chips

MDPI

Article

Machine-Learning-Assisted Mueller–Müller Phase Detection for Robust PAM4 Clock and Data Recovery

Soumobrata Ghosh * and Steven Bibyk

Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, USA
* Correspondence: ghosh.217@osu.edu

Abstract

High-speed PAM-4 SerDes and wireline interconnects used in data-center, artificial intelligence (AI), high-performance computing (HPC), chip-to-chip, and high-speed networking applications require low-power and hardware-efficient clock and data recovery (CDR) capable of maintaining accurate timing under channel impairments, jitter, and nonlinear phase-detector characteristics. Although the Mueller–Müller phase detector (MMPD) is widely adopted for baud-rate CDR due to its simplicity and robustness, its nonlinear gain characteristic introduces residual timing errors that limit tracking accuracy and increase jitter sensitivity. This work presents a hardware-efficient AI-assisted residual MMPD for baud-rate PAM-4 CDR systems. Instead of replacing the conventional phase detector, a lightweight fully connected neural network learns only the deterministic nonlinear residual correction and generates a compensation term that is added to the classical MMPD output. The proposed residual network employs a compact 1–8–8–1 architecture with ReLU activation, containing only 97 trainable parameters and enabling efficient fixed-point RTL and ASIC implementation. The proposed detector is integrated and evaluated within a complete baud-rate PAM-4 digital CDR comprising a baud-rate sampler, PAM-4 slicer, proportional–integral loop filter, and digitally controlled oscillator, while preserving the stability and low-complexity characteristics of the conventional architecture. Simulation results demonstrate that the proposed detector reduces the phase-detector RMSE from 0.2088 UI to 0.0073 UI, extends the effective linear operating range from approximately ±0.12 UI to ±0.45 UI, and achieves a 13.45% reduction in closed-loop RMS timing error compared with the classical MMPD while maintaining comparable detector noise performance. Owing to its minimal computational overhead, improved detector linearity, compatibility with existing baud-rate CDR architectures, and straightforward hardware realization, the proposed residual-learning framework provides a practical and scalable solution for AI-assisted timing recovery in next-generation high-speed PAM-4 wireline and SerDes receivers.

Keywords: clock and data recovery (CDR); Mueller–Müller phase detector (MMPD); PAM4 signaling; artificial intelligence; timing recovery; inter-symbol interference (ISI); jitter compensation; SerDes receivers

Check for updates

Academic Editors: Gianluca Giustolisi and Elisabetta Moisello

Received: 23 July 2026
Revised: 23 September 2026
Accepted: 24 September 2026
Published: 1 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The continuous growth of cloud computing, AI, HPC and data-center interconnects has significantly increased the demand for high-speed serial communication links capable of delivering higher bandwidth while maintaining low power consumption and implementation complexity. To satisfy these requirements, modern wireline transceivers have increasingly transitioned from conventional non-return-to-zero (NRZ) signaling toward multi-level

pulse amplitude modulation (PAM4), which doubles the transmitted data rate without increasing the channel bandwidth [1,2]. Although PAM4 substantially improves spectral efficiency, its reduced eye opening and increased susceptibility to inter-symbol interference (ISI), noise, and jitter make reliable clock and data recovery (CDR) considerably more challenging than in binary systems.

CDR remains one of the most critical building blocks of high-speed serial receivers because the recovered sampling clock directly determines timing accuracy and ultimately limits the achievable bit-error rate (BER). Early work on timing recovery established the theoretical foundations of symbol-rate synchronization using decision-directed and minimum mean-square error (MMSE) timing functions [3,4]. These principles later evolved into practical digital and mixed-signal CDR architectures suitable for multi-gigabit communication systems [5,6]. As serial data rates continued to increase, receiver architectures incorporated sophisticated equalization techniques including feed-forward equalization (FFE), continuous-time linear equalization (CTLE), and decision-feedback equalization (DFE), together with baud-rate timing recovery, to compensate for severe channel losses while maintaining acceptable power efficiency [7–10]. Behavioral and macro-modeling techniques have also become increasingly important for the design and verification of high-speed CDR systems, enabling rapid architectural exploration and mixed-signal system-level validation prior to transistor-level implementation [11,12].

Among the numerous timing recovery algorithms proposed over the past two decades, the Mueller–Müller phase detector (MMPD) remains one of the most widely adopted solutions for baud-rate CDR. Unlike oversampling architectures that require multiple clock phases and additional edge samples, the MMPD estimates timing error using only one sample per symbol together with symbol decisions, resulting in significantly lower hardware complexity, reduced clock distribution requirements, and excellent scalability to very high data rates [3,13]. Consequently, the MMPD has become a preferred timing detector for many modern PAM4 wireline and SerDes receivers.

Despite these advantages, the application of the classical MMPD to multilevel signaling introduces several well-known limitations. The nonlinear detector characteristic caused by multilevel symbol transitions produces a timing function whose gain varies with phase error and may contain multiple stable operating points. These characteristics increase phase acquisition time, reduce jitter tolerance, and under certain channel conditions may even lead to false locking behavior [13,14]. Additional degradation arises from residual ISI, data-dependent jitter (DDJ), additive noise, and imperfect symbol decisions, all of which perturb the detector output and reduce timing accuracy in practical PAM4 communication systems [15,16]. Recent high-speed CDR implementations have proposed numerous architectural enhancements including adaptive timing recovery, transition elimination, hybrid equalization, reference-less operation, collaborative CDR loops, and decision-directed MMSE timing recovery to improve robustness and energy efficiency [1,2,8,9,15–17]. Although these approaches improve specific aspects of timing recovery, they generally increase hardware complexity, require additional circuitry, or address only particular operating conditions.

Several improvements have also been proposed specifically for the Mueller–Müller detector. In particular, Musah et al. analyzed the nonlinear timing characteristics of the conventional Type-A detector and proposed a modified detector employing a fixed correction coefficient to suppress undesirable lock points and improve convergence [13]. While this approach improves acquisition behavior, the correction remains fixed over the entire operating range and therefore cannot accurately compensate the nonlinear detector characteristic under varying channel conditions, ISI levels, and timing offsets.

Recent advances in machine learning have demonstrated that compact neural networks can efficiently approximate nonlinear functions while maintaining modest computational

complexity, making them increasingly attractive for hardware-assisted communication systems. However, most AI-based communication receivers either replace conventional signal processing blocks entirely or require substantially larger models and online adaptation, limiting their practicality for high-speed ASIC implementation.

In this work, we propose a hardware-efficient *AI-assisted residual Mueller–Müller phase detector* for baud-rate PAM4 clock and data recovery. Rather than replacing the conventional timing detector, the proposed architecture preserves the classical MMPD and augments it with a lightweight fully connected neural network that learns only the nonlinear residual correction. This residual-learning formulation significantly reduces the learning complexity while maintaining compatibility with existing CDR architectures and preserving the stability of the conventional detector. The proposed network employs a compact 1–8–8–1 architecture containing only 97 trainable parameters and performs inference only during operation, making it well suited for fixed-point ASIC implementation.

The proposed AI-assisted phase detector is integrated into a complete digital CDR incorporating a proportional–integral loop filter and a digitally controlled oscillator. Behavioral simulations show improved timing-characteristic linearity and reduced residual timing error relative to both the classical Mueller–Müller detector and the previously reported fixed-β implementation. These improvements are achieved with modest implementation complexity, indicating that residual machine-learning correction can enhance conventional timing recovery without requiring a fundamentally different CDR architecture.

The primary contributions of this work are fourfold. First, an AI-assisted residual Mueller–Müller phase detector is introduced for baud-rate PAM4 clock and data recovery, in which a lightweight neural network learns the nonlinear correction to the conventional detector output rather than replacing the detector itself. Second, a compact fully connected 1–8–8–1 neural network with only 97 trainable parameters is developed to enable low-complexity nonlinear correction suitable for fixed-point ASIC implementation. Third, a bounded residual correction mechanism is employed to constrain the learned contribution and retain the stability and operating characteristics of the underlying classical detector. Finally, the proposed approach is verified within a closed-loop digital CDR and benchmarked against the classical Mueller–Müller and fixed-β detectors, demonstrating improved timing-recovery performance with limited additional implementation complexity.

2. Proposed AI-Assisted Residual Mueller–Müller Phase Detector

2.1. Classic MMPD—Working Principle and Limitations

The Mueller–Müller phase detector (MMPD) is one of the most widely adopted baud-rate timing error detectors for high-speed clock and data recovery (CDR) systems due to its simplicity, decision-directed operation, and requirement of only one sample per unit interval (UI). Unlike oversampling phase detectors that require additional edge samples and multiple clock phases, the MMPD performs timing recovery using only the present and previous baud-rate samples together with their corresponding symbol decisions. This significantly reduces hardware complexity, clock distribution, and power consumption, making the MMPD particularly attractive for high-speed PAM4 wireline receivers.

Let $x[n]$ denote the received equalized sample at the n^{th} symbol interval and $d[n]$ denote the corresponding sliced symbol decision. For PAM4 signaling,

$$d[n] \in \{-3, -1, +1, +3\}. \tag{1}$$

The classical Mueller–Müller timing error detector is given by

$$e_{\mathrm{MM}}[n] = d[n-1]x[n] - d[n]x[n-1]. \tag{2}$$

For a fixed timing-phase offset ϕ, the average MMPD characteristic is defined as $\bar{e}_{\mathrm{MM}}(\phi) \triangleq E\{e_{\mathrm{MM}}[n] \mid \phi[n] = \phi\}$.

The detector estimates the timing error by comparing the correlation between the current received sample and the previous symbol decision with that between the previous sample and the current decision. When the recovered clock samples the received waveform early, one correlation term dominates and the detector produces an error of one polarity. Conversely, if sampling occurs late, the polarity reverses. Consequently, the average detector output changes sign at the optimum sampling instant, allowing the CDR loop to align the recovered clock with the center of the data eye.

Ideally, the average timing characteristic should satisfy

$$\bar{e}_{\mathrm{MM}}(\phi) \approx K_{\mathrm{PD}}\phi. \tag{3}$$

where ϕ is the timing phase error and K_{PD} is the phase detector gain. A linear relationship provides nearly constant detector gain, simplifying loop design and improving acquisition and jitter tracking. However, practical PAM4 channels deviate significantly from this ideal behavior.

Recent work by Musah et al. [13] showed that the classical Mueller–Müller Type-A detector exhibits undesirable timing characteristics when applied to multilevel PAM signaling. In particular, the timing function may contain multiple stable zero-crossings due to weak correlation between transmitted and received symbols in transition regions. Consequently, the CDR may converge to an incorrect sampling phase, resulting in false lock or significantly increased acquisition time.

To improve convergence, Musah proposed a modified timing detector by introducing a data-dependent correction term,

$$e_{\beta}[n] = e_{\mathrm{MM}}[n] + \beta g[n], \tag{4}$$

where $g[n]$ is a correction function and β controls its contribution. The additional term suppresses undesirable lock points and improves phase acquisition. Nevertheless, the correction is governed by a single fixed coefficient. While an appropriately chosen β improves convergence, it cannot accurately compensate the nonlinear timing characteristic over the entire operating range. Furthermore, increasing β generally introduces a trade-off between acquisition robustness and steady-state jitter performance.

Despite its widespread adoption, the classical MMPD possesses several fundamental limitations in high-speed PAM4 systems.

First, the detector characteristic exhibits a limited linear operating region and becomes nonlinear at larger timing offsets [18]. Consequently, the effective phase-detector gain varies with timing offset, complicating loop bandwidth optimization and degrading jitter performance.

Second, the detector is highly sensitive to residual inter-symbol interference (ISI). Since the timing error in (2) depends directly on the received samples, any residual ISI, channel distortion, or equalization error perturbs the timing estimate. Two identical timing offsets may therefore produce different detector outputs depending on the surrounding data pattern.

Third, PAM4 signaling significantly increases timing recovery difficulty. Compared with NRZ, PAM4 has approximately one-third of the vertical eye opening, making the detector substantially more susceptible to noise, data-dependent jitter, slicer errors, and amplitude distortion. Because the MMPD is decision-directed, incorrect symbol decisions immediately propagate into the timing error calculation.

Finally, existing improvements such as fixed-β correction, coarse quantization, transition elimination, or dedicated training sequences improve only specific aspects of detector

performance while sacrificing detector gain, robustness, or implementation simplicity. Since these approaches rely on predetermined correction mechanisms, they cannot adapt to the nonlinear detector distortion introduced by varying channel conditions, ISI, and noise.

These observations motivate the proposed AI-assisted residual MMPD. Instead of replacing the classical detector, the proposed architecture preserves the conventional Mueller–Müller timing estimator while learning only its nonlinear residual correction. The correction $\Delta e_{\mathrm{NN}}[n]$ is generated by a lightweight neural network. By estimating only the residual correction, the proposed detector preserves the hardware efficiency and proven stability of the conventional MMPD while compensating for its nonlinear timing characteristic, ISI-dependent distortion, and the limitations of fixed-coefficient correction techniques.

2.2. Proposed Residual AI-Assisted MMPD

To overcome the nonlinear detector characteristic of the classical Mueller–Müller phase detector while preserving its simplicity and hardware efficiency, this work proposes an AI-assisted residual MMPD. Rather than replacing the conventional timing detector with a completely data-driven model, the proposed architecture augments the classical detector using a lightweight neural network that learns only the nonlinear residual error. This residual-learning strategy retains the proven stability and low computational complexity of the conventional MMPD while significantly improving the linearity of its timing characteristic.

Figure 1 illustrates the overall architecture of the proposed detector. The equalized baud-rate samples are first processed by the conventional MMPD to generate the classical timing error estimate. Simultaneously, the same timing information is supplied to a compact feed-forward neural network, which estimates a residual correction. The final detector output is obtained by summing the classical timing error and the learned residual before being applied to the loop filter.

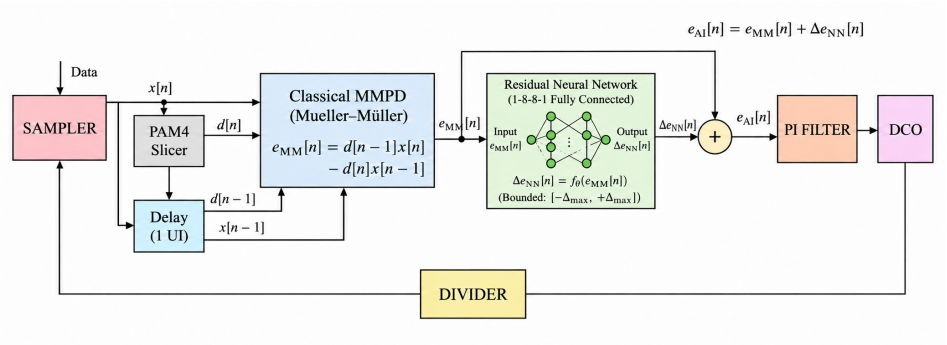


Figure 1. Proposed Phase Detector.

Let $x[n]$ denote the equalized baud-rate sample and $d[n]$ denote the corresponding PAM4 decision. Although the classical MMPD timing error defined in (2) provides an effective timing estimate, its average detector characteristic deviates from the desired linear response under practical PAM4 channel conditions. Ideally, the detector output should satisfy

$$e_{\mathrm{ideal}}(\phi) = K_{\mathrm{PD}}\phi, \tag{5}$$

where ϕ is the timing phase offset and K_{PD} is the desired detector gain. Instead, the classical detector produces a nonlinear characteristic

$$\bar{e}_{\mathrm{MM}}(\phi) \neq K_{\mathrm{PD}}\phi. \tag{6}$$

primarily due to residual ISI, multilevel symbol transitions, and decision-dependent distortion. Consequently, the nonlinear residual error can be defined as

$$\Delta e_{\text{target}}(\phi) = e_{\text{ideal}}(\phi) - \bar{e}_{\text{MM}}(\phi) = K_{\text{PD}}\phi - \bar{e}_{\text{MM}}(\phi). \tag{7}$$

Instead of directly estimating the timing phase, the proposed neural network learns the residual target defined in (7). The final detector output therefore becomes

$$e_{\text{AI}}[n] = e_{\text{MM}}[n] + \Delta e_{\text{NN}}[n], \tag{8}$$

where $\Delta e_{\text{NN}}[n]$ is the neural network prediction. Since the conventional MMPD remains the primary timing detector, the neural network acts only as a nonlinear correction stage. This residual-learning formulation significantly reduces the learning complexity compared to predicting the entire timing error directly while preserving the intrinsic operating characteristics of the classical detector.

Unlike previous fixed-β correction methods, which apply a constant correction coefficient to every operating condition, the proposed approach learns a continuous nonlinear mapping between the detector input and the required residual correction. Consequently, the correction automatically varies with timing offset, allowing the detector characteristic to closely approximate the ideal linear response over the complete operating range.

For hardware efficiency, a compact fully connected feed-forward neural network is employed. Since the objective is to estimate only the residual detector error, the network complexity can be kept extremely small without sacrificing accuracy. The proposed implementation uses a 1-8-8-1 architecture consisting of two hidden layers with ReLU activation functions and a linear output neuron. The network contains only 97 trainable parameters, making it suitable for ASIC implementation using fixed-point arithmetic. After offline training, only inference is performed during CDR operation, eliminating any online adaptation or iterative optimization.

The network input consists of the classical timing estimate generated by the MMPD,

$$u[n] = e_{\text{MM}}[n], \tag{9}$$

while the network output corresponds to the residual correction

$$\Delta e_{\text{NN}}[n] = f_\theta(e_{\text{MM}}[n]), \tag{10}$$

where $f_\theta(\cdot)$ represents the trained neural network. To guarantee robust CDR operation, the predicted correction is bounded according to

$$-\Delta_{\max} \leq \Delta e_{\text{NN}}[n] \leq +\Delta_{\max}. \tag{11}$$

thereby preventing excessive correction that could disturb loop stability.

The neural network is trained offline using synthetically generated PAM4 data that includes practical channel ISI, additive white Gaussian noise, and controlled timing offsets spanning the expected acquisition range. For each timing offset, the classical MMPD characteristic is first computed, after which the desired residual is obtained from (7). The network parameters are optimized by minimizing the mean-squared error between the predicted residual and the desired residual. Once trained, the network weights are quantized and exported for fixed-point hardware implementation.

The proposed residual-learning framework offers several important advantages. First, only the nonlinear error is learned, allowing a very small neural network to achieve high accuracy. Second, the conventional MMPD continues to provide the primary timing estimate, ensuring compatibility with existing baud-rate CDR architectures. Third, the bounded residual correction maintains predictable detector behavior and simplifies hardware verification. Finally, because the learned correction is nonlinear rather than coefficient-based,

the detector can compensate timing errors that cannot be addressed using conventional fixed-β methods.

The resulting AI-assisted residual MMPD therefore combines the robustness and simplicity of the classical Mueller–Müller detector with the approximation capability of neural networks, providing an improved detector characteristic while introducing only minimal computational and hardware overhead. The integration of this detector into the complete baud-rate CDR loop is presented in the following section.

2.3. Neural Network Architecture and Training

The objective of the proposed neural network is not to replace the classical Mueller–Müller phase detector, but rather to estimate its nonlinear residual error. Since the conventional MMPD already provides an accurate first-order timing estimate, the learning problem is significantly simplified by predicting only the residual correction required to linearize the detector characteristic. Consequently, the neural network can remain extremely compact while achieving high approximation accuracy.

The residual target is obtained from the difference between the ideal linear detector characteristic and the average output of the classical MMPD, as defined in (7). For a fixed timing-phase offset ϕ, the average neural-network correction is defined as $\overline{\Delta e}_{\mathrm{NN}}(\phi) \triangleq E\{\Delta e_{\mathrm{NN}}[n] \mid \phi[n] = \phi\}$. The neural network is trained such that

$$\overline{\Delta e}_{\mathrm{NN}}(\phi) \approx \Delta e_{\mathrm{target}}(\phi). \tag{12}$$

During inference, the instantaneous neural-network correction is added to the classical MMPD output according to (8), thereby compensating for the nonlinear detector distortion while preserving the conventional MMPD operating principle.

A lightweight fully connected feed-forward neural network was selected to minimize hardware complexity and facilitate ASIC implementation. The proposed network consists of one input neuron, two hidden layers containing eight neurons each, and one linear output neuron, resulting in a compact 1–8–8–1 architecture. The hidden layers employ Rectified Linear Unit (ReLU) activation functions, while the output layer uses a linear activation to generate the residual timing correction. To evaluate the effect of network size, residual networks with 4, 8, and 16 neurons per hidden layer were also examined under identical training conditions. Increasing the hidden-layer width from 4 to 8 neurons produced a substantial improvement in residual approximation accuracy, whereas increasing the width further from 8 to 16 neurons resulted in only a minor additional improvement while significantly increasing the number of trainable parameters and associated hardware complexity. Therefore, the 1–8–8–1 architecture was selected as a practical compromise between approximation accuracy and implementation complexity. The selected network contains only 97 trainable parameters, including weights and biases, making it substantially smaller than conventional deep learning models while remaining sufficiently expressive to approximate the nonlinear detector characteristic.

The network input is the normalized classical timing estimate generated by the MMPD. Prior to training, the input data are standardized according to

$$x_{\mathrm{norm}} = \frac{x - \mu_x}{\sigma_x}, \tag{13}$$

where μ_x and σ_x denote the mean and standard deviation of the training dataset, respectively. Normalization prevents saturation of the hidden neurons and improves convergence during optimization.

Training data were generated entirely offline using a behavioral PAM4 communication model. Random PAM4 symbols were pulse-shaped using a root-raised cosine filter and transmitted through a practical ISI channel. Additive white Gaussian noise corresponding to

an SNR of 22 dB was added to represent a moderate-noise receiver operating condition while preserving the underlying nonlinear MMPD characteristic required for residual learning. The network was trained only once at this SNR and was not retrained for subsequent evaluations. In particular, the same trained network was later evaluated within the complete closed-loop CDR at an SNR of 28 dB, providing evidence that the learned residual correction is not restricted to the exact SNR used during training. The timing offset was swept from −0.45 UI to +0.45 UI, producing the average classical MMPD characteristic over the entire acquisition range. The residual target in (7) was then computed for every timing offset and used as the regression label during training. Since the neural network is trained only once offline, no online adaptation or retraining is required during CDR operation.

Network optimization was performed using the Adam optimizer with a learning rate of 5×10^{-4}, a mini-batch size of 64, and 500 training epochs. The objective function minimizes the mean-squared error (MSE) between the predicted residual and the desired residual,

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(\Delta e_{\text{target},i} - \Delta e_{\text{NN},i} \right)^2. \tag{14}$$

Figure 2 illustrates the training convergence of the proposed network. The upper plot shows the root-mean-square error (RMSE), while the lower plot presents the corresponding loss function. A rapid reduction in both metrics is observed during the initial iterations, indicating that the dominant nonlinear characteristics of the detector are learned quickly. As training progresses, both the RMSE and the loss gradually converge to near-zero values with minimal oscillation, demonstrating stable optimization and the absence of divergence. The smooth convergence behavior indicates that the selected network capacity is sufficient to approximate the residual detector characteristic without requiring a significantly larger model.

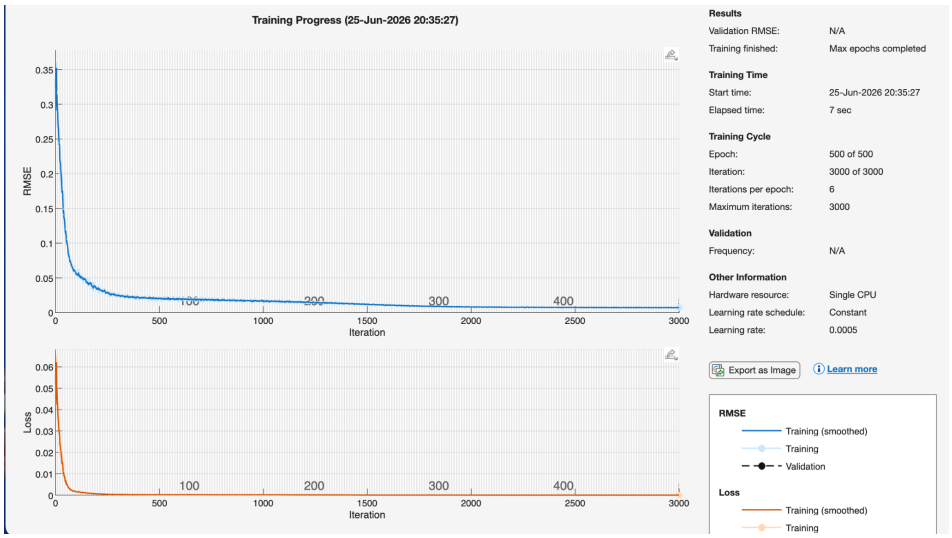


Figure 2. Training Curve.

After training, the floating-point network parameters are quantized into fixed-point representations for hardware implementation. The weights, biases, normalization parameters, and output limits are exported directly to the hardware inference engine, allowing the proposed AI-assisted residual MMPD to perform only feed-forward inference during real-time operation. Because the network is both compact and inference-only, its computational overhead is minimal while providing an accurate nonlinear correction to the classical Mueller–Müller detector.

3. Hardware Implementation

3.1. Hardware Realization of AI-Assisted Phase Detector

To demonstrate the hardware feasibility of the proposed AI-assisted Mueller–Müller phase detector, the trained residual neural network was translated into synthesizable Verilog RTL using fixed-point arithmetic. Training is performed offline, after which the network weights, biases, normalization parameters, and output limits are quantized and stored as constant hardware coefficients. Therefore, the deployed circuit performs only deterministic feed-forward inference and does not require backpropagation, gradient calculation, or online weight updates.

The RTL architecture implements the trained 1–8–8–1 network and contains three fully connected layers. The input to the network is the classical MMPD timing estimate, denoted by $e_{\mathrm{MM}}[n]$. The neural network produces a residual correction $\Delta e_{\mathrm{NN}}[n]$, which is added to the classical detector output according to

$$e_{\mathrm{AI}}[n] = e_{\mathrm{MM}}[n] + \Delta e_{\mathrm{NN}}[n]. \tag{15}$$

The complete network contains 97 parameters: eight weights and eight biases in the first layer, 64 weights and eight biases in the second layer, and eight weights and one bias in the output layer. The arithmetic performed by each layer and its corresponding RTL realization are described below.

3.1.1. Input Normalization

Before entering the neural network, the classical MMPD output is normalized using the mean μ_x and standard deviation σ_x obtained from the training dataset:

$$u[n] = \frac{e_{\mathrm{MM}}[n] - \mu_x}{\sigma_x}. \tag{16}$$

In the RTL implementation, division by σ_x is replaced by multiplication with a pre-computed reciprocal coefficient,

$$\gamma_x = \frac{1}{\sigma_x}, \tag{17}$$

such that

$$u[n] = (e_{\mathrm{MM}}[n] - \mu_x)\gamma_x. \tag{18}$$

Thus, the normalization hardware consists of one subtractor, one constant multiplier, and a fixed-point scaling stage. Both μ_x and γ_x are quantized offline and stored as signed constants in the RTL.

For a fixed-point representation with F fractional bits, a real-valued quantity a is represented by the integer

$$a_q = \mathrm{round}\left(a2^F\right). \tag{19}$$

The normalized RTL input is therefore computed as

$$u_q[n] = \mathrm{sat}\left[\frac{\left(e_{\mathrm{MM},q}[n] - \mu_{x,q}\right)\gamma_{x,q}}{2^F}\right], \tag{20}$$

where $\mathrm{sat}(\cdot)$ denotes signed saturation to the selected datapath width.

3.1.2. First Fully Connected Layer

The first fully connected layer receives one normalized input and produces eight neuron outputs. For neuron i, where $i \in \{0, \ldots, 7\}$, the arithmetic is

$$z_i^{(1)}[n] = W_i^{(1)}u[n] + b_i^{(1)}. \tag{21}$$

Because this layer contains only one input feature, each neuron requires one signed multiplier and one bias addition. In fixed-point RTL, the corresponding operation is

$$z_{i,q}^{(1)}[n] = \text{sat}\left[\frac{W_{i,q}^{(1)} u_q[n] + \left(b_{i,q}^{(1)} \ll F\right)}{2^F}\right]. \tag{22}$$

The multiplication of two signed fixed-point values produces a result with $2F$ fractional bits. The bias is therefore shifted left by F bits before it is added to the product. After accumulation, an arithmetic right shift by F bits returns the result to the original fixed-point format.

The corresponding RTL operation for each first-layer neuron can be expressed as

```
product_fc1 = input_norm * weight_fc1[i];
acc_fc1     = product_fc1 + (bias_fc1[i] <<< F);
fc1_out[i]  = saturate(acc_fc1 >>> F);
```

To reduce area, the eight multiplications may be time-multiplexed using a shared multiply–accumulate unit. Alternatively, all eight neurons may be evaluated in parallel when minimum inference latency is required.

3.1.3. First ReLU Activation

The output of each first-layer neuron is processed using the rectified linear unit activation function,

$$h_i^{(1)}[n] = \max\left(0, z_i^{(1)}[n]\right). \tag{23}$$

ReLU does not require multiplication or lookup-table logic. In two's-complement RTL, the sign bit of the neuron output directly determines the result:

$$h_{i,q}^{(1)}[n] = \begin{cases} 0, & z_{i,q}^{(1)}[n] < 0, \\ z_{i,q}^{(1)}[n], & z_{i,q}^{(1)}[n] \geq 0. \end{cases} \tag{24}$$

The corresponding Verilog operation is

```
relu1_out[i] = fc1_out[i][DATA_W-1] ?
               {DATA_W{1'b0}} : fc1_out[i];
```

Thus, each ReLU stage is implemented using only sign-bit detection and a multiplexer.

3.1.4. Second Fully Connected Layer

The second fully connected layer contains eight input neurons and eight output neurons. For output neuron j, the floating-point computation is

$$z_j^{(2)}[n] = b_j^{(2)} + \sum_{i=0}^{7} W_{j,i}^{(2)} h_i^{(1)}[n], \qquad j \in \{0, \ldots, 7\}. \tag{25}$$

Each second-layer neuron therefore requires eight signed multiply–accumulate operations. The fixed-point RTL expression is

$$z_{j,q}^{(2)}[n] = \text{sat}\left[\frac{\left(b_{j,q}^{(2)} \ll F\right) + \sum_{i=0}^{7} W_{j,i,q}^{(2)} h_{i,q}^{(1)}[n]}{2^F}\right]. \tag{26}$$

The accumulator width is selected to prevent overflow during the summation of eight signed products. For an operand width of B bits, each product requires approximately $2B$ bits, while the addition of eight products requires at least three additional guard bits. A wider accumulator is used in the RTL before rounding, shifting, and saturation return the result to the selected neuron width.

The arithmetic for one second-layer neuron is mapped to RTL as

```
acc_fc2 = bias_fc2[j] <<< F;

for (i = 0; i < 8; i = i + 1)
    acc_fc2 = acc_fc2 +
              relu1_out[i] * weight_fc2[j][i];

fc2_out[j] = saturate(acc_fc2 >>> F);
```

For a fully parallel architecture, the second layer requires 64 multipliers. A resource-shared implementation can instead reuse eight or fewer multipliers over several clock cycles, trading inference latency for lower silicon area.

3.1.5. Second ReLU Activation

The eight second-layer outputs are passed through another ReLU stage:

$$h_j^{(2)}[n] = \max\left(0, z_j^{(2)}[n]\right). \tag{27}$$

The fixed-point RTL implementation is identical to the first ReLU stage:

```
relu2_out[j] = fc2_out[j][DATA_W-1] ?
               {DATA_W{1'b0}} : fc2_out[j];
```

No additional numerical scaling is required because the ReLU operation preserves the fixed-point position of positive values.

3.1.6. Output Fully Connected Layer

The output layer combines the eight second-layer activations to generate one residual timing correction. Its floating-point operation is

$$\Delta e_{\text{raw}}[n] = b^{(3)} + \sum_{j=0}^{7} W_j^{(3)} h_j^{(2)}[n]. \tag{28}$$

Unlike the hidden layers, the output layer uses a linear activation because the residual correction can be either positive or negative. Its fixed-point implementation is

$$\Delta e_{\text{raw},q}[n] = \text{sat}\left[\frac{\left(b_q^{(3)} \ll F\right) + \sum_{j=0}^{7} W_{j,q}^{(3)} h_{j,q}^{(2)}[n]}{2^F}\right]. \tag{29}$$

The corresponding RTL arithmetic is

```
acc_fc3 = bias_fc3 <<< F;

for (j = 0; j < 8; j = j + 1)
    acc_fc3 = acc_fc3 +
              relu2_out[j] * weight_fc3[j];
```

```
residual_raw = saturate(acc_fc3 >>> F);
```

This layer requires eight multiply–accumulate operations and produces the signed residual correction in the same fixed-point format as the classical MMPD output.

3.1.7. Residual Bounding

To prevent an excessively large neural-network correction from disturbing the CDR feedback loop, the output is bounded to the interval $[-\Delta_{\max}, +\Delta_{\max}]$:

$$\Delta e_{\mathrm{NN}}[n] = \begin{cases} +\Delta_{\max}, & \Delta e_{\mathrm{raw}}[n] > \Delta_{\max}, \\ -\Delta_{\max}, & \Delta e_{\mathrm{raw}}[n] < -\Delta_{\max}, \\ \Delta e_{\mathrm{raw}}[n], & \text{otherwise.} \end{cases} \tag{30}$$

In RTL, this operation requires two signed comparators and a multiplexer:

```
if (residual_raw >  DELTA_MAX_Q)
    residual_bounded = DELTA_MAX_Q;
else if (residual_raw < -DELTA_MAX_Q)
    residual_bounded = -DELTA_MAX_Q;
else
    residual_bounded = residual_raw;
```

The clipping operation preserves the residual-learning behavior while imposing a deterministic upper bound on the AI contribution to the phase detector.

3.1.8. Final AI-Assisted MMPD Output

The bounded residual correction is finally added to the classical MMPD output:

$$e_{\mathrm{AI},q}[n] = \mathrm{sat}\Big(e_{\mathrm{MM},q}[n] + \Delta e_{\mathrm{NN},q}[n]\Big). \tag{31}$$

The corresponding RTL implementation consists of one signed adder and output saturation logic:

```
ai_sum   = mmpd_classical + residual_bounded;
mmpd_ai  = saturate(ai_sum);
```

The generated AI-assisted timing error is then supplied to the digital proportional–integral loop filter in the same manner as the output of the conventional MMPD. Therefore, the neural correction engine can be inserted into an existing baud-rate CDR without modifying the remaining loop architecture.

Overall, one inference requires 80 multiplications: eight in the first layer, 64 in the second layer, and eight in the output layer. It also requires 80 weight additions, 17 bias additions, 16 ReLU operations, residual clipping, and one final residual addition. Because all coefficients are constant following offline training, synthesis tools may further optimize constant multiplications using shift–add structures, canonical signed-digit representations, or shared multiply–accumulate resources. The resulting datapath provides a direct hardware realization of the trained residual model while retaining compatibility with conventional synthesizable RTL-to-GDS implementation flows.

3.2. Programmable Weight and Bias Storage

The RTL implementation presented in the previous subsection assumes that the neural-network coefficients are available before inference begins. Rather than hardwiring the weights and biases as Verilog constants, the proposed architecture stores all neural-network parameters in programmable storage elements. Specifically, each weight and bias is mapped to a fixed-point register or embedded memory location whose width matches the selected datapath precision. During normal inference, these registers provide the coefficient inputs to the multiply–accumulate (MAC) units in each fully connected layer.

At system initialization, the programmable registers are loaded through a configuration interface from an external controller or non-volatile memory. Once loading is complete, the registers retain their programmed values and are read repeatedly during every inference cycle without modifying the arithmetic datapath. Consequently, the MAC units, ReLU blocks, accumulators, and output logic remain identical regardless of the programmed neural-network model.

For larger neural networks, the coefficient registers may alternatively be implemented using embedded SRAM or ROM with address generation logic. Each layer sequentially reads the required weights and biases from memory during inference, while the same arithmetic hardware performs the multiply–accumulate operations. This approach reduces register count and scales efficiently with increasing network size.

The programmable storage architecture allows the neural-network parameters to be updated without modifying the RTL implementation. During post-layout validation, refined weights and biases generated from timing-accurate simulations can simply overwrite the previously programmed coefficient values through the configuration interface. Since only the stored parameters change, the inference datapath, control logic, and arithmetic units remain completely unchanged, allowing the same synthesized hardware implementation to support successive generations of trained neural-network models.

3.3. Cross-Node ASIC Implementation

To demonstrate the hardware realizability and technology portability of the proposed AI-assisted clock and data recovery (CDR) architecture, a digital implementation was synthesized and physically implemented across four semiconductor technology nodes using an automated RTL-to-GDS design flow. A key advantage of the proposed CDR architecture is the strict use of digital blocks for analog switching behavior, enabling increased portability across process nodes [19]. The implemented digital core includes the classical Mueller–Müller phase detector, the proportional–integral (PI) loop filter, the numerically controlled oscillator (NCO), and the associated digital control logic required for closed-loop timing recovery. The neural-network inference engine is implemented using fixed-point synthesizable RTL and performs only feed-forward inference during operation. The network parameters are obtained through offline training and stored as fixed-point coefficients, eliminating the need for online training or backpropagation. This digital implementation enables the AI-assisted detector to be integrated with the conventional CDR architecture using a technology-independent RTL description.

The implementation flow combines AI-assisted circuit design with conventional ASIC implementation methodologies. The trained neural network is translated into a lightweight fixed-point hardware inference engine and integrated with the digital CDR architecture using synthesizable Verilog RTL. Once generated, the same RTL description is preserved for all technology nodes, while only the technology-dependent process design kit (PDK), standard-cell libraries, timing libraries, and physical implementation constraints are replaced during synthesis. A similar RTL-based portability approach has been demonstrated for a synthesizable ADPLL implemented across multiple open-source

process technologies [20]. This approach enables identical functionality to be maintained across multiple fabrication technologies without modifying either the neural-network architecture or the surrounding digital signal-processing logic. This methodology is consistent with digital-twinning approaches that preserve the RTL description, configuration parameters, and verification environment when regenerating mixed-signal CDR systems across technology nodes [21].

Figure 3 presents the synthesized GDS layouts obtained for GF180, Sky130HD, Nangate45, and ASAP7 technologies. The layouts were generated from the identical RTL description using the same automated physical design flow, demonstrating that the proposed AI-assisted CDR architecture can be successfully implemented across multiple CMOS technology generations without redesigning the digital hardware. As expected, technology scaling results in progressively smaller silicon area, increased routing density, and higher operating frequency while preserving the overall structural organization of the design.

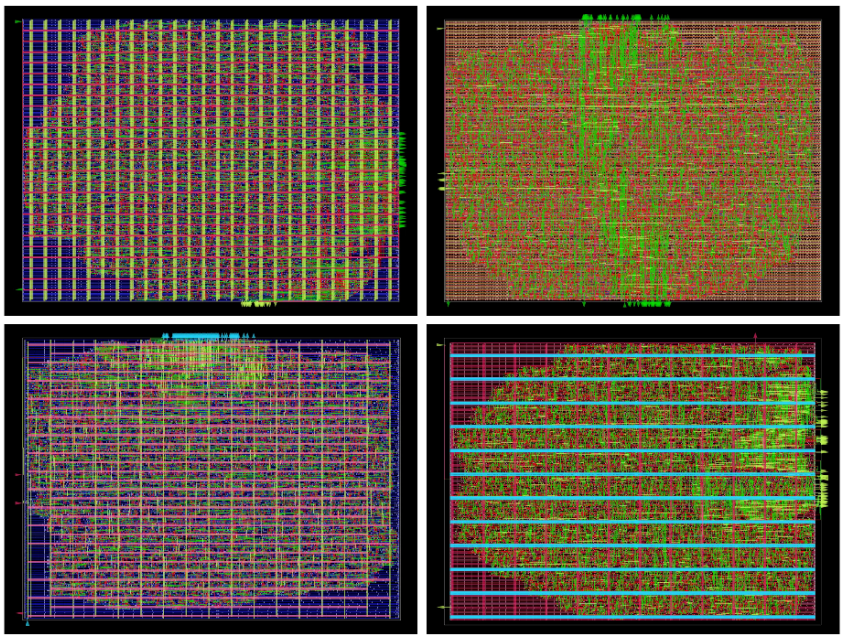


Figure 3. Synthesized GDS layouts of the proposed AI-assisted baud-rate PAM4 CDR digital core across four CMOS technology nodes. Top row: GF180 (180 nm) and Sky130HD (130 nm). Bottom row: Nangate45 (45 nm) and ASAP7 (7 nm FinFET). The same synthesizable RTL and automated RTL-to-GDS implementation flow were used for all technologies, demonstrating successful cross-node portability of the AI-assisted architecture.

To quantitatively evaluate the physical implementation and cross-node scalability, timing, power, area, and routing metrics were extracted from the post-implementation results for each technology. Table 1 summarizes the corresponding implementation characteristics for ASAP7, Nangate45, Sky130HD, and GF180. These results provide a quantitative comparison of the achievable operating speed, estimated power consumption, physical area, and routing resources while preserving the same RTL functionality across the four PDK technologies.

As shown in Table 1, technology scaling produces a substantial improvement in achievable operating frequency and physical area. The ASAP7 implementation achieves the highest maximum operating frequency, approximately 4.10–4.46 GHz, followed by Nangate45 at 2.25–2.36 GHz. The mature Sky130HD and GF180 implementations achieve approximately 455–475 MHz and 380–400 MHz, respectively. A corresponding reduction in physical area is observed with technology scaling. The implemented core occupies approximately

1.376 mm^2 in GF180, 0.253 mm^2 in Sky130HD, 0.0547 mm^2 in Nangate45, and 0.00424 mm^2 in ASAP7.

Table 1. Cross-node ASIC implementation results for the CDR digital core.

Metric	ASAP7	N45	S130	G180
Technology node	7 nm	45 nm	130 nm	180 nm
F_{max} (MHz)	4100–4460	2250–2360	455–475	380–400
T_{min} (ps)	240–250	440–460	2100–2200	2520–2610
Power (W)	0.94–1.33	0.0106–0.0116	0.044–0.052	2.27–2.68
Clock power (%)	14–19	6–7	13–14	5
Area (μm^2)	4239	54,718	253,179	1,375,873
Normalized area	1.0	12.9	59.7	324.7
Routing layers	5	6	5	4

The estimated power consumption exhibits a stronger dependence on the technology library, operating conditions, clock constraints, and implementation parameters. Consequently, the reported power values are intended to characterize the individual physical implementations rather than provide a technology-normalized power comparison. Nevertheless, the results demonstrate successful physical implementation over a wide range of CMOS technology generations while maintaining the same underlying RTL functionality. The reported area corresponds to the complete physically implemented digital CDR core. A hierarchical area breakdown of the individual MMPD, neural-network inference engine, PI loop filter, NCO, and control-logic blocks was not extracted in the present implementation flow; therefore, their individual area contributions are not reported separately.

The cross-node implementations further demonstrate that lightweight neural network can integrate naturally within a standard ASIC design flow. Because the AI engine performs inference only and all weights can be fixed following offline training, no retraining or architectural modifications are required when the design is migrated to a different fabrication technology. Consequently, the proposed implementation provides a practical methodology for deploying AI-assisted mixed-signal communication systems across successive CMOS technology nodes while minimizing redesign effort and maintaining compatibility with existing digital implementation flows.

4. Results

4.1. Phase Detector Characterization

The proposed AI-assisted residual Mueller–Müller Phase Detector (MMPD) was first evaluated independently of the clock and data recovery (CDR) loop in order to quantify its intrinsic phase detection performance before closed-loop operation. The objective of this characterization is to determine how effectively the proposed residual neural network linearizes the classical detector while preserving its low-complexity architecture and noise characteristics. All three detectors—the conventional Mueller–Müller detector, the fixed-β MMPD, and the proposed AI-assisted residual MMPD—were evaluated using identical PAM4 waveforms generated with root-raised cosine pulse shaping (32 samples/symbol, roll-off factor 0.35), a four-tap ISI channel, additive white Gaussian noise corresponding to an SNR of 22 dB, and timing offsets spanning $\pm$0.45 UI. The received samples were interpolated for each timing offset, quantized using a PAM4 slicer, and processed by the respective phase detectors. For every timing offset, the detector output was averaged over 12,000 transmitted symbols to obtain a statistically stable phase-detector characteristic. The relatively long sequence length was selected to reduce data-pattern-dependent and noise-induced fluctuations in the estimated mean detector response while maintaining a

practical behavioral simulation time. Since the phase-detector characteristic is obtained from the sample-averaged detector output at each timing offset, increasing the number of transmitted symbols reduces the statistical uncertainty of the estimated mean, with progressively diminishing improvement as the sequence length increases. The 12,000-symbol sequence was therefore selected as a practical compromise between statistical averaging accuracy and behavioral simulation time, and this averaging length does not affect the architecture or hardware complexity of the proposed detector. The AI residual network was trained only once using the deterministic residual error between the ideal linear detector response and the measured classical MMPD characteristic, allowing the learned correction to be directly added to the conventional detector output without modifying the underlying MMPD structure.

Figure 4 presents the average phase detector transfer characteristic obtained from the above procedure. The horizontal axis represents the imposed sampling phase offset, while the vertical axis denotes the mean detector output after averaging across the complete PAM4 sequence. The dashed reference line corresponds to the ideal linear phase detector characteristic. As expected, the classical Mueller–Müller detector exhibits gain compression as the timing offset increases, resulting in a progressively flatter response near the edges of the operating range. Although the fixed-β detector slightly modifies the average detector gain through the addition of the polarity correction term, it still exhibits noticeable nonlinear saturation at larger timing errors. In contrast, the proposed AI-assisted detector closely overlaps the ideal linear response throughout almost the complete ± 0.45 UI operating region. Because the neural network predicts only the nonlinear residual component rather than replacing the detector itself, the detector retains the desirable behavior of the classical MMPD while significantly improving its linearity and extending its usable operating range.

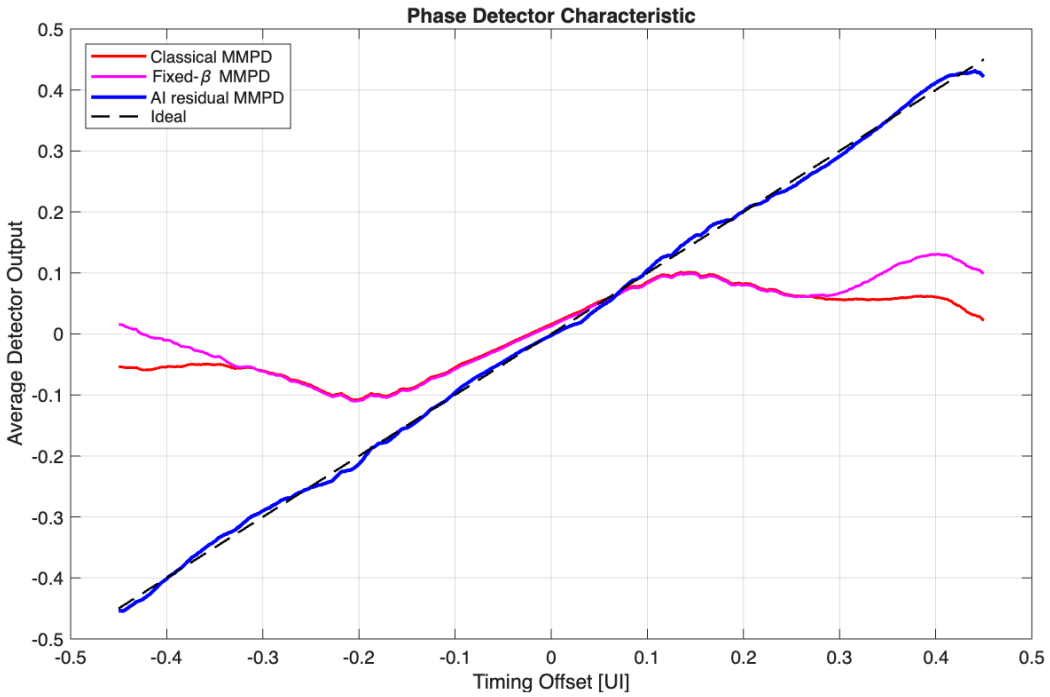


Figure 4. Average phase detector characteristic for the classical MMPD, fixed-β MMPD, and the proposed AI-assisted residual MMPD. The AI-assisted detector closely follows the ideal linear response over a significantly wider timing range.

The residual correction learned by the neural network is illustrated in Figure 5. During training, the target residual was defined as the difference between the ideal linear detector response and the measured output of the classical MMPD. Consequently, the network

learns only the deterministic nonlinear error introduced by the conventional detector. The network accepts a single normalized phase-error input and processes it using two fully connected hidden layers containing eight neurons each with ReLU activation functions before generating a single residual correction value. The learned mapping remains close to zero around the lock point, where the classical detector is already approximately linear, while progressively larger corrections are generated toward larger positive and negative timing offsets where detector saturation becomes significant. The smooth and symmetric shape of the learned residual demonstrates that the network successfully captures the deterministic nonlinear distortion of the classical MMPD using only one input feature. Furthermore, the entire network requires only 97 trainable parameters, making it well suited for low-area RTL implementation and fixed-point hardware realization.

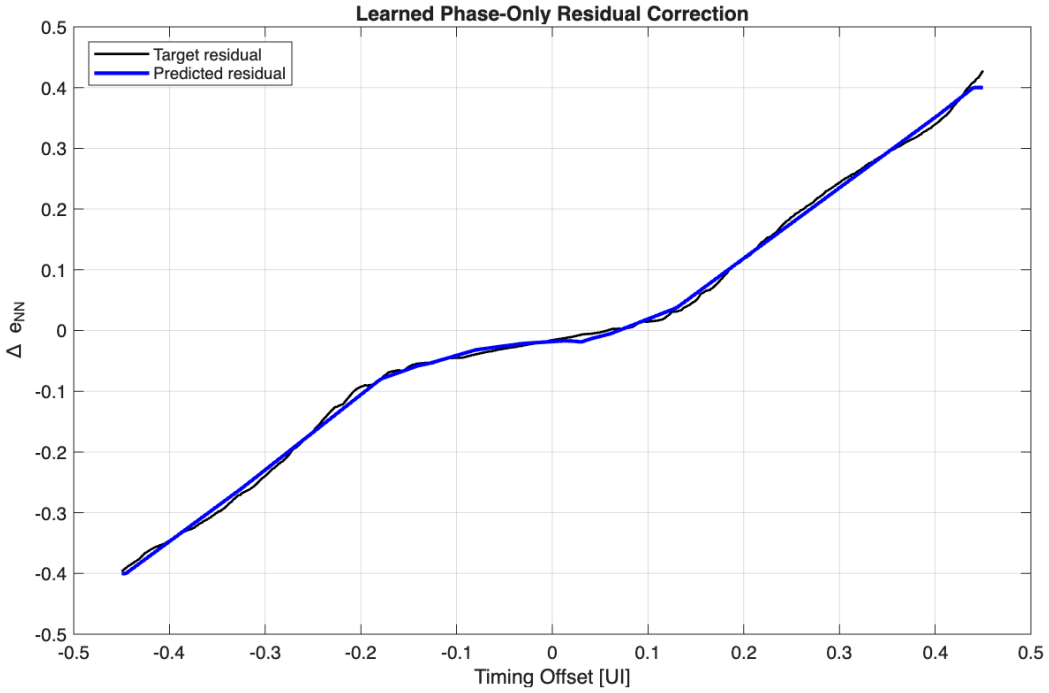


Figure 5. Learned phase-only residual correction generated by the proposed neural network. The network estimates only the nonlinear residual error that is added to the classical MMPD output.

To quantify detector linearity, the static phase detector error is evaluated in Figure 6. For every imposed timing offset, the linearity error is calculated as the difference between the ideal detector response and the measured detector output. The dashed horizontal lines indicate the ±0.05 UI linearity tolerance used to determine the effective linear operating range of each detector. The classical and fixed-β detectors remain reasonably linear only around the lock point before their error rapidly increases with timing offset due to gain compression. The proposed AI-assisted detector maintains an almost constant error close to zero across nearly the entire operating region, with the maximum error remaining within approximately ±0.03 UI. This substantial reduction in static error confirms that the residual neural network effectively compensates for the deterministic nonlinear behavior of the classical detector while preserving its original operating principle.

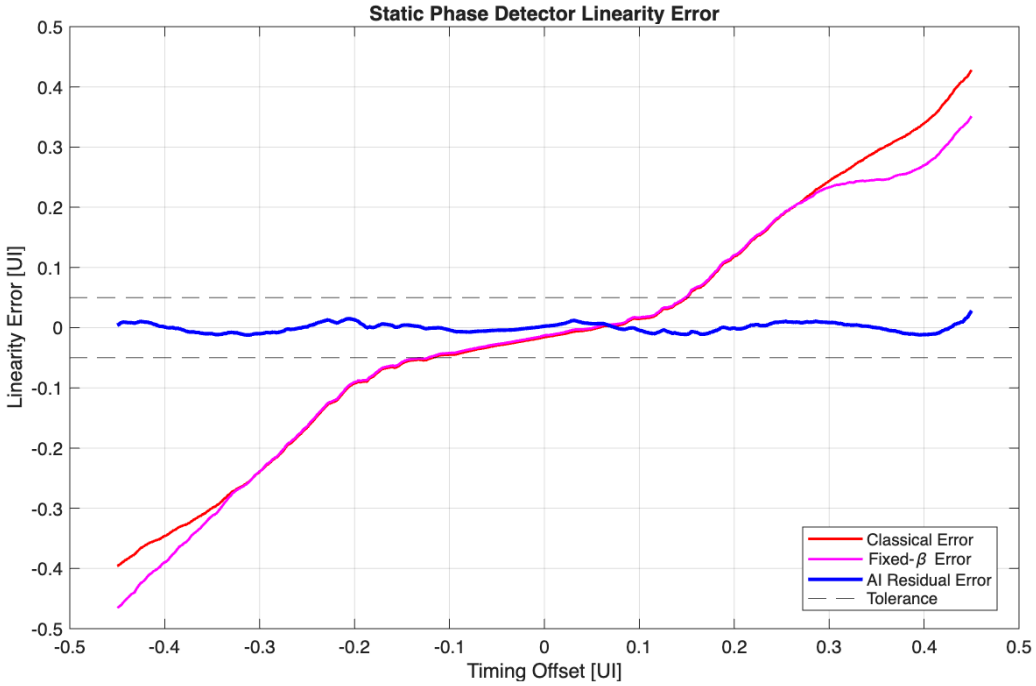


Figure 6. Static phase detector linearity error versus timing offset. The proposed AI-assisted detector maintains near-zero linearity error across almost the entire operating region.

While improving detector linearity is important, it is equally critical that the correction does not amplify detector noise. Figure 7 therefore evaluates the detector output distribution when the sampling phase is fixed at zero timing offset. The received PAM4 waveform was processed under identical channel and AWGN conditions, and the detector output mean was removed so that only the stochastic detector noise remained. The resulting histograms demonstrate that the AI-assisted detector produces nearly the same output noise distribution as the classical MMPD, indicating that the residual correction primarily compensates deterministic nonlinear distortion rather than amplifying random noise. The measured detector output noise standard deviation remains approximately 0.363 UI for both the classical and AI-assisted detectors, whereas the fixed-β detector exhibits noticeably broader output fluctuations due to the additional polarity-dependent correction term. These results demonstrate that the proposed approach achieves substantially improved detector linearity without sacrificing intrinsic noise performance.

The dynamic behavior of the detector was next evaluated by applying sinusoidal timing jitter to the received sampling instants. For each experiment, the received waveform was sampled according to a known sinusoidal timing error before the detector output was smoothed using a 64-sample moving-average filter to suppress random detector noise. The processed detector output was then compared with the known injected timing error using the Pearson correlation coefficient, which measures how accurately the detector tracks the applied timing variation. Figure 8 illustrates the detector correlation as the sinusoidal jitter amplitude increases from 0.01 UI to 0.20 UI while maintaining a fixed jitter frequency. Small jitter amplitudes are accurately tracked by all three detectors; however, as the timing excursion increases and the detector increasingly operates within its nonlinear region, the performance of the classical and fixed-β detectors gradually deteriorates. Owing to its substantially improved linear characteristic, the proposed AI-assisted detector consistently maintains the highest correlation across the entire amplitude range, demonstrating superior tracking capability during large timing excursions.

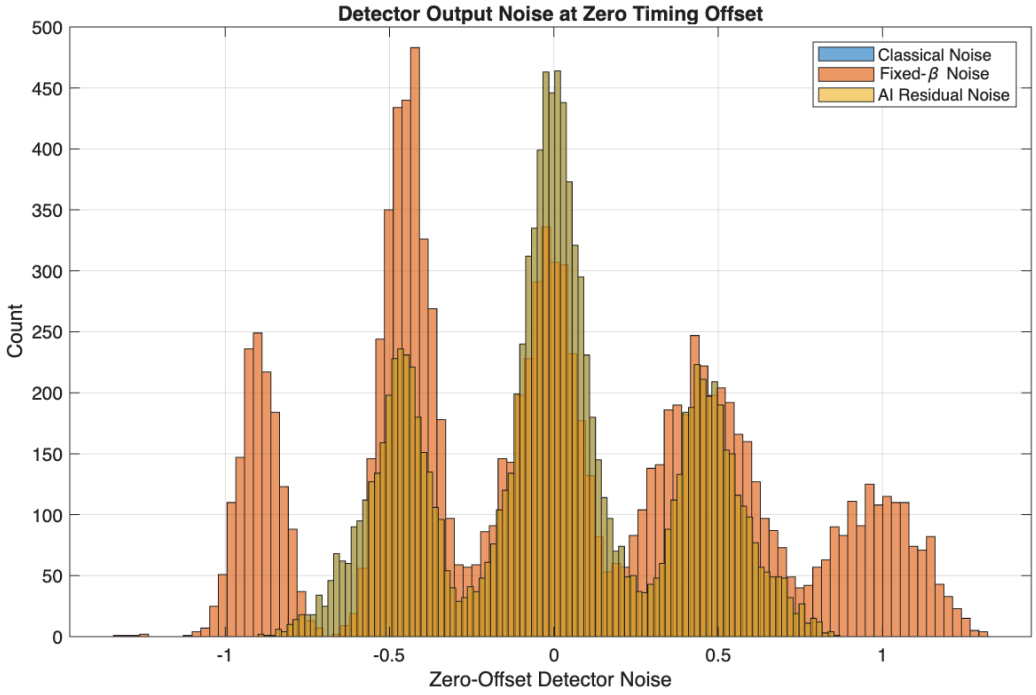


Figure 7. Zero-offset detector output noise distribution for the three phase detectors. The proposed detector preserves the noise characteristics of the classical MMPD while significantly improving linearity.

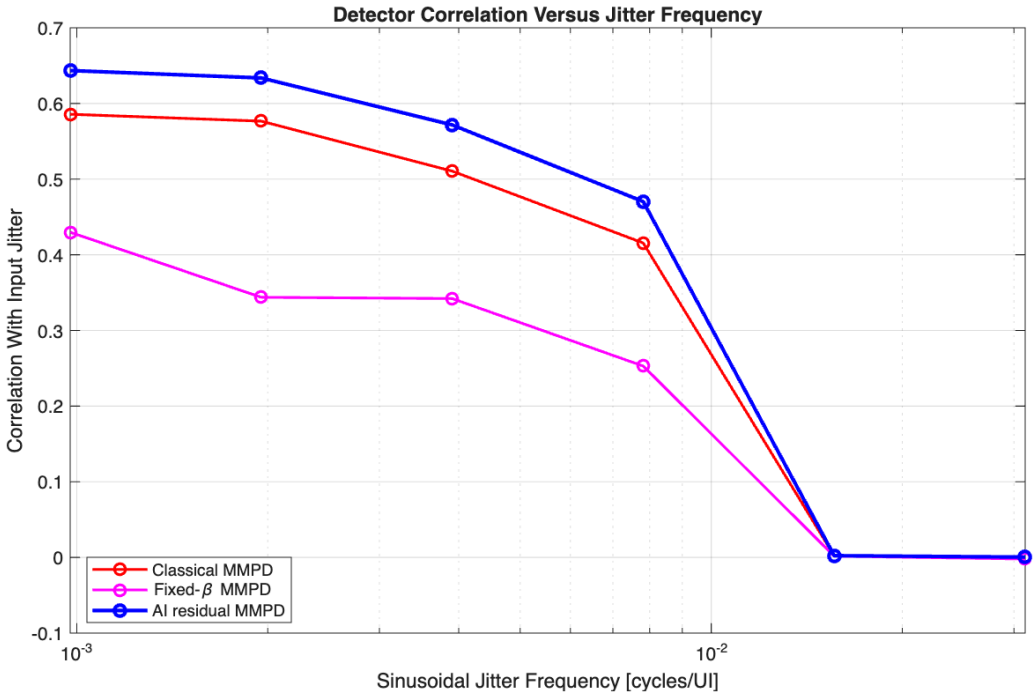


Figure 8. Detector correlation with input sinusoidal jitter as the jitter amplitude increases. Higher correlation indicates improved timing estimation accuracy.

A complementary experiment was performed by fixing the sinusoidal jitter amplitude at 0.05 UI while sweeping the jitter frequency from 1/1024 to 1/32 cycles/UI. Figure 9 presents the resulting correlation between the detector output and the applied timing error. At low modulation frequencies all detectors accurately reproduce the input timing variation. As the jitter frequency increases, however, the limited dynamic response and nonlinear characteristics of the classical detectors reduce their tracking accuracy. Throughout the useful operating bandwidth, the AI-assisted detector consistently achieves the highest

correlation, indicating more faithful tracking of rapidly varying timing errors. Eventually, at sufficiently high modulation frequencies, the correlation of all three detectors decreases as the detector bandwidth becomes the limiting factor rather than the detector nonlinearity itself. Nevertheless, the proposed detector maintains the largest useful tracking bandwidth among the evaluated architectures.

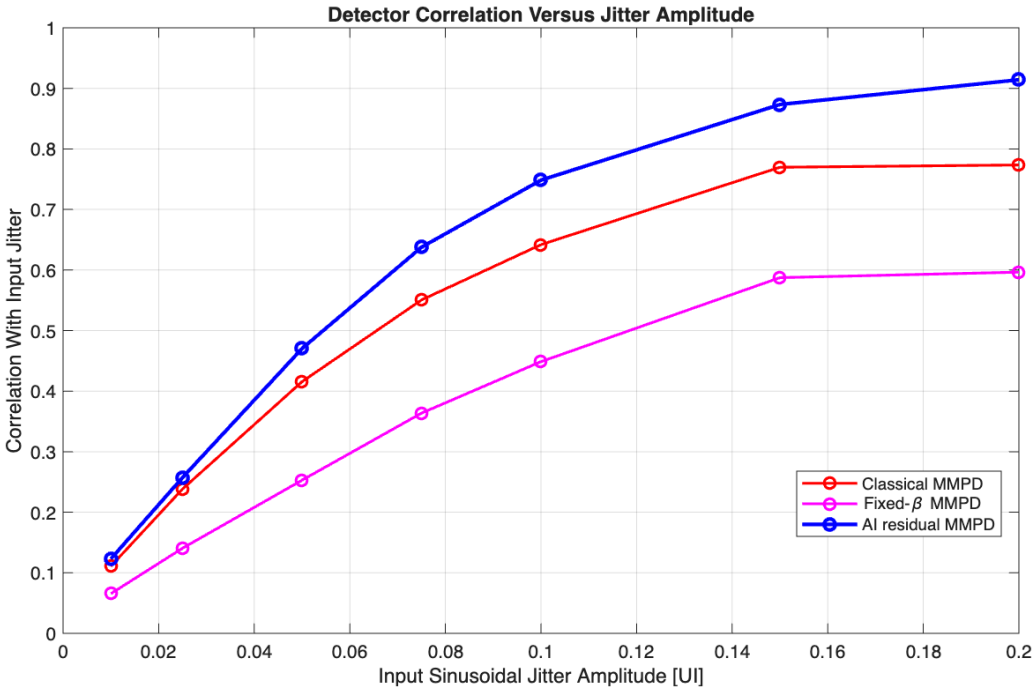


Figure 9. Detector correlation with input sinusoidal jitter as the jitter frequency increases. The proposed detector exhibits superior tracking throughout the useful operating bandwidth.

The quantitative detector performance is summarized in Table 2. Compared with the classical MMPD, the proposed AI-assisted detector reduces the phase detector RMSE from 0.2088 UI to only 0.0073 UI while reducing the peak linearity error from 0.4282 UI to 0.0282 UI. The average detector gain improves from approximately 0.70 to an almost ideal value of 1.0027, and the usable linear operating range expands from approximately ±0.12 UI to ±0.45 UI. The number of monotonicity violations is simultaneously reduced from 176 to only 12, confirming a significantly smoother transfer characteristic. Dynamic evaluation further demonstrates that the proposed detector achieves the highest tracking correlation (0.4703) together with the best tracking SNR (−1.24 dB) while maintaining essentially the same detector output noise as the conventional implementation. These results collectively demonstrate that learning only the deterministic residual error enables substantial improvements in both static detector linearity and dynamic timing estimation while requiring only a compact neural network containing 97 trainable parameters that is suitable for efficient digital hardware implementation.

Table 2. Performance comparison of Phase Detector Architectures.

Metric	Classical MMPD	Fixed-β MMPD	AI Residual MMPD
RMSE (UI)	0.2088	0.2042	0.0073
Mean Absolute Error (UI)	0.1640	0.1593	0.0062
Peak Error (UI)	0.4282	0.4653	0.0282
Average Detector Gain (K_{PD})	0.7025	0.7055	1.0027
Linear Operating Range (±UI)	0.117	0.122	0.450
Monotonicity Violations	176	170	12

Table 2. *Cont.*

Metric	Classical MMPD	Fixed-β MMPD	AI Residual MMPD
Noise Standard Deviation (UI)	0.363	0.579	0.363
Tracking Correlation	0.4156	0.2529	0.4703
Tracking SNR (dB)	−2.06	−5.83	−1.24

4.2. AI-Assisted MMPD Integrated in the Baud-Rate CDR

To evaluate the practical impact of the proposed AI-assisted residual Mueller–Müller phase detector (MMPD), the detector was integrated into a complete baud-rate PAM4 clock and data recovery (CDR) loop and compared with the conventional MMPD and the fixed-β MMPD under identical operating conditions. The CDR consists of a root-raised cosine (RRC) pulse-shaped PAM4 transmitter, a baud-rate sampler, a four-level PAM4 slicer, the selected MMPD, a digital proportional–integral (PI) loop filter, and a numerically controlled oscillator (NCO) that updates the sampling phase once per symbol. The transmitted waveform passes through an ISI channel with additive white Gaussian noise (AWGN), while both sinusoidal jitter (SJ) and random jitter (RJ) are injected before sampling to emulate realistic serial-link impairments. Fractional-delay interpolation is employed to realize sub-UI sampling resolution. All detector configurations use identical transmitted data, channel conditions, loop parameters, and acquisition settings to ensure a fair comparison.

The simulations were performed at a symbol rate of 1 Gbaud with 64 samples per symbol. The received signal experiences an SNR of 28 dB together with 0.040 UI sinusoidal jitter and 0.003 UI RMS random jitter. The value $\beta = 0.45$ was selected as a fixed correction coefficient close to the nominal $\beta = 0.5$ adopted in [13] for comparative evaluation. As demonstrated in [13], the correction factor controls the trade-off between phase-acquisition performance and steady-state jitter, and its optimum value depends on the specific channel and loop requirements. Accordingly, $\beta = 0.45$ was not treated as a globally optimal value in this work, but was kept constant throughout all closed-loop comparisons to provide a consistent fixed-coefficient baseline for evaluating the proposed AI-assisted residual detector.

Table 3 summarizes the overall closed-loop performance of the three detector architectures. The proposed AI-assisted detector achieves the lowest RMS timing error of 0.022321 UI, corresponding to improvements of 13.45% and 16.29% compared with the classical and fixed-β MMPDs, respectively. It also reduces the peak timing error to 0.061563 UI, demonstrating improved steady-state timing stability. Although the fixed-β detector achieves the lowest symbol error rate under these simulation conditions, the proposed detector is optimized to improve timing recovery rather than symbol decisions, resulting in significantly better phase tracking while maintaining reliable CDR operation. Future detectors can optimize over both timing recovery and symbol detection equalization. All three detectors successfully lock over the complete tested phase acquisition range of ± 0.45 UI and frequency acquisition range of ± 400 ppm with a 100% lock success rate.

Figure 10 compares the closed-loop phase error trajectories of the classical and AI-assisted detectors. Both loops converge from the same initial timing offset and successfully acquire lock. However, after the transient period, the AI-assisted detector consistently produces smaller phase excursions and lower steady-state oscillations. This reduction demonstrates that the residual neural correction effectively compensates for the nonlinear response of the classical MMPD, allowing the recovered sampling clock to follow the incoming data timing more accurately.

Figure 11 presents the moving RMS timing error computed over a sliding observation window after lock acquisition. The AI-assisted detector consistently exhibits the lowest moving RMS timing error across the entire simulation interval, whereas the fixed-β detector

shows the largest fluctuations. The reduced temporal variation confirms that the proposed residual correction suppresses nonlinear detector errors, producing a more stable recovered clock without increasing loop complexity.

Table 3. Performance comparison of the baud-rate PAM4 CDR using classical, fixed-β, and AI-assisted residual MMPD.

Metric	Classical	Fixed-β	AI-Assisted
RMS Timing Error (UI)	0.025789	0.026666	0.022321
Mean Timing Error (UI)	−0.013012	−0.012535	−0.013360
Peak Timing Error (UI)	0.069631	0.078005	0.061563
Symbol Error Rate	3.338×10^{-4}	2.225×10^{-4}	4.451×10^{-4}
Mean Acquisition Lock Time (UI)	41.2	22.2	24.4
Mean Settling Time (UI)	124.6	32.3	88.2
Mean Post-Lock RMS Error (UI)	0.014699	0.002598	0.011217
Worst Post-Lock Peak Error (UI)	0.033247	0.015091	0.027867
Integrated Control Effort	3.0371×10^{-3}	1.9819×10^{-2}	3.1760×10^{-3}

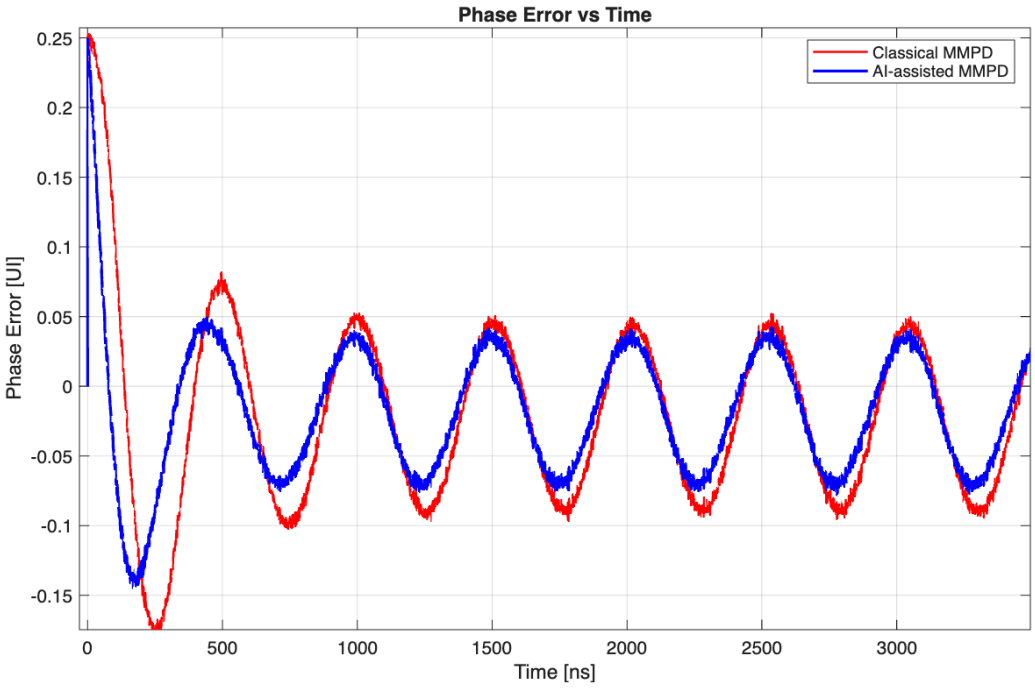


Figure 10. Phase Error vs. Time for the classical and AI-assisted baud-rate PAM4 CDR. The proposed detector exhibits reduced steady-state phase excursions after acquisition while maintaining stable lock.

Figure 12 shows the closed-loop jitter tolerance of the three detector architectures. At each sinusoidal-jitter frequency, the maximum tolerable jitter amplitude is determined while simultaneously satisfying successful loop lock, RMS timing-error limits, and symbol-error-rate constraints. As expected for a second-order CDR, the tolerable jitter amplitude decreases with increasing jitter frequency because the finite loop bandwidth limits high-frequency tracking. Nevertheless, the proposed AI-assisted detector consistently provides the highest jitter tolerance at low frequencies, tolerating approximately 0.21 UI compared with approximately 0.19 UI for the classical detector and 0.18 UI for the fixed-β detector. At higher jitter frequencies, the responses converge as the loop dynamics become the dominant limitation rather than the detector characteristic.

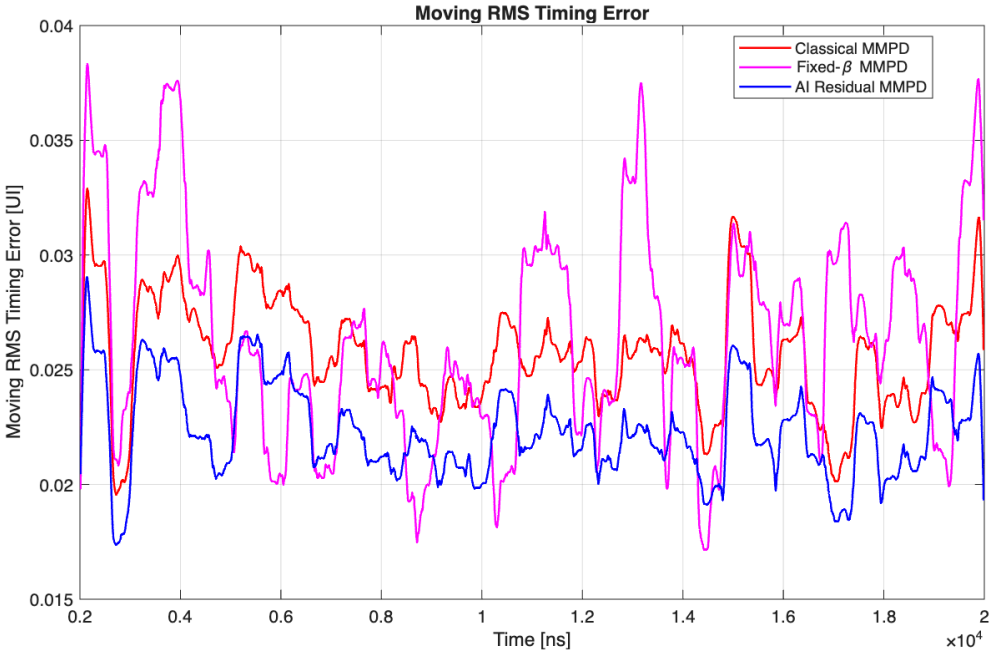


Figure 11. Moving RMS Timing Error of the three CDR implementations. The proposed AI-assisted detector maintains the lowest timing-error variation throughout steady-state operation.

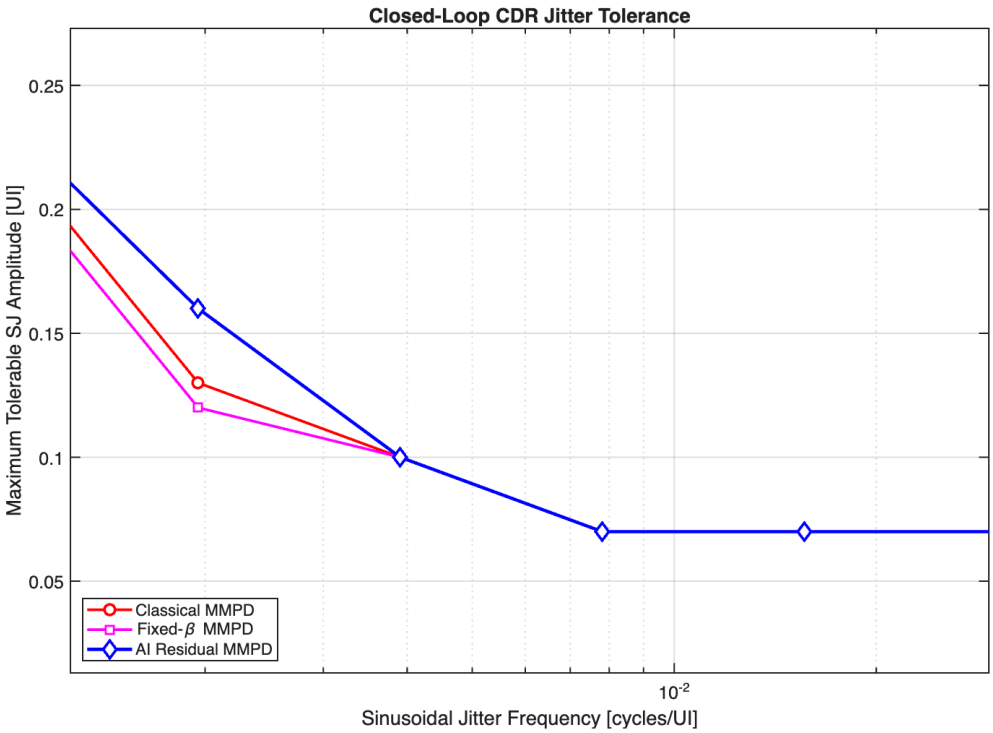


Figure 12. Closed-Loop CDR Jitter Tolerance. The AI-assisted detector tolerates larger low-frequency sinusoidal jitter amplitudes while maintaining successful lock, acceptable timing error, and symbol error rate.

The acquisition results reported in Table 3 further demonstrate that the proposed detector preserves robust closed-loop operation. All three architectures achieve a 100% phase-lock success rate over the complete tested acquisition range of 0.90 UI and successfully recover from frequency offsets spanning 800 ppm. The AI-assisted detector maintains acquisition performance comparable to the conventional detector while substantially improving steady-state timing accuracy. Although the fixed-β detector exhibits faster acquisition and lower post-lock error under these specific test conditions, its overall jitter tolerance and

nonlinear phase-tracking capability remain inferior to those of the proposed detector. Furthermore, the integrated control effort of the AI-assisted detector (3.1760×10^{-3}) remains close to that of the classical implementation, indicating that the improvement is achieved primarily through detector linearization rather than increased loop activity.

Overall, integrating the proposed residual neural correction into the baud-rate PAM4 CDR significantly improves timing recovery while preserving the simplicity of the conventional MMPD architecture. The proposed detector provides lower RMS and peak timing errors, improved long-term timing stability, enhanced low-frequency jitter tolerance, and reliable acquisition across the tested phase and frequency offset ranges, demonstrating the effectiveness of AI-assisted residual compensation for nonlinear MMPD characteristics.

5. Conclusions

This paper presented a hardware-efficient AI-assisted residual Mueller–Müller phase detector (MMPD) for baud-rate PAM4 clock and data recovery (CDR). Instead of replacing the conventional phase detector with a fully data-driven model, the proposed approach preserves the classical MMPD and employs a lightweight fully connected neural network to learn only the nonlinear residual timing error. By estimating only the residual correction, the proposed architecture significantly reduces learning complexity while maintaining compatibility with existing baud-rate CDR implementations and preserving the stability of the conventional timing recovery loop.

Behavioral simulations demonstrated that the proposed residual-learning framework substantially improves the intrinsic phase detector characteristics. The AI-assisted detector closely follows the ideal linear timing response over the complete tested phase range while reducing the detector RMSE from 0.2088 UI to 0.0073 UI, increasing the effective detector gain from 0.7025 to 1.0027, extending the linear operating range from approximately $\pm$0.12 UI to $\pm$0.45 UI, and reducing monotonicity violations from 176 to 12. The detector also maintains output noise comparable to the classical MMPD while providing superior timing estimation under sinusoidal jitter.

The proposed detector was further verified within a complete 1-Gbaud PAM4 baud-rate CDR consisting of a baud-rate sampler, PAM4 slicer, digital PI loop filter, and numerically controlled oscillator. Compared with the classical and fixed-β MMPDs, the proposed detector achieved the lowest closed-loop RMS timing error of 0.022321 UI, corresponding to improvements of 13.45% and 16.29%, respectively, while simultaneously reducing peak timing error and improving low-frequency jitter tolerance. All detector configurations successfully acquired lock over the tested phase and frequency acquisition ranges, demonstrating that the proposed residual correction improves steady-state timing accuracy without compromising acquisition robustness or loop stability.

The proposed neural network contains only 97 trainable parameters and performs inference only during operation, making the computational overhead minimal and well suited for fixed-point ASIC implementation. Since the residual-learning framework augments rather than replaces the conventional Mueller–Müller detector, it can be incorporated into existing baud-rate CDR architectures with only modest additional hardware. Consequently, the proposed AI-assisted residual MMPD provides a practical and scalable approach for improving timing recovery in next-generation high-speed PAM4 wireline, SerDes, optical interconnect, and data-center communication systems.

The proposed approach also has several limitations. First, the residual neural network is trained using behavioral data, and its correction accuracy therefore depends on how well the training conditions represent the channel, jitter, and implementation conditions encountered during operation. Significant changes in these conditions may require retraining or updating the stored network coefficients. Second, although the network is intentionally small, its

implementation introduces additional multiply–accumulate operations, coefficient storage, and fixed-point quantization compared with the conventional MMPD. In the behavioral waveform model, 32 samples per symbol are used to provide sufficient temporal resolution for pulse shaping, interpolation, and timing-offset characterization. This oversampling factor is a simulation-resolution parameter and does not correspond to the number of neural-network inputs or to the number of samples processed by the hardware inference engine. The network receives only a single normalized phase-error input and produces one residual correction per detector update. Consequently, increasing the number of waveform samples per symbol primarily increases behavioral simulation and training-data generation cost, approximately in proportion to the oversampling factor, while providing finer timing resolution; reducing it lowers simulation cost at the expense of a coarser representation of timing offsets. The neural-network architecture, its 97 trainable parameters, and its inference complexity remain unchanged. Finally, the present results are based primarily on behavioral and digital implementation studies, and experimental silicon validation remains necessary to quantify the effects of process, voltage, temperature, channel variation, and fixed-point implementation nonidealities.

Future work will explore both fixed-point and programmable hardware implementation of the neural network, RTL and ASIC realization of the complete AI-assisted CDR architecture, and experimental validation using FPGA and silicon measurements under practical high-speed communication channels. The programmable hardware implementation of the neural network inference circuit enables refining the training models through various stages of the CDR design and implementation. Since CDR architectures typically push the baud rate as high as possible in a given process node circuit timing, post layout circuit extraction with timing simulation methods can be used to train the neural network to optimize CDR performance at a given process node. CDR architectures can also be instrumented to collect experimental data to retrain the neural network to enhance yield at deep process nodes.

Author Contributions: S.G. developed concepts, CDR methodology, implementation, and writing of the proposed work as part of his postdoc work from his PhD Dissertation. S.B. conceptualized the use of neural network nonlinear filters in circuits from earlier work on associative memory circuits for programmable mixed signal circuits, and as the PhD advisor also contributed to writing and revising neural network use cases in mixed signal circuits in the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors acknowledge the support of the Department of Electrical and Computer Engineering at The Ohio State University. The authors also thank the paper reviewers for their helpful comments.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Zhao, X.; Chen, Y.; Mak, P.-I.; Martins, R.P. A 0.14-to-0.29-pJ/bit 14-GBaud/s Trimodal (NRZ/PAM-4/PAM-8) Half-Rate Bang-Bang Clock and Data Recovery Circuit in 28-nm CMOS. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 89–102. [CrossRef]
2. Zhao, X.; Chen, Y.; Wang, L.; Mak, P.-I.; Maloberti, F.; Martins, R.P. A Sub-0.25-pJ/bit 47.6-to-58.8-Gb/s Reference-Less FD-Less Single-Loop PAM-4 Bang-Bang CDR with a Deliberate-Current-Mismatch Frequency Acquisition Technique in 28-nm CMOS. *IEEE J. Solid-State Circuits* **2022**, *57*, 1358–1371. [CrossRef]

3. Aziz, P.M.; Surendran, S. Symbol Rate Timing Recovery for Higher Order Partial Response Channels. *IEEE J. Sel. Areas Commun.* **2001**, *19*, 635–644. [CrossRef]
4. Musa, F.A.; Carusone, A.C. Clock Recovery in High-Speed Multilevel Serial Links. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Bangkok, Thailand, 25–28 May 2003.
5. Balan, V.; Caroselli, J.; Chen, J.-G.; Desai, C.; Liu, C. A 4.8–6.4 Gb/s Serial Link for Back-Plane Applications Using Decision Feedback Equalization. *IEEE J. Solid-State Circuits* **2004**, *40*, 1957–1967.
6. Sonntag, J.L.; Stonick, J. A Digital Clock and Data Recovery Architecture for Multi-Gigabit/s Binary Links. *IEEE J. Solid-State Circuits* **2006**, *41*, 1867–1878. [CrossRef]
7. Spagna, F.; Chen, L.; Deshpande, M.; Fan, Y.; Gambetta, D.; Gowder, S.; Iyer, S.; Kumar, R.; Kwok, P.; Krishnamurthy, R.; et al. A 78 mW 11.8 Gb/s Serial Link Transceiver with Adaptive RX Equalization and Baud-Rate CDR in 32 nm CMOS. In Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 7–11 February 2010.
8. Francese, P.A.; Toifl, T.; Buchmann, P.; Brändli, M.; Menolfi, C.; Kossel, M.; Morf, T.; Kull, L.; Andersen, T.M. A 16 Gb/s 3.7 mW/Gb/s 8-Tap DFE Receiver and Baud-Rate CDR with 31 kppm Tracking Bandwidth. *IEEE J. Solid-State Circuits* **2014**, *49*, 2490–2503. [CrossRef]
9. Musah, T.; Jaussi, J.E.; Balamurugan, G.; Hyvonen, S.; Hsueh, T.-C.; Keskin, G.; Shekhar, S.; Kennedy, J.; Sen, S.; Inti, R. et al. A 4–32 Gb/s Bidirectional Link with 3-Tap FFE/6-Tap DFE and Collaborative CDR in 22 nm CMOS. *IEEE J. Solid-State Circuits* **2014**, *49*, 3079–3092. [CrossRef]
10. Balachandran, A.; Chen, Y.; Boon, C.C. A 32-Gb/s 3.53-mW/Gb/s Adaptive Receiver AFE Employing a Hybrid CTLE, Edge-DFE and Merged Data-DFE/CDR in 65-nm CMOS. In *IEEE Asian Solid-State Circuits Conference (A-SSCC)*; IEEE: Piscataway, NJ, USA, 2018.
11. Ghosh, S.; Kaje, S.; Bibyk, S. Performance of a Baud-Rate Clock Data Recovery with Analog Time Encoding. In Proceedings of the 2026 IEEE 69th International Midwest Symposium on Circuits and Systems (MWSCAS), Cincinnati, OH, USA, 9–12 August 2026. [CrossRef]
12. Rutenbar, R.A.; Gielen, G.G.E.; Roychowdhury, J. Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs. *Proc. IEEE* **2007**, *95*, 640–669. [CrossRef]
13. Musah, T.; Namachivayam, A. Robust Timing Error Detection for Multilevel Baud-Rate CDR. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 3927–3939. [CrossRef]
14. Musa, F.A.; Carusone, A.C. Modeling and Design of Multilevel Bang–Bang CDRs in the Presence of ISI and Noise. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2007**, *54*, 2137–2149. [CrossRef]
15. Hossain, M.; Aurangozeb; Nguyen, N. DDJ-Adaptive SAR TDC-Based Timing Recovery for Multilevel Signaling. *IEEE J. Solid-State Circuits* **2019**, *54*, 2833–2845. [CrossRef]
16. Verbeke, M.; Torfs, G.; Rombouts, P. The Truth About 2-Level Transition Elimination in Bang-Bang PAM-4 CDRs. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 469–482. [CrossRef]
17. Xu, D.; Kou, Y.; Lai, P.; Cheng, Z.; Cheung, T.Y.; Moser, L.; Zhang, Y.; Liu, X.; Lam, M.P.; Jia, H.; et al. A Scalable Adaptive ADC/DSP-Based 1.25-to-56 Gb/s/112 Gb/s High-Speed Transceiver Architecture Using Decision-Directed MMSE CDR in 16 nm and 7 nm. In Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 13–22 February 2021.
18. Liu, T.; Li, T.; Lv, F.; Liang, B.; Zheng, X.; Wang, H.; Wu, M.; Lu, D.; Zhao, F. Analysis and Modeling of Mueller–Muller Clock and Data Recovery Circuits. *Electronics* **2021**, *10*, 1888. [CrossRef]
19. Bibyk, S.; Noonan, A.; Ghosh, S.; Manning, C.; Sarkar, S. Analog Re-Use in Mixed Signal Circuits via Analog Time Encoding in Shared Die Tapeouts. In Proceedings of the 2024 IEEE 67th International Midwest Symposium on Circuits and Systems (MWSCAS), Springfield, MA, USA, 11–14 August 2024. [CrossRef]
20. Kaje, S.; Ghosh, S.; Bibyk, S. A Low-Complexity, Synthesizable ADPLL for Resource-Constrained Mixed-Signal Systems. In Proceedings of the 2026 IEEE 69th International Midwest Symposium on Circuits and Systems (MWSCAS), Cincinnati, OH, USA, 9–12 August 2026. [CrossRef]
21. Ghosh, S.; Bibyk, S. Mixed Signal Cross Node Porting Using Digital Twinning with AI and Analog Time Encoding. In Proceedings of the NAECON 2026—IEEE National Aerospace and Electronics Conference, Cincinnati, OH, USA, 9–12 August 2026. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.