Deep Reinforcement Learning-Based DRAM Equalizer Parameter Optimization Using Latent Representations

Muhammad Usama and Dong Eui Chang
Control Laboratory, School of Electrical Engineering, KAIST,
Daejeon, 34141, Republic of Korea

***Abstract*—Equalizer parameter optimization for signal integrity in high-speed Dynamic Random Access Memory systems is crucial but often computationally demanding or model-reliant. This paper introduces a data-driven framework employing learned latent signal representations for efficient signal integrity evaluation, coupled with a model-free Advantage Actor-Critic reinforcement learning agent for parameter optimization. The latent representation captures vital signal integrity features, offering a fast alternative to direct eye diagram analysis during optimization, while the reinforcement learning agent derives optimal equalizer settings without explicit system models. Applied to industry-standard Dynamic Random Access Memory waveforms, the method achieved significant eye-opening window area improvements: 42.7% for cascaded Continuous-Time Linear Equalizer and Decision Feedback Equalizer structures, and 36.8% for Decision Feedback Equalizer-only configurations. These results demonstrate superior performance, computational efficiency, and robust generalization across diverse Dynamic Random Access Memory units compared to existing techniques. Core contributions include an efficient latent signal integrity metric for optimization, a robust model-free reinforcement learning strategy, and validated superior performance for complex equalizer architectures.**

***Index Terms*—Signal Integrity, DRAM, Equalization, Decision Feedback Equalizer, Reinforcement Learning, Latent Representations, Autoencoder, Parameter Optimization, A2C**

I. Introduction

THE increasing demand for higher data rates in modern Dynamic Random Access Memory (DRAM) systems presents significant challenges in maintaining robust signal integrity (SI). As data rates rise, signal distortions, most notably inter-symbol interference (ISI), become increasingly severe, degrading the quality of signals received by DRAM modules. Equalizers, such as continuous-time linear equalizers (CTLE), feed-forward equalizers, and decision feedback equalizers (DFE), are essential for compensating channel impairments and restoring SI. However, the optimization of equalizer parameters is a complex and critical task that directly impacts system reliability and performance.

Traditional equalizer optimization methods rely on eye diagram analysis combined with exhaustive or heuristic parameter search algorithms, such as genetic algorithms, to determine optimal settings for mitigating ISI and channel loss. While effective, these approaches are computationally intensive and lack adaptability to dynamic channel conditions. Adaptive algorithms based on the least-mean-square criterion [1], [2] offer computational efficiency but require accurate channel models and explicit error signals, and may converge to suboptimal local minima in channels with severe ISI [3]. Alternative heuristic [4], [5] and iterative [6], [7] optimization strategies have also been explored, but their ability to generalize across diverse operational scenarios is limited [8], [9].

Recent advances in machine learning have introduced new paradigms for equalizer parameter optimization. Surrogate modeling, using deep neural networks and support vector regression, has accelerated SI analysis and parameter estimation [10]–[12]. Autoencoders and generative models have been used to compress eye diagram information into latent codes, enabling efficient SI metrics [13], [14]. Reinforcement learning (RL), particularly actor-critic methods, have been applied to equalizer optimization in high-bandwidth memory links, demonstrating improvements over traditional search-based methods [15]–[18]. The integration of latent-space representations with RL frameworks has further enhanced learning stability and efficiency.

Despite these advances, several challenges remain. Eye diagram-based SI evaluation is computationally expensive and not well-suited for rapid optimization [19]. Many optimization algorithms require accurate mathematical models of both the channel and equalizer, which are often difficult to obtain in practice. Furthermore, existing machine learning and RL-based approaches may require extensive training data or struggle to balance optimization effectiveness with computational efficiency, especially for complex equalizer structures.

This work addresses these challenges by proposing a purely data-driven framework for equalizer parameter optimization in high-speed DRAM systems. We develop an efficient SI metric based on learned latent representations, eliminating the need for computationally expensive eye diagram analysis. Equalizer parameter optimization is formulated as a model-free RL problem using the Advantage Actor-Critic (A2C) algorithm, with the latent representation of the signal as the state and the equalizer parameters as continuous actions. The reward is defined by the proximity of the equalized signal's latent representation to that of an ideal signal.

Our contributions are as follows: (i) a novel SI evaluation method that captures relevant signal characteristics in a learned latent space, providing both accuracy and computational effi-

M. Usama and D. E. Chang are with the Control Laboratory, School of Electrical Engineering, KAIST, Daejeon, 34141, Republic of Korea (e-mail: usama@kaist.ac.kr; dechang@kaist.ac.kr).

ciency; (ii) a model-free RL formulation for equalizer parameter optimization that adapts to system characteristics without requiring explicit mathematical models; and (iii) extensive experimental validation demonstrating superior performance and efficiency compared to traditional and machine learning-based methods for both DFE and cascaded CTLE+DFE structures.

The remainder of this paper is organized as follows. Section II describes the DRAM waveform dataset. Section III details our latent representation-based SI evaluation method and RL-based optimization framework. Section IV presents the experimental setup and baseline methods. Section V analyzes the experimental results. Section VI discusses key insights and implications, and Section VII concludes the paper with future research directions.

II. Dataset

The dataset used in this study comprises signal waveforms representing data transmission between the central processing unit and DRAM within a server environment operating at 6400 Mbps. These waveforms were generated through simulations of write operations, utilizing registered dual in-line memory commonly found in server systems. The dataset contains a total of 300,000 recorded values for both input and output data pairs for each of the 8 DRAMs, resulting in a total of 4.8 million samples. Each set of data gathered included two waveforms for each of the eight DRAMs: the initial input waveform written by the CPU and the output waveform, which represents the degraded signal received by the DRAM after traversing the server system components.

The dataset is organized into samples, where each sample consists of 10,000 consecutive values, $n_x = 10000$, representing the input waveform written by the central processing unit and the corresponding output waveform received by the DRAM. Samples were constructed using a rolling window approach with a single-step increment. Both the input and output waveforms were sampled at a rate of 10 ps, with one unit interval spanning 156.3 ps. To generate eye diagrams for visual analysis, interpolation was applied to the recorded values, ensuring a time separation of 1 ps between consecutive data points. The overall data collection setup and a visualization of the dataset are shown in Fig. 1.

Each output data sample $d_o \in \mathbb{R}^{n_x}$ was labeled with a binary value indicating its validity, determined through eye diagram analysis. A rectangular window measuring 80 mV in height and 35 ps in width was defined within the eye-opening region. If the signal of a data sample intersects this window, it is labeled as invalid ($y = 0$); otherwise, it is considered valid ($y = 1$), as illustrated in Figure 2.

III. Proposed Methodology

We propose a two-stage framework for equalizer parameter optimization: (i) latent representation-based signal integrity (SI) evaluation, and (ii) reinforcement learning (RL) based equalizer optimization using the Advantage Actor-Critic (A2C) algorithm. The approach is designed for computational efficiency and robustness to channel/model uncertainties.

Algorithm 1 Autoencoder Training with Valid-Only Classification Gradients

Require: Training dataset $\mathscr{D} = \{(\mathbf{x}, y)\}$, latent dimension l
Ensure: Trained encoder $\ell(\cdot)$

```
for each batch (x, y) from D do
    z ← ℓ(x)
    x̂ ← g(z)
    ŷ ← c(z)
    L_r = ||x − x̂||_2^2
    if y = 1 then
        L_c = −y log(ŷ)
        Update networks using L_r + L_c
    else
        Update networks using L_r
    end if
end for
return Trained encoder ℓ(·)
```

A. Latent Representation-Based SI Evaluation

Traditional SI evaluation via eye diagrams is computationally expensive. We instead employ a learned latent space representation for rapid SI assessment. Let $\mathbf{d} \in \mathbb{R}^{n_x}$ denote a waveform segment. An autoencoder with encoder $\ell(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}^l$ and decoder $g(\cdot) : \mathbb{R}^l \to \mathbb{R}^{n_x}$ is trained to map $\mathbf{d}$ to a latent vector $\mathbf{z} = \ell(\mathbf{d})$. The architecture uses fully connected layers with ReLU activations and a symmetric decoder, as shown in Fig. 3.

To ensure SI-relevant features are captured, the reconstruction objective is augmented with a classification loss. A classifier attached to the encoder output predicts the validity of $\mathbf{d}$ based on eye mask criteria. The combined loss is

$$\mathscr{L} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 - y \cdot \log(\hat{y}), \tag{1}$$

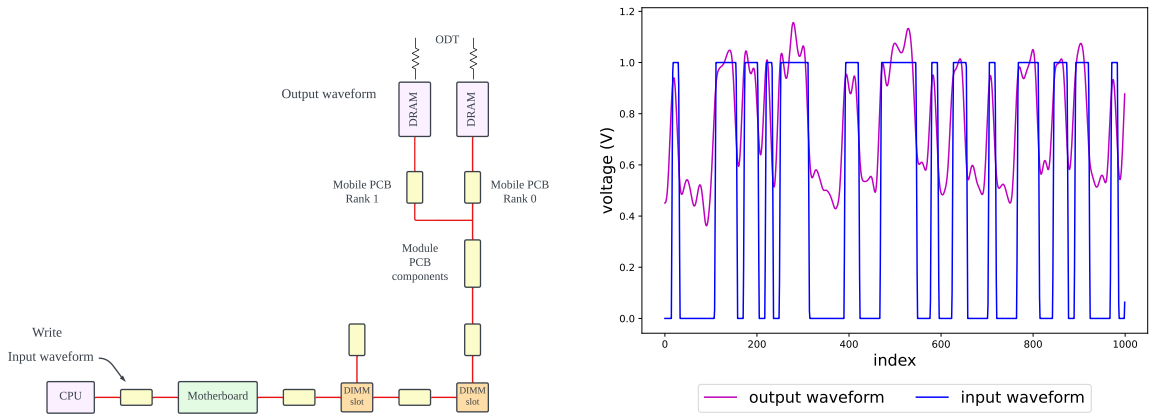
where $\mathbf{x}$ is the input, $\hat{\mathbf{x}}$ the reconstruction, y the validity label, and $\hat{y}$ the classifier output. Gradients from the classifier are only backpropagated for valid signals ($y = 1$), enforcing tight clustering of valid representations in latent space. The autoencoder training procedure, including the valid-only classification gradient strategy, is summarized in Algorithm 1, and the overall training architecture is illustrated in Fig. 4.

Let $\mathbb{S} = \left\{ \ell(\mathbf{d}) \in \mathbb{R}^l \mid \mathbf{d} \in \mathbb{R}^{n_x},\ y = 1 \right\}$ denote the set of latent vectors corresponding to all valid signals in the dataset. The anchor point $\mathbf{c} \in \mathbb{R}^l$ is computed as the Fermat-Weber point of $\mathbb{S}$:

$$\mathbf{c} = \arg\min_{\mathbf{k} \in \mathbb{S}} \sum_{j=1}^{m} \|\ell(\mathbf{d}_j) - \mathbf{k}\|_2. \tag{2}$$

B. Reinforcement Learning-Based Equalizer Parameter Optimization

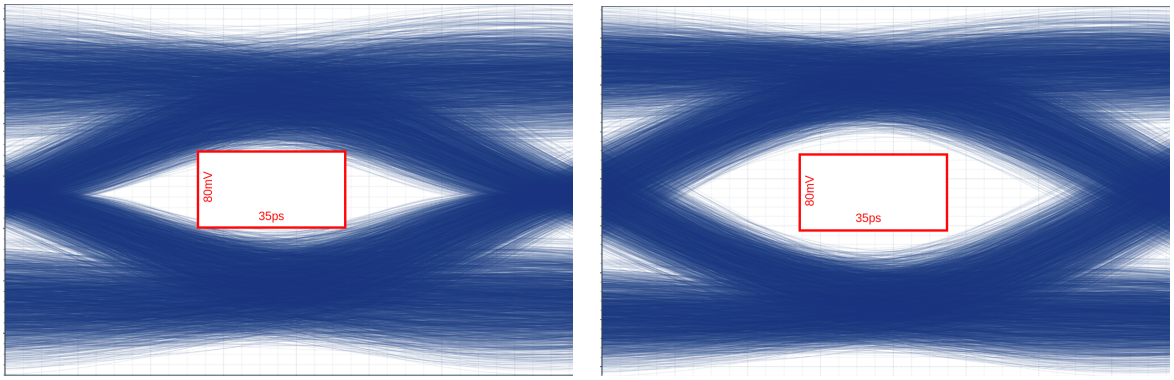
Equalizer parameter optimization is formulated as an episodic Markov Decision Process (MDP). The state space $\mathscr{S}$ comprises latent representations $\mathbf{s} = \ell(\mathbf{d}_o)$ of the output waveform $\mathbf{d}_o$. The action space $\mathscr{A} = [0,1]^d$ consists of d-dimensional vectors of normalized values, where d is the number of equalizer parameters (e.g., $d = 4$ for DFE, $d = 8$ for CTLE+DFE). These normalized actions $\mathbf{a}_t \in \mathscr{A}$ are transformed into actual



(a) Data collection setup.

(b) Visualization of the collected dataset.

Fig. 1. The figure shows (a) the server memory system with double-sided DIMMs used to generate our dataset, and (b) a visualization of 1000 sample values for DRAM 1 from the dataset that plots the DRAM output waveform and the corresponding input waveform.



(a) Example of an invalid signal ($y=0$)

(b) Example of a valid signal ($y=1$)

Fig. 2. Illustration of the signal validity labeling criteria. The rectangular window (80 mV *times* 35 ps) is shown in red. (a) An invalid signal where signal transitions intersect the window, and (b) a valid signal where no transitions occur within the window region.

equalizer parameters $\mathbf{p}_t$ through a predefined mapping function $M:[0,1]^d \to \mathscr{P}_{actual}$, where $\mathscr{P}_{actual}$ represents the space of valid physical parameter ranges for the specific equalizer. Thus, $\mathbf{p}_t = M(\mathbf{a}_t)$.

The environment transition is deterministic: applying the mapped parameters $\mathbf{p}_t = M(\mathbf{a}_t)$ to the current output signal $\mathbf{d}_o$ yields the equalized output signal $\mathbf{d}_o^e = \mathrm{EQ}(\mathbf{d}_o, \mathbf{p}_t)$. The next state is obtained by encoding this equalized signal, $\mathbf{s}_{t+1} = \ell(\mathbf{d}_o^e)$. The reward is the negative Euclidean distance in latent space to the anchor point $\mathbf{c}$:

$$R(\mathbf{s}_t, \mathbf{a}_t) = -\|\mathbf{c} - \ell(\mathrm{EQ}(\mathbf{d}_o, M(\mathbf{a}_t)))\|_2, \tag{3}$$

where $\mathrm{EQ}(\cdot,\cdot)$ denotes equalization applied to the signal $\mathbf{d}_o$ using the mapped parameters $M(\mathbf{a}_t)$.

We use the A2C algorithm with entropy regularization. The actor parameterizes a Gaussian policy $\pi_\theta(\mathbf{a}|\mathbf{s})$; the critic estimates $V_\omega(\mathbf{s})$. The advantage is

$$A(\mathbf{s}_t, \mathbf{a}_t) = r_t + \gamma V_\omega(\mathbf{s}_{t+1}) - V_\omega(\mathbf{s}_t). \tag{4}$$

The joint loss is

$$\begin{aligned}\mathscr{L}(\theta,\omega) = &-\mathbb{E}_{\mathbf{s}_t,\mathbf{a}_t}\left[A(\mathbf{s}_t,\mathbf{a}_t)\log\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\right] \\ &+\frac{c_v}{2}\mathbb{E}_{\mathbf{s}_t}\left[\left(r_t+\gamma V_\omega(\mathbf{s}_{t+1})-V_\omega(\mathbf{s}_t)\right)^2\right] \\ &-\beta\mathbb{E}_{\mathbf{s}_t}\left[H(\pi_\theta(\cdot|\mathbf{s}_t))\right],\end{aligned} \tag{5}$$

where c_v and β are weighting coefficients.

The overall A2C-based optimization process is detailed in Algorithm 2. Actor and critic networks are fully connected, taking $\mathbf{s} \in \mathbb{R}^l$ as input. The actor outputs mean and log standard deviation for the Gaussian policy; the critic outputs the value estimate. The network architectures for both actor and critic are illustrated in Fig. 5. During inference, the mean action is selected as the optimal parameter vector.

IV. Experimental Setup

The proposed framework was evaluated on two equalizer configurations. Both equalizers were implemented in Python using NumPy. The first configuration is a 4-tap DFE. The DFE output $y[n]$ for a received signal $r[n]$ is $y[n] = r[n] - \sum_{i=1}^{4} t_i \hat{s}[n-i]$, where $t_i \in [0,1]$ are the tap weights and $\hat{s}[n-i]$ are prior hard decisions. The parameter vector is $\mathbf{p}_{DFE} = \{t_1, t_2, t_3, t_4\}$, with

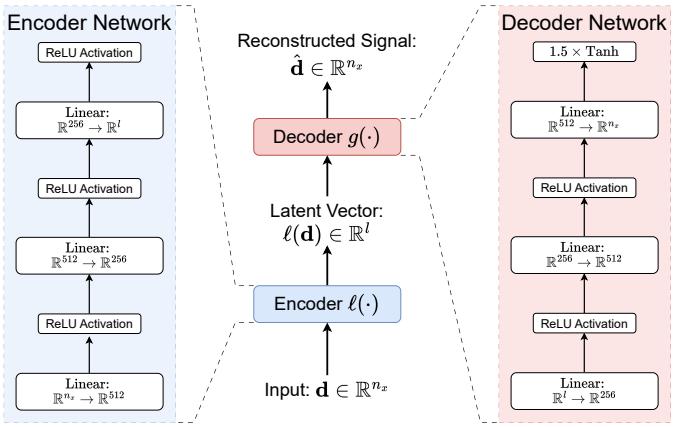


Fig. 3. Autoencoder network architecture for latent SI evaluation. The encoder maps the input waveform to a latent vector via three fully connected layers with ReLU activations. The decoder reconstructs the waveform using a symmetric structure and a scaled tanh output.

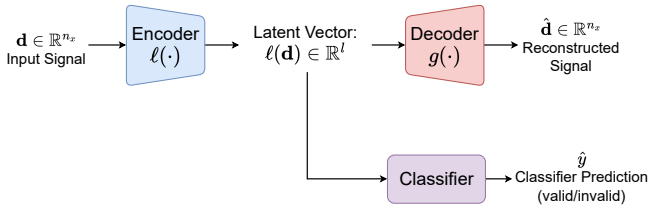


Fig. 4. Autoencoder training architecture showing the encoder-decoder structure augmented with a classification head. The classifier gradients are only backpropagated for valid signals ($y = 1$) to enforce tight clustering in the latent space.

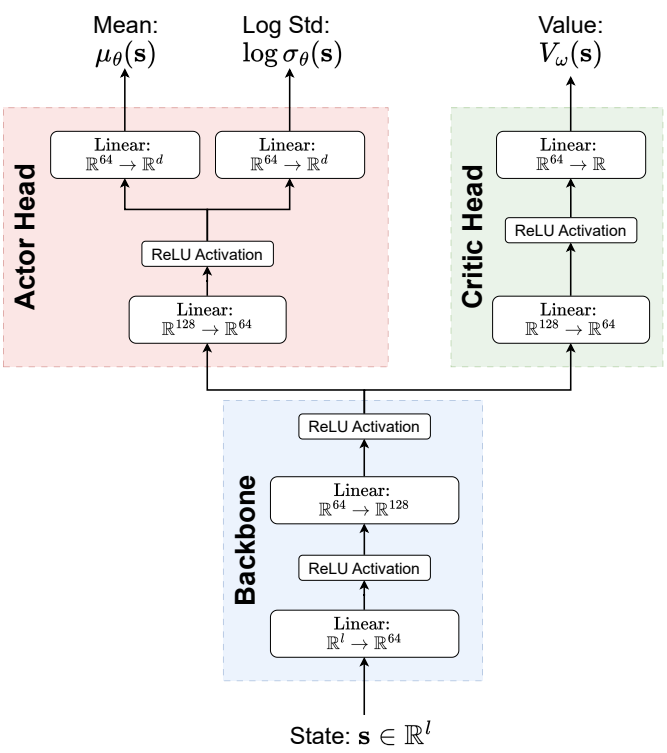


Fig. 5. Network architectures for the actor and critic in the A2C framework. Both networks share a backbone of fully connected layers with ReLU activations. The actor outputs the mean and log standard deviation for the Gaussian policy; the critic outputs the state value estimate.

tap weights clipped to $[0,1]$. The second configuration is a cascaded CTLE followed by a 4-tap DFE. The CTLE is modeled by $H(s) = G_{dc} \cdot \frac{s+\omega_z}{s+\omega_p} \cdot \frac{\omega_p}{\omega_z}$, with DC gain $G_{dc} \in [0,10]$, zero frequency $f_z \in [0,1]$ GHz, pole frequency $f_p \in [0,10]$ GHz, and peaking gain $G_p \in [0,20]$ dB. It is discretized using the bilinear transform. The DFE then processes the CTLE output. The parameter vector is $\mathbf{p}_{CTLE+DFE} = \{G_{dc}, f_z, f_p, G_p, t_1, t_2, t_3, t_4\}$.

The autoencoder for latent SI evaluation, detailed in Section III-A and Fig. 3, was trained for 200 epochs using the Adam optimizer to minimize the combined loss in Eq. 1. Key hyperparameters are listed in Table IV. The resulting training loss is shown in Fig. 6.

The A2C RL agent optimized equalizer parameters, as formulated in Section III-B. Actor and critic networks (Fig. 5) utilized three fully connected layers with ReLU activations and were trained with the Adam optimizer. The RL training followed a one-step MDP formulation (Algorithm 2) for up to 300 epochs, processing the dataset in batches, with early stopping based on reward convergence. Each batch processing constituted one training step. A2C hyperparameters are detailed in Table IV. The training loss and reward curves for the RL agent are shown in Fig. 7 and Fig. 8, respectively.

Performance was assessed by: (i) Average Window Area Improvement (%), calculated as the percentage increase in the largest rectangular eye-opening area (via PyEye package [20]), and (ii) Relative Computational Time, normalized to the fastest method. Baselines included Particle Swarm Optimization (PSO), Q-learning, Bayesian optimization [21], policy optimization [22], and genetic algorithms. Further implementation details for the genetic algorithm, Bayesian optimization [21], policy optimization [22], and Q-learning are provided in Appendices A, B, C, and D, respectively.

TABLE I
HYPERPARAMETERS FOR AUTOENCODER AND RL AGENT TRAINING

Component	Hyperparameter	Value
Autoencoder	Latent Dimension (l)	11
	Learning Rate	1×10^{-3}
	Weight Decay	1×10^{-5}
	Batch Size	256
	Epochs	200
A2C RL Agent	Learning Rate	5×10^{-4}
	Discount Factor (γ)	0.98
	Entropy Coefficient (β)	1×10^{-2}
	Value Loss Coefficient (c_v)	0.5
	Epochs	300
	Batch Size	64

Algorithm 2 A2C-Based Equalizer Parameter Optimization

Require: Pre-trained Encoder $\ell(\cdot)$, anchor point $\mathbf{c}$, training data $\mathscr{D}_{train} = \{\mathbf{d}_o\}$, actor π_θ, critic V_ω, epochs E, batch size B, learning rate α, discount factor γ, action mapping $M(\cdot)$

Ensure: Trained actor π_θ

for epoch $e = 1$ to E **do**
  Shuffle $\mathscr{D}_{train}$ (optional)
  for each batch $\{\mathbf{d}_o^{(k)}\}_{k=1}^{B}$ from $\mathscr{D}_{train}$ **do**
    for $k = 1$ to B **do**
      $\mathbf{s}_{curr}^{(k)} \leftarrow \ell(\mathbf{d}_o^{(k)})$ ▷ Get current state
      $\mathbf{a}^{(k)} \sim \pi_\theta(\cdot|\mathbf{s}_{curr}^{(k)})$ ▷ Sample action
      $\mathbf{p}^{(k)} \leftarrow M(\mathbf{a}^{(k)})$ ▷ Map action to parameters
      $\mathbf{d}_{o,eq}^{(k)} \leftarrow \mathrm{EQ}(\mathbf{d}_o^{(k)}, \mathbf{p}^{(k)})$ ▷ Equalize signal
      $\mathbf{s}_{next}^{(k)} \leftarrow \ell(\mathbf{d}_{o,eq}^{(k)})$ ▷ Get next state
      $r^{(k)} \leftarrow -\|\mathbf{c} - \mathbf{s}_{next}^{(k)}\|_2$ ▷ Calculate reward
      $A^{(k)} \leftarrow r^{(k)} + \gamma V_\omega(\mathbf{s}_{next}^{(k)}) - V_\omega(\mathbf{s}_{curr}^{(k)})$ ▷ Calculate advantage
    end for
    Update θ, ω using batch gradients of $\mathscr{L}(\theta, \omega)$ from (5)
  end for
end for
return Trained actor π_θ

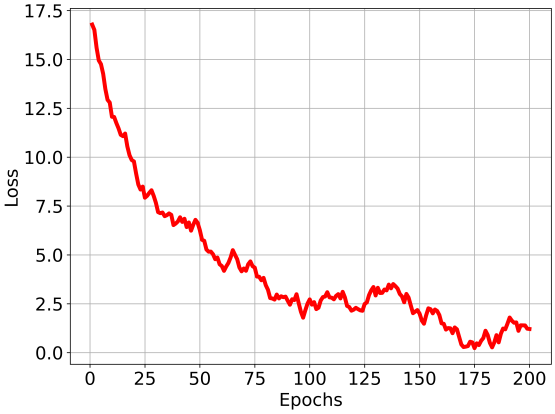


Fig. 6. Training loss curve for the autoencoder network.

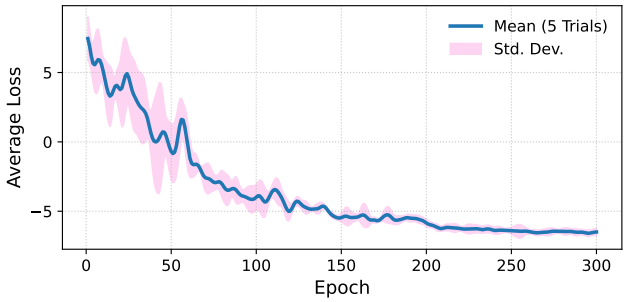


Fig. 7. Training loss curve for the RL agent.

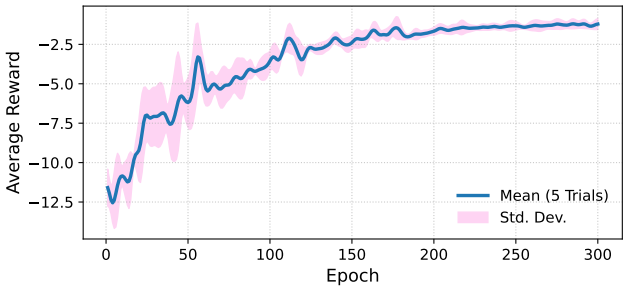


Fig. 8. Training reward curve for the RL agent.

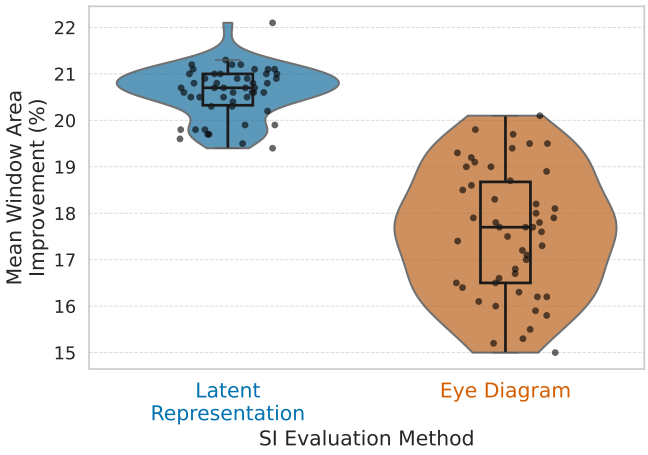


Fig. 9. Distribution of mean window area improvement (%) for DFE parameter optimization using the latent representation SI metric (blue, left) and the conventional eye diagram SI metric (orange, right) across 50 independent PSO trials. The violin plot shows the data distribution, the box plot indicates median and quartiles, and individual trial results are overlaid.

V. Results

We evaluated our proposed methods on optimizing parameters for two equalizer structures using DRAM waveform data provided by Samsung.

A. Latent Representation SI vs. Eye Diagram SI

The latent representation-based SI metric was quantitatively compared to the conventional eye diagram-based SI evaluation for DFE parameter optimization using 50 independent PSO trials per method. As shown in Fig. 9, the latent representation approach consistently yields higher mean window area improvements with a tighter distribution. Specifically, Table II reports an average improvement of 20.68% (standard deviation 0.54%) for the latent method, compared to 17.54% (standard deviation 1.34%) for the eye diagram approach. Furthermore, the latent representation metric achieves a 51× reduction in computational time. These results demonstrate that the latent representation SI metric provides both superior and more consistent optimization performance with significantly greater efficiency.

B. Results for Optimizing 4-Tap DFE Structure

Table III summarizes the quantitative results for 4-tap DFE optimization. The A2C-based RL method achieved the highest window area improvement at 36.8% with the lowest baseline computational cost 1.0×. Q-learning reached 26.1%

TABLE II
COMPARISON OF DFE OPTIMIZATION PERFORMANCE USING LATENT REPRESENTATION SI VS. EYE DIAGRAM SI (BASED ON 50 PSO TRIALS)

SI Evaluation Method	Average Window Area Improvement (%)	Standard Deviation of Improvement (%)	Relative Computational Time
Latent Representation	20.68	0.54	1×
Eye Diagram	17.54	1.34	51×

TABLE III
COMPARISON OF OPTIMIZATION METHODS FOR 4-TAP DFE

Method	Average Window Area Improvement (%)	Relative Computational Time
Ours (RL - A2C)	**36.8%**	**1.0×**
Q-learning	26.1	7.0×
PSO (Latent Rep.)	19.8	2.0×
Grid Search	13.8	8.5×
Policy Opt. [22]	15.5	4.0×
Bayesian Opt. [21]	11.7	5.0×
Genetic Algorithm	14.2	4.5×

TABLE IV
COMPARISON OF OPTIMIZATION METHODS FOR CASCADED CTLE+DFE

Method	Average Window Area Improvement (%)	Relative Computational Time
Ours (RL - A2C)	**42.7**	**1.0×**
Q-learning	28.5	13.0×
PSO (Latent Rep.)	25.4	3.0×
Grid Search	15.2	32.0×
Policy Opt. [22]	21.3	5.0×
Bayesian Opt. [21]	18.9	6.5×
Genetic Algorithm	20.5	5.5×

improvement but required 7.0× more computation. PSO with latent representation yielded 19.8% improvement at 2.0× cost. Grid search and policy optimization [22] achieved 13.8% and 15.5% improvements, with computational costs of 8.5× and 4.0×, respectively. Bayesian optimization [21] and the genetic algorithm resulted in 11.7% and 14.2% improvements, with 5.0× and 4.5× computational time, respectively.

C. Results for Optimizing Cascaded Equalizer Structure

Table IV summarizes the results for the cascaded CTLE+DFE structure. The A2C-based RL method achieved the highest window area improvement at 42.7% with the lowest baseline computational cost 1.0×. Q-learning resulted in 28.5% improvement at 13.0× computational time. PSO with latent representation achieved 25.4% at 3.0× cost. Policy optimization [22] and genetic algorithm yielded 21.3% and 20.5% improvements, with computational costs of 5.0× and 5.5×, respectively. Grid search and Bayesian optimization [21] achieved 15.2% and 18.9% improvements, with 32.0× and 6.5× computational time, respectively.

D. Visualization of Latent Space and Equalization Effect

Figure 10 shows the two-dimensional t-SNE embeddings [23] of signal latent representations, where valid signals form

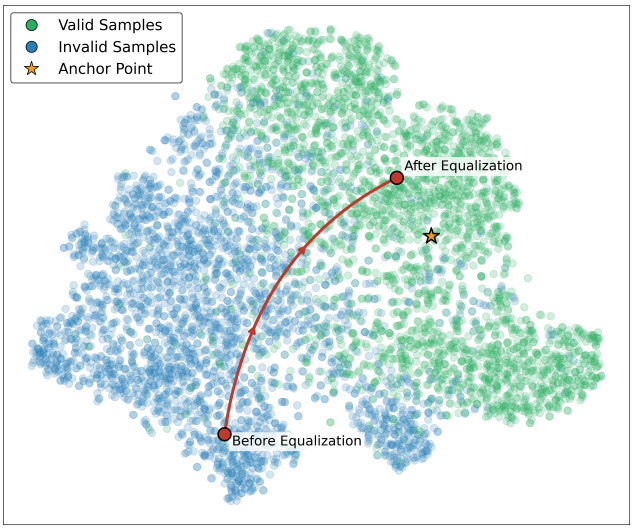


Fig. 10. Two-dimensional t-SNE visualization of latent representations showing valid signals (green), invalid signals (blue), anchor point (gold star), and equalization trajectory (red line) from invalid to valid cluster.

a cluster (green points), invalid signals form another cluster (blue points), and the anchor point (gold star) is positioned within the valid cluster. A sample signal's trajectory (red line) demonstrates the equalization effect: the signal starts in the invalid cluster and moves to the valid cluster after applying the optimized DFE parameters, ending near the anchor point. The clear spatial separation between valid and invalid signals in the latent space confirms the effectiveness of our SI assessment approach.

E. Latent Space Dimensionality

We evaluated latent dimensions from 5 to 20 for both equalizer configurations (Figure 11). For DFE, the mean window area improvement increased from 22.5% (dimension 5) to 36.8% (dimension 11), with subsequent improvements below 1.2 percentage points up to dimension 20. For CTLE+DFE, the improvement increased from its initial value to 42.7% (dimension 11), with subsequent improvements below 0.3 percentage points. Therefore, we selected 11 as the latent dimension for our experiments.

F. Generalization Across Different DRAM Units

To evaluate the robustness of our approach and its ability to generalize, we conducted a systematic evaluation following a rigorous train-test protocol. The development phase utilized data from DRAMs 1-6, during which we trained both the autoencoder and the RL agent. These six DRAMs provided

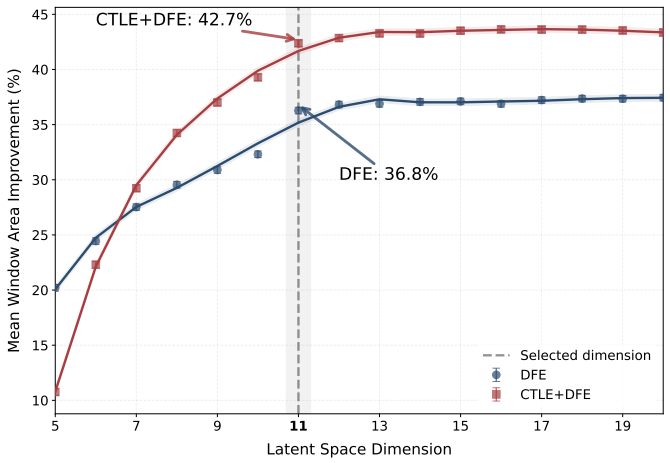


Fig. 11. Mean Window area improvement versus latent space dimension. At dimension 11, DFE achieves 36.8% and CTLE+DFE achieves 42.7% improvement.

TABLE V
GENERALIZATION RESULTS: PERFORMANCE COMPARISON BETWEEN TRAINING AND HELD-OUT DRAMS

Evaluation Set	Mean Window Area Improvement (%)	
	DFE	CTLE+DFE
Training DRAMs (1–6)	36.3	42.7
Held-Out DRAMs (7–8)	33.7	39.8
Generalization Gap	2.6	2.9

all the data needed for model training, hyperparameter tuning, and architectural decisions. DRAMs 7-8 were completely held out from this development process, serving as independent test units to assess true generalization performance.

We evaluated our trained models in two contexts: first on the familiar DRAMs 1-6 used during training, and then on the completely unseen DRAMs 7-8. Table V presents these comprehensive results. For the DFE configuration, performance on the familiar DRAMs achieved a 36.3% improvement in window area, while the held-out DRAMs showed a 33.7% improvement. This modest 2.6 percentage point difference suggests strong generalization. Similarly, the CTLE+DFE configuration maintained robust performance, with a 42.7% improvement on familiar DRAMs decreasing only slightly to 39.8% on the held-out units, representing a small 2.9 percentage point gap.

These results demonstrate two important characteristics of our approach. First, the CTLE+DFE configuration consistently outperforms the DFE-only structure across both familiar and unseen DRAMs, maintaining its advantage regardless of the evaluation context. Second, the small performance degradation when moving to completely new DRAM units indicates that our method has learned generally applicable optimization strategies rather than overfitting to specific unit characteristics.

The robust generalization capability shown by both equalizer configurations is particularly important for practical applications. Manufacturing variations between DRAM units are inevitable, and an optimization approach must work effectively across these variations. The minimal performance drop observed on the independent test DRAMs suggests our method can readily handle such manufacturing tolerances, making it suitable for real-world deployment.

VI. DISCUSSION

The experimental results highlight several key advantages of the proposed methodology for equalizer parameter optimization in high-speed DRAM interfaces. First, the latent representation-based signal integrity evaluation proves to be a highly efficient alternative to traditional eye diagram analysis. As detailed in Table II, this approach not only achieves superior mean window area improvement (20.68% vs. 17.54%) with lower variance but also offers a significant 51× reduction in computational time. The t-SNE visualization in Fig. 10 further substantiates the efficacy of the autoencoder, illustrating a clear demarcation between valid and invalid signal clusters in the latent space and the strategic placement of the anchor point within the valid region. This confirms that the learned latent space effectively captures salient SI characteristics.

Second, the A2C reinforcement learning framework exhibits superior optimization performance compared to established baseline methods for both DFE and cascaded CTLE+DFE equalizer structures. For the 4-tap DFE, the A2C agent achieved a 36.8% improvement in window area, surpassing Q-learning (26.1%) and PSO with latent representation (19.8%), while also demonstrating the lowest relative computational cost (Table III). This performance advantage is even more pronounced for the more complex 8-parameter CTLE+DFE configuration, where A2C achieved a 42.7% improvement, compared to 28.5% for Q-learning and 25.4% for PSO (Table IV). This underscores the A2C agent's capability to effectively navigate higher-dimensional parameter spaces, a critical attribute for optimizing sophisticated equalizer designs.

Third, the proposed method demonstrates robust generalization capabilities. The performance evaluation on held-out DRAM units (DRAMs 7-8), which were not used during training, revealed only a marginal decrease in window area improvement compared to the training units (DRAMs 1-6). Specifically, the generalization gap was 2.6 percentage points for the DFE configuration and 2.9 percentage points for the CTLE+DFE configuration, as shown in Table V. This indicates that the learned optimization policies are not overfitted to the characteristics of specific DRAM units but rather capture more generalizable features, making the approach resilient to typical manufacturing variations.

Finally, the ablation study on latent space dimensionality (Fig. 11) identified a dimension of 11 as providing an optimal balance between model expressiveness and computational overhead. While increasing dimensionality beyond 11 yielded marginal gains in window area improvement (less than 1.2 percentage points for DFE and 0.3 percentage points for CTLE+DFE), it would incur additional computational costs. The consistent performance improvements and computational efficiency observed across different equalizer structures and DRAM units highlight the practical viability of the proposed framework for optimizing high-speed serial links.

VII. Conclusion

This work presented a data-driven framework for optimizing equalizer parameters in high-speed DRAM interfaces, combining an efficient SI evaluation metric from learned latent signal representations with a model-free A2C reinforcement learning agent. The proposed method achieved substantial eye-opening window area improvements of 42.7% for a cascaded CTLE+DFE and 36.8% for a DFE-only structure. These results significantly outperformed traditional and other learning-based baselines while demonstrating superior computational efficiency and robust generalization across different DRAM units.

References

[1] K. Azadet, E. F. Haratsch, H. Kim, F. Saibi, J. H. Saunders, M. Shaffer, L. Song, and M. L. Yu, "Equalization and fec techniques for optical transceivers," IEEE Journal of Solid-State Circuits, vol. 37, no. 3, pp. 317–327, 2002.

[2] S. U. H. Qureshi, "Adaptive equalization," Proceedings of the IEEE, vol. 73, no. 9, pp. 1349–1387, 1985.

[3] J. G. Proakis and M. Salehi, Digital Communications, 5th ed. McGraw-Hill, 2007.

[4] R. Lopes and J. Barry, "The soft-feedback equalizer for turbo equalization of highly dispersive channels," IEEE Transactions on Communications, vol. 54, no. 5, pp. 783–788, 2006.

[5] J. Tao, "On low-complexity soft-input soft-output decision-feedback equalizers," IEEE Communications Letters, vol. 20, no. 9, pp. 1737–1740, 2016.

[6] D. George, R. Bowen, and J. Storey, "An adaptive decision feedback equalizer," IEEE Transactions on Communication Technology, vol. 19, no. 3, pp. 281–293, 1971.

[7] S. Sahin, A. M. Cipriano, C. Poulliat, and M.-L. Boucheret, "Iterative decision feedback equalization using online prediction," IEEE Access, vol. 8, pp. 23 638–23 649, 2020.

[8] M. A. Dolatsara, H. Yu, J. A. Hejase, W. Dale Becker, and M. Swaminathan, "Invertible neural networks for inverse design of ctle in high-speed channels," in 2020 IEEE Electrical Design of Advanced Packaging and Systems (EDAPS), 2020, pp. 1–3.

[9] X. Yang, J. Tang, H. M. Torun, W. D. Becker, J. A. Hejase, and M. Swaminathan, "Rx equalization for a high-speed channel based on bayesian active learning using dropout," in 2020 IEEE 29th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), 2020, pp. 1–3.

[10] L. Wu, J. Zhou, H. Jiang, X. Yang, Y. Zhan, and Y. Zhang, "Predicting the characteristics of high-speed serial links based on a deep neural network (dnn)-transformer cascaded model," Electronics, vol. 13, no. 15, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/15/3064

[11] B. Shi, Y. Zhao, H. Ma, T. Nguyen, E. Li, A. C. Cangellaris, and J. E. Schutt-Ainé, "Decision feedback equalizer (dfe) taps estimation with machine learning methods," 2021 IEEE Electrical Design of Advanced Packaging and Systems (EDAPS), pp. 1–3, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:245514140

[12] Y. Hui, Y. Nong, H. Ma, J. Lv, L. Chen, L. Du, and Y. Du, "A cnn-based one-shot blind rx-side-only equalization scheme for high-speed serdes links," in AICAS, 2024, pp. 61–65. [Online]. Available: https://doi.org/10.1109/AICAS59952.2024.10595918

[13] J. Song, B. Peng, C. Häger, H. Wymeersch, and A. Sahai, "Learning physical-layer communication with quantized feedback," IEEE Transactions on Communications, vol. 68, pp. 645–653, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:125953653

[14] J. Song, C. Häger, J. Schröder, A. G. i Amat, and H. Wymeersch, "Model-based end-to-end learning for wdm systems with transceiver hardware impairments," IEEE Journal of Selected Topics in Quantum Electronics, vol. 28, pp. 1–14, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244714321

[15] S. Choi, M. Kim, H. Park, K. Son, S. Kim, J. Kim, J. Park, H. Kim, T. Shin, K. Kim, and J. Kim, "Sequential policy network-based optimal passive equalizer design for an arbitrary channel of high bandwidth memory using advantage actor critic," 2021 IEEE 30th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), pp. 1–3, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244530752

[16] S. Choi, M. Kim, H. Park, H. R. Kim, J. Park, J. Kim, K. Son, S. Kim, K. Kim, D. Lho, J. Yoon, J. Song, K. Kim, J. Park, and J. Kim, "Deep reinforcement learning-based channel-flexible equalization scheme: An application to high bandwidth memory," in DesignCon, 2022.

[17] S. Choi, K. Son, H. Park, S. Kim, B. Sim, J. Kim, J. Park, M. Kim, H. Kim, J. Song, Y. Kim, and J. Kim, "Deep reinforcement learning-based optimal and fast hybrid equalizer design method for high-bandwidth memory (hbm) module," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 13, no. 11, pp. 1804–1816, 2023.

[18] D. Lho, H. Park, K. Kim, S. Kim, B. Sim, K. Son, K. Son, J. Kim, S. Choi, J. Park, H. Kim, K. Kong, and J. Kim, "Deterministic policy gradient-based reinforcement learning for ddr5 memory signaling architecture optimization considering signal integrity," in 2022 IEEE 31st Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), 2022, pp. 1–3.

[19] T. Nguyen, T. Lu, J. Sun, Q. Le, K. We, and J. Schut-Aine, "Transient simulation for high-speed channels with recurrent neural network," in 2018 IEEE 27th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), 2018, pp. 303–305.

[20] M. Usama and D. E. Chang, "Pyeye: An integrated approach for signal integrity assessment and eye diagram generation," in 2023 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2023, pp. 1–3.

[21] R. Medico, D. Spina, D. V. Ginste, D. Deschrijver, and T. Dhaene, "Machine-learning-based error detection and design optimization in signal integrity applications," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 9, pp. 1712–1720, 2019.

[22] Y. Xu, L. Huang, W. Jiang, L. Xue, W. Hu, and L. Yi, "Automatic optimization of volterra equalizer with deep reinforcement learning for intensity-modulated direct-detection optical communications," Journal of Lightwave Technology, vol. 40, pp. 5395–5406, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251325192

[23] L. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579–2605, 2008.

[24] N. Knudde, J. van der Herten, T. Dhaene, and I. Couckuyt, "Gpflowopt: A bayesian optimization library using tensorflow," arXiv: Machine Learning, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:55544345

[25] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," CoRR, vol. abs/1509.02971, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:16326763

Appendix

A. Genetic Algorithm Implementation

Our genetic algorithm baseline optimizes the equalizer parameters using a population-based evolutionary approach. For the DFE configuration, it optimizes four parameters $\mathbf{p} = \{t_1, t_2, t_3, t_4\}$, while for the CTLE+DFE it handles eight parameters $\mathbf{p} = \{G_{dc}, f_z, f_p, G_p, t_1, t_2, t_3, t_4\}$, each within the continuous range [0, 1].

The process starts with an initial population of 25 chromosomes, randomly selected and defined by genes representing the equalizer parameters. The fitness of each chromosome is evaluated based on the window area improvement metric described in Section IV. Chromosomes with higher fitness are probabilistically selected as parents through roulette wheel selection.

For the DFE configuration, single-point crossover is applied at the second gene index, while for CTLE+DFE, two-point crossover is used at indices 3 and 6 to preserve parameter groupings. Controlled mutations, uniformly sampled in range $[-0.1, 0.1]$, are introduced to maintain diversity. Let $\zeta_t \in \mathbb{R}$ denote the best fitness value in the population at generation t. The algorithm terminates when $|\zeta_t - \zeta_{t-1}| \leq 0.0025$ for 10 consecutive generations.

B. Bayesian Optimization [21] Baseline Implementation

Our Bayesian optimization based baseline is based on the optimization method presented in [21]. First, we train an autoencoder network using the process described in Section III. Given an output signal $\mathbf{d}_o$, we use the trained autoencoder to obtain the SI metric score, i.e. $\mu(\mathbf{d}_o)$, where $\mu(\mathbf{d}_o)$ gives the SI metric value of the signal $\mathbf{d}_o$ as described in [21].

For the DFE configuration, the equalizer parameters are $\mathbf{p} = \{t_1, t_2, t_3, t_4\}$. For the cascaded CTLE+DFE configuration, the parameters are $\mathbf{p} = \{G_{dc}, f_z, f_p, G_p, t_1, t_2, t_3, t_4\}$. These parameters $\mathbf{p}$ are sampled, and the signal $\mathbf{d}_o$ is equalized with the sampled parameters to get the equalized signal, $\mathbf{d}_o^e$. The SI metric score for the equalized signal $\mathbf{d}_o^e$ is calculated using the encoder network to get $\mu(\mathbf{d}_o^e)$. The SI metric scores with and without equalization are normalized using the following procedure:

$$l = \frac{\mu(\mathbf{d}_o) - \mu(\mathbf{d}_o^e)}{\mu(\mathbf{d}_o)}.$$

If $l > 0$, the equalization operation with parameters $\mathbf{p}$ has caused an improvement in the signal integrity of the signal $\mathbf{d}_o$. The objective function for the optimization problem is $\max_{\mathbf{p}} l$.

For the implementation of the Bayesian optimization algorithm, we use GPflowOpt [24]. We use Gaussian Process as our surrogate model and run the optimization process for 200 iterations. The autoencoder architecture used for this baseline is the same as our method, shown in Figure 3.

C. Policy Optimization [22] Baseline Implementation

Our policy optimization based baseline utilizes the deep deterministic policy gradient (DDPG) agent [25], which learns equalizer parameters sequentially. The agent interacts with the environment (equalizer model) a number of times equal to the number of parameters to be optimized.

For the 4-tap DFE, the agent determines four parameters $\{t_1, t_2, t_3, t_4\}$ in four sequential interactions. The state $\mathbf{s}_j$ at step $j \in \{1, \ldots, 4\}$ consists of the first j estimated parameters $\{\hat{t}_1, \ldots, \hat{t}_j\}$ padded with zeros for the remaining $4 - j$ parameters. The initial state is $\mathbf{s}_0 = \{0, 0, 0, 0\}$, and the terminal state is $\mathbf{s}_4 = \{\hat{t}_1, \hat{t}_2, \hat{t}_3, \hat{t}_4\}$.

For the cascaded CTLE+DFE, the agent determines eight parameters $\{G_{dc}, f_z, f_p, G_p, t_1, t_2, t_3, t_4\}$ in eight sequential interactions. The state $\mathbf{s}_j$ at step $j \in \{1, \ldots, 8\}$ consists of the first j estimated parameters padded with zeros for the remaining $8 - j$ parameters. The initial state is an 8-dimensional zero vector, and the terminal state contains all eight estimated parameters.

In each interaction, the agent outputs a normalized parameter value in $[0, 1]$, subsequently mapped to the actual parameter range. The actor and critic networks use the same architecture as our proposed A2C method (Fig. 5). The reward $R = 100 \times (1 - \text{BER})$ is calculated after all parameters are determined. A replay memory of 50000 and policy noise $\text{clip}(\mathcal{N}(0, 0.075^2), -0.025, 0.025)$ are used during training.

D. Q-learning Baseline Implementation

Our Q-learning baseline formulates equalizer parameter optimization as a one-step MDP using a Q-network with action branching architecture. The state space comprises latent representations $\ell(\mathbf{d}) \in \mathbb{R}^l$ of the output signal $\mathbf{d} \in \mathbb{R}^{n_x}$. The action space structure varies depending on the equalizer configuration. For the DFE case, the action space is discretized into $k = 16$ levels per tap parameter, resulting in a 4-dimensional discrete space $\mathbb{Z}_k^4$, with actions mapped to tap weights through $f : \mathbb{Z}_k^4 \to [0, 1]^4$. The CTLE+DFE configuration expands this to an 8-dimensional discrete space $\mathbb{Z}_k^8$ with $k = 16$ levels per parameter, mapped through $f : \mathbb{Z}_k^8 \to [0, 1]^8$ to obtain the full parameter set $\{G_{dc}, f_z, f_p, G_p, t_1, t_2, t_3, t_4\}$.

The Q-network architecture adapts accordingly, featuring four heads for DFE and eight heads for CTLE+DFE, where each head outputs k Q-values for its corresponding parameter. Given state $\mathbf{s}_t = \ell(\mathbf{d}_o)$, the agent selects action $\mathbf{a}_t$ using an ε-greedy policy. The reward function $r_t = -\|\ell(\mathbf{d}_i) - \ell(\mathbf{d}^e_{o,\mathbf{p}_t})\|_2$ encourages equalized signals to match the ideal input signal in latent space. Network updates use temporal difference learning with target $q^{\text{target}} = r_t$, shared across all action dimensions as $\mathbf{q}^{\text{target}} = r_t \cdot \mathbf{1}_m$ where m equals 4 for DFE and 8 for CTLE+DFE configurations.

The training process employs a replay memory of size 50000 with batch size 128. An initial exploration rate $\varepsilon = 1.0$ decays by a factor of 0.975 each epoch until reaching a minimum of 0.005. The learning rate begins at 10^{-3} and decreases by 10x every 25 epochs, eventually fixing at 10^{-5}. Training concludes when the standard deviation of average rewards over a 20-epoch window falls below 0.025, indicating convergence of the learned policy.