IEEE 22nd Workshop on Signal and Power Integrity (SPI)*, 2018, pp. 1–4.
- [4] 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.
- [5] 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.
- [6] Z. Wang and T. Oates, “Imaging time-series to improve classification and imputation,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015, pp. 3939–3945.