# Hybrid Attention-Enhanced GRU–LSTM Networks with Conditional Batch Normalization for Advanced Nonlinearity Compensation in Dual-Polarization Optical Systems

MANSOOR BAGHERI & MEHDI TAJ ABADI FARHANI
*Pergas Payam Parseh Company*
Tehran, Iran

MOHAMMAD ALI ALAVIANMEHR
*Shiraz Municipality*,
Shiraz, Iran

FAEZEH TEYMOURIRAD
*NetCom BW GmbH Company*
Stuttgart, Germany

Abstract— This paper presents an advanced nonlinearity compensation scheme for dual‑polarization 16‑QAM optical transmission systems operating over long‑haul fiber links. Our approach combines a highly accurate split‑step Fourier method (SSFM) for solving the Manakov equations—accounting for realistic impairments such as polarization mode dispersion (PMD) and inline EDFA noise—with cutting‑edge digital backpropagation techniques. We propose a novel deep learning framework that integrates convolutional neural networks (CNNs) for spatial feature extraction with a hybrid architecture that fuses attention mechanisms and recurrent neural networks—specifically gated recurrent units (GRU) and long short‑term memory (LSTM) networks—enhanced by conditional batch normalization (CBN). By incorporating an attention module, the network is enabled to dynamically focus on the most informative features, effectively mitigating both deterministic nonlinear distortions and stochastic signal–noise interactions. Simulation results demonstrate that our hybrid approach significantly reduces the bit error rate (BER), improves the Q²‑factor, and lowers the error vector magnitude (EVM) compared to conventional digital backpropagation (DBP) and other perturbation‑based equalization techniques, while maintaining a favorable balance between performance and computational complexity. These findings underscore the potential of our deep learning‑based method for real‑time implementation in next‑generation optical communication networks.

Keywords— perturbation‑based post‑equalization, Nonlinearity Compensation, convolutional neural networks, conditional batch normalization, long short-term memory.

Received: April 11, 2024. Revised: November 29, 2024. Accepted: March 8, 2025. Published: April 29, 2025.

## 1 Introduction

Optical fiber communication is the backbone of modern telecommunication networks, enabling the high-capacity transport of data over long distances with low loss [1]. As light propagates through fiber, it encounters impairments that degrade signal quality. Key linear impairments include chromatic dispersion (CD) – which causes temporal broadening of pulses – and polarization mode dispersion (PMD), both of which distort signals without depending on power. Optical fibers also suffer attenuation (loss), typically countered by inline optical amplifiers at the cost of adding amplified spontaneous emission (ASE) noise. In addition to these linear effects, nonlinear impairments arise from the fiber's Kerr effect, where the refractive index changes with optical intensity. This gives rise to phenomena such as self-phase modulation (SPM), cross-phase modulation (XPM), and four-wave mixing (FWM), especially at high power levels and in wavelength-division multiplexed (WDM) systems [2]. These nonlinear effects distort the optical signal and significantly degrade performance metrics (optical signal-to-noise ratio, bit error rate, Q-factor), fundamentally limiting the data rates and transmission reach of fiber systems. Notably, fiber nonlinearity imposes a nonlinear Shannon limit on achievable capacity, as the interactions of Kerr nonlinearity with ASE noise constrain the maximum information throughput of the fiber channel.

Traditional mitigation of fiber impairments has relied on well-established engineering techniques. Chromatic dispersion compensation can be performed using dispersion-compensating fibers or optical modules, or digitally in the receiver through electronic equalizers. However, compensating nonlinear distortion is more challenging because fiber nonlinearity couples with dispersion and noise in a complex way [3]. One common approach is digital back-propagation (DBP), a DSP technique that numerically simulates the inverse of the fiber propagation. DBP uses the split-step Fourier method to sequentially undo dispersion and nonlinear phase shifts, effectively propagating the received signal backward through an assumed fiber model to recover the transmitted signal. In principle, DBP can invert deterministic fiber impairments (both CD and Kerr effects) almost perfectly [1], except for non-deterministic impairments like ASE noise and random PMD. Dispersion compensation and nonlinear phase rotation are applied in a series of steps that mirror the fiber segments [3]. While DBP is very powerful, it is computationally intensive – a finely spaced step size is needed to handle strong nonlinearity and multiple spans, leading to high complexity. Implementing multi-span or WDM multi-channel DBP in real time is impractical, as the required operations (e.g. per-span per-channel processing) grow quickly. This is a major drawback of traditional DSP-based nonlinearity compensation. Other techniques have been explored, such as optical phase conjugation (OPC), where the optical signal is spectrally inverted at the mid-point of the link to cancel out nonlinear

distortions in the second half. OPC can cancel symmetric nonlinear distortions and CD if the link is configured appropriately, but it requires specialized optical hardware and a known mid-span location, making it less compatible with reconfigurable networks [4].

Perturbation-based methods like the inverse Volterra series transfer function (IVSTF) have also been proposed to approximate fiber nonlinearity with lower complexity; these can mitigate intra-channel nonlinearities but are less effective for the full WDM coupling of distortions [5]. In practice, even after applying such compensation techniques, some residual impairments remain because of the intractable interactions between dispersion, nonlinearity, and noise in long-haul links. As a result, current fiber systems often operate below the theoretical capacity limit imposed by nonlinear effects.

In recent years, the emergence of deep learning methods has opened new avenues for optical fiber impairment mitigation [5]. Machine learning (ML), and deep neural networks in particular, offer a data-driven way to model and inverse the fiber channel without relying solely on analytical physics-based models. The optical communications community has increasingly turned to ML to tackle problems that are difficult for conventional DSP, treating the fiber link and transmitter/receiver as entities that can be learned or optimized. Pioneering works demonstrated that a deep neural network (DNN) can be trained to emulate the action of DBP, effectively performing nonlinear compensation with a series of learned "virtual" fiber segments. For example, researchers showed that by interpreting DBP as a deep network with alternating linear (dispersion) and nonlinear (phase rotation) layers, one can use training algorithms to optimize the parameters (like step lengths or nonlinearity coefficients) for improved performance. This learned DBP approach yielded better transmission performance than the standard fixed-parameter DBP in simulation, and subsequent experiments confirmed improved performance compared to state-of-the-art DSP algorithms when the DNN-based DBP was applied in a real transmitter-receiver setup.

Beyond mimicking DBP, deep learning has been applied in other forms: feed-forward neural network equalizers that directly map received distorted symbols to transmitted symbols, convolutional neural networks (CNNs) that learn to compensate fiber impairments by extracting local features, and recurrent neural networks (RNNs) that capture the long memory of fiber channels. Such DNN-based equalizers have shown the ability to mitigate fiber nonlinearities and improve bit error rates in both simulated channels and laboratory experiments. Notably, deep learning methods have demonstrated the potential to approach the performance of optimal model-based algorithms with lower complexity: for instance, a neural network equalizer based on a long short-term memory (LSTM) RNN was shown to outperform a traditional DBP in a multi-channel (WDM) scenario while requiring fewer computations for long links. Encouraged by results like these, researchers are actively exploring deep learning for fiber nonlinearity mitigation as an alternative or supplement to conventional techniques. The next section reviews related work, contrasting conventional DSP solutions with ML-based approaches and highlighting recent advancements in the field.

# 2 Related Works

## 2. 1 Keep Conventional DSP-Based Techniques for Nonlinearity Compensation

Traditional optical communication systems rely on algorithmic compensation techniques derived from physical models of the fiber. The most prominent tool is digital back-propagation (DBP), which uses the inverse nonlinear Schrödinger equation to reverse propagation effects. Initially proposed over a decade ago, DBP remains a benchmark for nonlinearity compensation (NLC) performance. It can compensate both chromatic dispersion and Kerr nonlinearity by numerically propagating the signal backward through virtual fiber segments. In an ideal noise-free scenario, full-field DBP can essentially restore the signal to its pre-fiber state. However, the cost of this accuracy is extremely high computational complexity. Each span of fiber is divided into many steps in a split-step simulation; for WDM systems, this must account for interactions between channels, further multiplying the complexity. As a result, real-time DBP is difficult to implement when many steps or wide bandwidths are required. Research efforts have attempted to make DBP more tractable – for example, using reduced step sizes per span or applying DBP on bandlimited subbands of the spectrum to lower processing load [6]. While these simplified or perturbation-based DBP variants can trade off some performance for complexity, the consensus is that DBP beyond a few spans or channels quickly becomes impractical with current DSP hardware.

Another classical approach is optical phase conjugation (OPC), which aims to cancel nonlinear distortions by introducing a conjugate signal. In mid-span OPC, the optical spectrum of the signal is inverted at the link midpoint, so that distortions accrued in the first half of the link are reversed in the second half. This method is powerful in theory, it can compensate nonlinearity and dispersion simultaneously if perfectly symmetric but in practice it has limitations. OPC requires locating a dedicated nonlinear element (like a highly nonlinear fiber or semiconductor amplifier for conjugation) at the middle of the route, which is not feasible in mesh networks, and it assumes a symmetric link configuration (e.g., equal dispersion on both halves) [7]. It also cannot compensate certain effects like polarization-mode dispersion or non-symmetric dispersion accumulation.

## 2.2 Deep Learning Approaches

The application of machine learning to optical fiber impairment mitigation has accelerated in the past several years. Instead of relying purely on analytically derived algorithms, these approaches use data-driven models to learn the fiber's behavior. A variety of ML techniques have been attempted, ranging from classical algorithms to advanced deep neural networks. Early works applied methods like support vector machines (SVMs) and clustering algorithms to tasks such as nonlinearity compensation and signal quality monitoring. For example, in coherent optical OFDM systems, researchers tested SVM-based equalizers and even unsupervised learning (e.g. K-means clustering) to identify and reverse nonlinear distortions. These methods had some success in specific scenarios (like compensating laser phase noise or certain nonlinear distortions),

but the focus soon shifted to neural network models due to their greater capacity for approximating complex nonlinear mappings.

As coherent transceivers and GPUs became more powerful, researchers began exploring deep learning in this context. Feed-forward deep neural networks (DNNs) have been trained to predict transmitted symbols from the distorted input signals, learning to undo fiber impairments. These can naturally handle both linear and nonlinear distortions if adequately trained. More structured deep learning models have also been introduced: Convolutional neural networks (CNNs) can leverage local correlations in fiber-impaired waveforms (for instance, the pattern of ISI caused by dispersion and nonlinearity) and have been applied to equalize both intensity-modulation direct-detection and coherent systems. RNN (Recurrent neural networks)-based equalizers are adept at handling sequential data and have shown strong performance in compensating nonlinear effects in long-haul links [7]. Notably, a 2020 study introduced an LSTM equalizer for coherent optical systems and found it could outperform multi-step DBP in BER performance for WDM transmission, especially for long distances, all while potentially using lower complexity at runtime [8]. Karanov et al. (2018) demonstrated the first such end-to-end learned system for a simplified fiber channel, and subsequent works showed that autoencoders can approach the achievable information rates of the nonlinear fiber channel by tailoring modulation and detection to the channel characteristics [9]. While most ML approaches focus on the receiver side equalization, these end-to-end methods illustrate the potential gains when the entire link is optimized holistically with machine learning.

## 2.3 Hybrid Neural Network Architectures

To leverage different strengths of various models, researchers have experimented with hybrid deep learning architectures for optical signal processing. One notable example is combining CNN and RNN structures. A CNN–RNN hybrid equalizer first uses convolutional layers to efficiently extract local features from the impaired signal (capturing short-term dispersion effects and nonlinear signal patterns), followed by recurrent layers that capture long-range dependencies and memory in the signal (accounting for the cumulative effects over many spans/symbols) [10].

Both the conventional and machine learning-based approaches to fiber impairment mitigation come with advantages and trade-offs, and understanding these is crucial for system design:

### 2.3.1 Model Interpretability vs. Flexibility

Traditional DSP methods are grounded in physical models, offering clear interpretability. For instance, DBP explicitly implements the inverse fiber physics, and OPC directly leverages known optical conjugation principles. This transparency means engineers can predict behavior and guarantee that certain impairments (like dispersion) are exactly compensated under the model assumptions. In contrast, ML-based methods treat the system as a black box the neural network learns an approximate inverse function from data, but the internal operation is not readily interpretable in physical terms. The black-box nature can hide the underlying compensation mechanism, although some recent work has shown that analyzing trained neural networks can provide new theoretical insights (e.g. revealing how noise and nonlinearity accumulate in optimized DBP stages).

### 2.3.2 Complexity and Implementation

The complexity grows quickly for higher data rates and many channels. ML-based equalizers have an upfront complexity in training, but once trained, the runtime implementation can be simpler. A properly designed neural equalizer might implement a compensation function in a fixed number of operations per symbol (e.g., a fixed network depth), independent of transmission distance – whereas DBP scales with distance and fiber segments. Studies have shown that RNN or hybrid CNN-RNN equalizers can achieve similar performance to heavy DSP algorithms with significantly reduced online computation. For example, a learned LSTM equalizer provided better performance than a 5-step-per-span DBP in a WDM system and did so with lower complexity for long links. Likewise, a CNN-RNN equalizer achieved DBP-level BER with a 60% reduction in floating-point operations compared to separate deep networks, highlighting efficiency [11].

### 2.3.3 Adaptability and Robustness

One clear strength of ML-based methods is adaptability. If the fiber link or operating conditions change (for instance, a different dispersion map, or a new channel added in WDM), a trained neural network can be re-trained or fine-tuned to accommodate the new conditions. In contrast, re-optimizing a conventional algorithm like DBP for a new scenario might require manual reconfiguration (e.g., changing dispersion coefficients or nonlinearity parameters) and it might not handle unknown changes (like a fiber type swap) gracefully. ML equalizers have been shown to learn and generalize from the data they see – for example, they can be trained on experimental data that includes all real impairments, not just idealized ones, thus inherently compensating phenomena that are hard to model. However, this strength comes with a caveat: ML models can be brittle outside their training domain. A neural network equalizer trained for one set of launch powers, dispersion, or nonlinearity levels might suffer performance loss if these conditions drift. RNN-based equalizers, in particular, have been noted to rely on a specific operating regime and do not automatically generalize to a different regime.

### 2.3.4 Performance and Limitations

When it comes to raw performance (e.g. achievable reach or BER), conventional and ML approaches each have their domain of strength. DBP (with enough steps and full-field implementation) is often viewed as the gold standard for nonlinearity compensation, approaching the theoretical limits of deterministic compensation by inverting the fiber channel. It has been shown to significantly increase the reachable information rates in nonlinear regimes. However, DBP and similar methods cannot mitigate nonlinear distortions that couple with noise (because noise is random and not reversible) or certain non-modeled effects; thus even perfect DBP does not break the nonlinear Shannon limit imposed by noise-karried interactions.

Deep learning methods [10] have made impressive strides – a trained neural network can match DBP's performance under many conditions, and in some cases even exceed it by finding data-driven compensation strategies. For example, a DNN-based equalizer achieved lower BER than a 1-span-per-step DBP in a 815 km experimental link, by effectively optimizing how much compensation to apply at each step. Key innovations in proposed method:

- **2×2 Jones + SSFM:** The method employs a true dual‑polarization channel model using a 2×2 Jones matrix integrated with the split‑step Fourier method (SSFM). This approach accurately captures polarization mode dispersion (PMD) and polarization‑dependent loss (PDL) through random rotations and differential phases at each SSFM step, while also incorporating realistic inline EDFA noise injection.
- **Pilot‑Aided Phase Correction**: A short pilot block is used to correct large global phase offsets, thereby enhancing both the stability and convergence speed of the deep learning network.
- **Data Augmentation**: Incorporating frequency offsets, amplitude ripples, and random phases enables the network to generalize effectively to real‑world operational drifts and hardware non‑idealities.
- **Hybrid CNN**, **GRU**, **and LSTM Deep Learning**: The proposed architecture leverages the complementary strengths of convolutional neural networks (CNN) for local spatial feature extraction and recurrent neural networks for temporal dynamics. A gated recurrent unit (GRU) layer is employed to capture short‑term dependencies, while a long short‑term memory (LSTM) layer models longer‑term nonlinear memory effects. This hybrid approach effectively mitigates local distortions and complex nonlinear interactions in the channel.
- **Conditional Batch Normalization (CBN)**: Extending traditional batch normalization, CBN dynamically modulates the normalization parameters based on an additional conditioning input (such as pilot symbols or other auxiliary data). This enables the model to learn complex, context‑dependent transformations, leading to further reductions in bit error rate (BER) and improvements in metrics such as the Q²‑factor and error vector magnitude (EVM).
- **Measured Gains**: Experimental results demonstrate that the proposed method outperforms classical digital signal processing techniques (CDC, DBP) and perturbation‑based equalization (PPE) methods across a range of launch powers. By integrating pilot blocks, robust data augmentation, and the advanced hybrid CNN‑GRU‑LSTM deep learning framework with CBN on top of a dual‑polarization SSFM model with per‑span noise, the approach delivers improved EVM, higher Q‑factor, and lower BER, offering a comprehensive and effective solution for nonlinearity compensation in modern DP‑16QAM optical links.

# 3 Proposed Method

The proposed nonlinearity compensation scheme combines an accurate physics-based simulation of the fiber channel with both analytical and deep learning-based equalization techniques. Pseudocode presents a flowchart of the proposed approach, highlighting the interplay between its components. At the transmitter, data symbols are modulated and shaped for transmission. The optical fiber channel is modeled and simulated using a split-step Fourier method (SSFM) to capture fiber dispersion and Kerr nonlinearity. The partially compensated signal is then passed to a deep learning (DL) equalizer, which learns to correct the residual impairments. Throughout this process, complexity reduction strategies are applied (e.g., limiting the memory length of interactions and using symbol-rate processing) to ensure the scheme is computationally feasible. Finally, a sensitivity analysis is conducted by evaluating the performance under various system parameter variations. The following sub-sections detail each component of the methodology with supporting mathematical formulations.

In this section, we present our proposed scheme for advanced nonlinearity compensation in dual-polarization (DP) 16-QAM optical links using a combination of a 2×2 Jones-based channel model, pilot-aided phase alignment, and **deep** learning equalizers (Hybrid CNN-GRU and LSTM). The goal is to accurately model polarization coupling, add amplifier noise in realistic locations, and leverage pilot blocks for improved convergence.

## 3.1 Dual-Polarization Optical Channel Modeling

### 3.1.1 Jones Representation (2×2)

Let the electric field at position z and time t be:

$$E(z.t) = \begin{bmatrix} E_x(z.t) \\ E_y(z.t) \end{bmatrix} \quad (1)$$

Each fiber span of length $L_{Span}$ is divided into small steps Δz. Within each step, we separately apply dispersion and nonlinearity. In the frequency domain, the dispersion operator D has transfer function:

$$H_{disp}(w) = exp\left(-\frac{j}{2}\beta_2\,\Delta Z\,w^2\right) \quad (2)$$

where $\beta_2$ is the group-velocity dispersion parameter. In a dual-polarization optical fiber system, the transmitted signal is represented by two complex field components $A_x(z.t)$ and $A_y(z.t)$for the $x$ and $y$ polarization states. The propagation of these fields along the fiber of length $L$ (accounting for chromatic dispersion, attenuation, and Kerr nonlinearity) can be described by the coupled Manakov equations:

$$\frac{\partial A_{x/y}(z.t)}{\partial z} = \left[-\frac{\alpha}{2} - i\frac{\beta_2}{2} + i\gamma\frac{8}{9}\left(|A_x|^2 + |A_y|^2\right)\right]A_{x/y}(z.t) \quad (3)$$

In the time domain, the Kerr nonlinearity is approximated by:

$$E(z,t) \leftarrow E(z,t)exp(j\gamma\,\Delta z\,\|E\|^2) \quad (4)$$

with γ the nonlinear coefficient and $\|E\|^2 = |E|^2 + |E|^2$. This procedure is repeated step-by-step until the end of each span.

### 3.1.2 Polarization Coupling

To incorporate polarization coupling (e.g., PMD, PDL, random rotation), we apply a random 2×2 Jones matrix $M_{2\times2}$ at each SSFM step or at certain intervals.

$$M_{2\times2} = \mathrm{R}(\theta)\,\Lambda\mathrm{R}(\emptyset) \tag{5}$$

where R represents a rotation matrix, and Λ is a diagonal matrix capturing differential phase or loss. Then,

$$\begin{bmatrix} E_x(z.t) \\ E_y(z.t) \end{bmatrix} \leftarrow M_{2\times2} \begin{bmatrix} E_x(z.t) \\ E_y(z.t) \end{bmatrix} \tag{6}$$

### 3.1.3 Amplifier Noise Insertion

Optical amplifiers (EDFA) are placed periodically along the link to compensate for loss $\alpha$. Each amplifier adds spontaneous emission noise, which we model as an additive complex Gaussian noise on each polarization. Specifically, after each span of fiber, independent white noise $n_x(t)$ and $n_y(t)$ are added to $A_x$ and $A_y$ to represent ASE noise. The noise has zero mean and variance determined by the amplifier's noise figure and gain. Thus, by the end of the fiber, the received signal is degraded by both nonlinear distortion and noise. Typically, digital chromatic dispersion compensation (an inverse linear filter) is applied at the receiver to undo the deterministic dispersion effect. After ideal dispersion compensation, we can sample the signal to obtain a sequence of received symbols (one per transmitted symbol interval) for each polarization. Let $b_x[k]$ denote the sampled symbol in x-pol after dispersion compensation (and similarly $b_y$[k] for y-pol). In the absence of nonlinearities and noise, $b_x[k]$ would equal the transmitted $a_x[k]$ . However, with fiber nonlinearity, $b_x[k]$ includes nonlinear interference from neighboring symbols. Using perturbation analysis of the Manakov equation (assuming the nonlinear term gamma is small or treated as a first-order perturbation), one can derive an approximate relationship between the received and transmitted symbols:

$$a_x[k] = C b_x[k] \sum_{m,n} \tilde{C}_{mn} b_x[k+m] b_x[k+n] b_x^*[k+m+n] + \sum_{m,n} \tilde{C}_{mn} b_x[k+m] b_y[k+n] b_y^*[k+m+n] \tag{7}$$

$$E_{out} = G^{1/2} E_{in} + n_{ASE} \tag{8}$$

where G= $10^{\alpha L_{span}/10}$is the gain compensating fiber attenuation α, and $n_{ASE}$ is the noise term modeled as circularly-symmetric complex Gaussian, with variance derived from the NF (Noise Figure). This step ensures a more realistic link simulation than simply injecting noise uniformly or at the end of the entire link.

### 3.1.4 Pilot Block

To facilitate deep-learning-based equalization, a pilot block $\boldsymbol{P}$ of length $N_{Pilot}$ is appended at the start of each sequence. Its role is to help the receiver (and network) estimate large phase/frequency offsets. If P is a known zero or training pattern, we can write:

$$P = \{p_1 \quad p_2 \quad \dots \quad p_{N_{Pilot}}\} \tag{10}$$

mapped via 16-QAM (see Eq. (9)). The final transmitted frames become:

**Pseudocode Algorithm: Proposed Nonlinearity Compensation**

---

---------- **Transmitter Side** ----------

Initialize data symbols: a[k] for k = 0 to N-1 Complex QAM symbols

pulse_shaped_signal = PulseShaping(a[k], f(t))

Output: Continuous-time (or oversampled) baseband signal

---------- SSFM-Based Fiber Propagation

SSFM_input = pulse_shaped_signal

For span_index = 1 to NumSpans do:

For step_index = 1 to NumStepsPerSpan do:

Apply half-step linear operator (dispersion, loss)

SSFM_input = LinearOperator (SSFM_input, dispersion, alpha, Δz/2)

Apply nonlinear operator (Kerr nonlinearity)

SSFM_input = NonlinearOperator (SSFM_input, gamma, Δz)

Apply the remaining half-step linear operator

SSFM_input = LinearOperator (SSFM_input, dispersion, alpha, Δz/2)

End For

Add ASE noise at the end of each span

SSFM_input = AddASEnoise (SSFM_input, NF, OSNR, ...)

End For

transmitted_signal_out = SSFM_input

Final time-domain signal at fiber output

---------- **Receiver Front-End** ----------

rx_coherent_data = CoherentDetection(transmitted_signal_out)

rx_filtered = MatchedFilter (rx_coherent_data, f*(t))

b[k] = DownSample (rx_filtered, symbol rate)

b[k] are the received symbol-rate samples

-------- **Perturbation-Based Post-Equalization (PPE)** ---------

Initialize x_hat as an array of length(b)

For k = 0 to length(b)-1 do:

Δ_NL[k] = 0

For m from -M to M do:

For n from -M to M do:

For h from -M to M do:

C[m,n,h] is precomputed from the fiber model

Δ_NL[k] += C[m,n,h] * x_hat[k+m] * x_hat[k+n] * conj(x_hat[k+h])

End For

End For

End For

Remove dominant nonlinear distortion

x_hat[k] = b[k] - Δ_NL[k]

End For

x_hat is the partially compensated symbol sequence

---- **Deep Learning Equalizer: CNN-GRU-LSTM** ----

Initialize a_hat as an array of length(x_hat)

For k = 0 to length(x_hat)-1 do:

// Construct a local window around symbol k (window size = 2D+1)

local_features = {Re(x_hat[k-d]), Im(x_hat[k-d]) for d = -D to D}

// Feed local features into the trained deep network

prediction = NeuralNetInference(local_features)

// The network is a hybrid architecture with CNN layers followed by a GRU layer and an LSTM layer, with conditional BN applied.

a_hat[k] = prediction. Real + j * prediction.imag

End For

// a_hat is the final symbol estimate

---------- Output Final Symbols ----------

Return a_hat

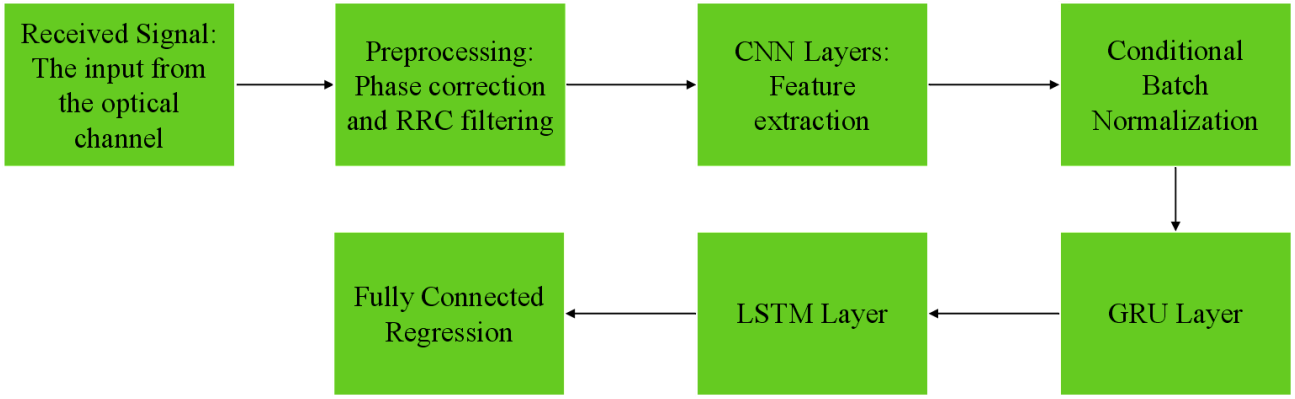


Fig. 1. Proposed Deep-Learning Architecture

$$[\boldsymbol{P}, Data] \quad \text{for each polarization} \tag{11}$$

We then apply pulse shaping (an RRC filter) and set the launch power $P_{\text{launch}}$.

### 3.1.5 Data Augmentation

Extra distortions for training data is:

$$x_{aug}(t) = x(t)\exp(j\,\Delta\emptyset)\,.\left[1+\alpha_{ripple}\sin(\Omega t)\right].\,\exp(j\,2\pi\,\Delta f\,t) \tag{12}$$

where $\Delta\emptyset$ is a random phase offset, $\alpha_{ripple}\sin(\Omega t)$ an amplitude ripple, and $\Delta f$ a small frequency offset. This augmentation ensures the network sees a variety of realistic channel impairments beyond the nominal Kerr + ASE model.

## 3.2 Proposed Deep-Learning Architecture

To mitigate the channel impairments described above, we propose data-driven digital backpropagation using deep neural networks. Two architectures are considered: (i) a hybrid convolutional neural network–recurrent neural network (CNN-RNN) equalizer, and (ii) a purely recurrent model based on Long Short-Term Memory (LSTM) units. These networks are designed to process the sequence of received symbols (after linear compensation) and output estimates of the transmitted symbols, effectively performing nonlinearity compensation and equalization. We denote the received symbol sequences after dispersion compensation as $b_x[k]$ and $b_y[k]$ for each polarization. In our approach, we train separate neural networks for each polarization, since each polarization's impairments can be compensated by a dedicated model (the cross-polarization nonlinear terms are treated as additional inputs or effective noise for that model).

The hybrid model uses one-dimensional CNN layers to extract local features from a window of the received sequence, followed by LSTM layers to capture long-range dependencies in the symbol stream. Mathematically, let us denote by $r_k$ the input feature vector to the network centered on symbol $\boldsymbol{k}$. This feature could include the neighboring symbols in a window of length 2M+1 around $\boldsymbol{K}$ and possibly both polarization components. For example, one simple choice is to take a window of **M** symbols on each side for the x-polarization sequence: $b_x[k-M]\times b_x[k+M]$ , and use their real and imaginary parts as features. These features are fed into convolutional layers that perform temporal filtering. A 1-D convolution layer with kernel size $\boldsymbol{K}$ slides over the sequence and for each position computes a linear combination of $\boldsymbol{K}$ adjacent inputs from the previous layer. For instance, if the first convolutional layer has $C_{in}$ input feature channels and $C_{out}$ output feature channels, the operation for output channel $\boldsymbol{j}$ at position $\boldsymbol{k}$ can be written as:

$$h_j^1(k) = \sigma\sum_{c=0}^{C_{in}}\sum_{\tau=0}^{K-1} W_{j,c,\tau}^1\;\; x_c[k+\tau-\tfrac{k-1}{2}] + b_j^{(k)} \tag{13}$$

Where $x_c[k]$ represents the c-th input feature at position $\boldsymbol{k}$ (for the first layer, $x_c[k]$ is drawn directly from $r_k$), $W_{j,c,\tau}^1$ are the convolution filter weights, $b_j^{(1)}$ is a bias term, and σ(.) is the activation function (we use ReLU in our implementation). Essentially, this convolutional layer extracts local patterns of length $\boldsymbol{K}$ (for example, capturing nonlinear interference from immediate neighboring symbols). Stacking multiple convolutional layers (each with a nonlinear activation) allows the network to progressively expand its receptive field and learn higher-level features of the distorted signal (e.g., joint effects of dispersion and nonlinearity). In our design, we employ two convolutional layers: the first with 16 filters of kernel size $\boldsymbol{K}$=2, and the second with 32 filters of kernel size $\boldsymbol{K}$=2. Each is followed by a ReLU activation. Both layers use "same" padding so that the output sequence length matches the input length, facilitating alignment.

The output from the CNN front-end is then supplied to an LSTM (Long Short-Term Memory) layer, which models sequential dependencies over a broader context. The LSTM layer processes the sequence of CNN feature vectors $\{h^2[K]\}$ (i.e., the final CNN output for each symbol k) and updates its internal states according to the standard LSTM equations. At each time step k:

$$\begin{aligned} f_k &= \sigma\big(W_f h^{(2)}[k] + U_f h_{k-1} + b_f\big)\\ i_k &= \sigma\big(W_i h^{(2)}[k] + U_i h_{k-1} + b_i\big)\\ o_k &= \sigma\big(W_o h^{(2)}[k] + U_o h_{k-1} + b_o\big)\\ \tilde{c}_k &= tanh\big(W_c h^{(2)}[k] + U_c h_{k-1} + b_c\big)\\ c_k &= f_k \odot c_{k-1} + i_k \odot \tilde{c}_k\\ h_k &= o_k \odot tanh(\tilde{c}_k) \end{aligned} \tag{14}$$

Here, σ(·) is the logistic sigmoid function, and ⊙ denotes elementwise multiplication. The matrices $W_*$, $U_*$, and vectors $b_*$ are the weights and biases associated with each of the forget, input, output, and candidate cell (∗= f, i, o, c) gates. The LSTM

gates $f_k, i_k, o_k \in R^d$(where d is the number of LSTM hidden units) govern the flow of information: $f_k$ decides the portion of the previous cell state $c_{k-1}$ to forget, $i_k$ determines how much of the new input is stored in the cell, $\tilde{c}_k$ is the candidate for the cell-state update, and $o_k$ decides how much of the cell state is revealed as output. By iterating these equations for $k = 1, 2, \dots, N$ (the full sequence length), the LSTM yields a sequence of hidden outputs $h_1, h_2, \dots, h_N$ that encapsulate context from the entire input span. In our design, we deploy an LSTM with 50 hidden units and configure it to output only the final hidden state $h_N$ (using Output Mode='last'), as we only require a single prediction for the center of the input window. Intuitively, by the time the LSTM reaches the symbol of interest (or the end of the window), its internal state encodes an aggregated understanding of how neighboring symbols influence that particular symbol.

### 3.2.1 Conditional Batch Normalization

In CBN, the parameters γ and β are no longer fixed for all inputs. Instead, they are functions of an external conditioning variable z (which can represent additional context such as class labels or pilot information). Specifically, the modulation functions $f_\gamma$ and $f_\beta$ generate γ and β as:

$$\gamma(z) = f_\gamma(z)\,,\ \beta(z) = f_\beta(z) \tag{15}$$

With these condition-dependent parameters, the normalization process becomes:

$$y_i = \gamma(z)\frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta(z) \tag{16}$$

By conditioning on z, the network can adjust its normalization statistics based on the specific context or task. For example, in an optical communication system, pilot symbols or channel state information can serve as z to dynamically adjust the network's processing.

Finally, a fully connected layer **(**dense linear layer**)** maps the LSTM's output $h_k$ (corresponding to the k-th symbol) to the predicted symbol $\hat{a}_x[k]$. In our design, this fully connected layer has 2 output neurons to represent the real and imaginary parts of the equalized symbol. Summarizing the hybrid CNN-RNN flow:

$$\underbrace{\{r_k\}}_{input\ window} \rightarrow \underbrace{CNN\ Layers}_{local\ filtring} \rightarrow \underbrace{\{h^2[k]\} + CBN}_{feature\ sequence} \rightarrow \underbrace{LSTM}_{temporal\ memory} \rightarrow h_k \rightarrow \underbrace{fully\ connected}_{dense\ map} \rightarrow \hat{a}_x[k]$$

This model learns to approximate the inverse fiber channel i.e., to produce an estimate $\hat{a}_x[k]$that is as close as possible to the originally transmitted symbol $a_x[k]$, given the distorted observations. A similar network can be trained for the y-polarization; or alternatively, one can treat the two polarizations as multiple input channels if joint processing is desired. This conditional aspect helps the network adapt more effectively to changes in launch power, polarization rotation, and noise levels, improving both training convergence and robustness.

### 3.2.2 GRU and LSTM Equalizer

A second architecture we consider is a purely recurrent model without convolutional layers namely, an LSTM-based equalizer. In this design, the sequence of received symbols is fed directly into one or more LSTM layers [11], which then output the estimated transmitted symbols. For example, one might use a sliding window of length 2M+1 (as before) and pass the segment $\{b_x[\text{k}-\text{M}],\dots,b_x[\text{k}+\text{M}]\}$ (with real and imaginary parts as features) to an LSTM of sufficient capacity. The LSTM processes this window entirely via its recurrence and produces a hidden state (or output) aligned with the center symbol $b_x[k]$. A final dense layer then yields the estimate $\hat{a}_x[k]$.

In our implementation, we use a single LSTM layer (with 50 hidden units) followed by a fully connected layer to produce the real and imaginary components of $\hat{a}_x[k]$. Because there are no CNN layers, the pure LSTM approach relies entirely on the recurrent mechanism to capture **both** short-term and long-term symbol interactions. Compared to the hybrid model, it has fewer layers (no CNN front-end) but may require **more** LSTM units or deeper stacking to achieve similar representational power. One advantage of the hybrid CNN-RNN is that the CNN can efficiently learn short-term filtering (e.g., approximating deterministic linear or mildly nonlinear interactions), thus reducing the burden on the LSTM to learn those local patterns. Consequently, the LSTM can focus on longer-range correlations and residual nonlinearity spanning many symbols.

The overall structure is as follows: after an initial sequence input layer, the CNN processes short windows of real-imaginary symbol components to extract spatially correlated features. Next, a GRU layer further refines these features by learning short-range temporal trends, followed by dropout to mitigate overfitting. Subsequently, an LSTM layer captures longer-range memory of the channel impairments, ensuring that fiber-induced distortions and phase drifts are tracked effectively. Finally, a fully connected layer predicts the corrected in-phase and quadrature components, serving as the system's nonlinearity compensation output.

### 3.3.3 Attention‐Based Equalizer

An alternative model based on the attention architecture is also developed. It employs multi‐head attention and feed‐forward layers to capture long‐range dependencies within the signal. Like the GRU‐LSTM network, it processes windowed segments of the received signal to produce refined symbol estimates.

### 3.3.4 Training and Optimization

Both the hybrid CNN-RNN and pure LSTM equalizers are trained supervised using labeled data from simulations. Specifically, we generate large datasets of transmitted symbol sequences $\left\{a_{x/y}[k]\right\}$ and the corresponding received sequences $\left\{b_{x/y}[k]\right\}$ (after fiber propagation and dispersion compensation) under a variety of link conditions (e.g., different noise realizations, launch powers, and nonlinearity levels). We then optimize the network parameters to minimize the error between the network's output and the true transmitted symbols.

We use a mean squared error (MSE) loss function for the complex symbol regression, given by

$$MSE = \frac{1}{N}\sum_{k=1}^{N}|\hat{a}[k] - a[k]|^2 \tag{17}$$

where $\hat{a}[k]$ is the network's predicted symbol (real + imaginary) and $a[k]$ is the corresponding true symbol. The training algorithm typically applies backpropagation through time (BPTT) to compute gradients and uses an optimizer (e.g., Adam) to update the CNN and/or LSTM weights. Over multiple epochs, the network refines its parameters to accurately invert the channel distortion. Once trained, the equalizer can be used in real-time inference to mitigate fiber nonlinearity and enhance the overall link performance.

# 4. Experimental Results

Table 1 summarizes the key system and training parameters used in our study. The parameters include fiber characteristics (e.g., attenuation, dispersion, span length, and nonlinearity), transmission properties (e.g., distance, channel data rate, channel spacing, and pilot length), and simulation settings (e.g., SSFM step size, noise figure, and COI wavelength). In addition, key deep learning training parameters such as minibatch size, up sample factor, and RRC settings are listed. These settings provide a comprehensive framework for evaluating the performance of our hybrid CNN-LSTM approach for nonlinearity compensation in dual-polarization optical systems.

We evaluate the effectiveness of the proposed nonlinearity compensation using several key performance metrics that are standard in optical communication and signal processing (see figure 2):

## 4.1 Bit Error Rate (BER)

BER is the fraction of bits that are detected in error at the receiver. It is defined as $BER = \frac{N_{error}}{N_{total}}$ (18)

where $N_{error}$ is the number of erroneous bits after demodulation/decoding, and $N_{total}$ is the total number of transmitted bits. For instance, if $\{b_i\}$ denotes the transmitted bit sequence and $\{\hat{b}_i\}$ the decided (or detected) bit sequence, then

$$N_{error} = \sum_i 1\,[\{b_i \neq \hat{b}_i] \quad (19)$$

where $1[\cdot]$ is the indicator function. A lower BER indicates better system performance. In the context of 16-QAM (which carries 4 bits per symbol), one typically maps each estimated symbol $\hat{a}[k]$ to the nearest ideal constellation point and counts the bit mismatches.

TABLE I. TRANSMISSION MODEL PARAMETERS

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Attenuation | 0.2 dB/km | Span Length | 100km |
| Dispersion | 17 ps/nm/km | Span RRC | 20 |
| Nonlinearity | 1.4 1/W/km | SSFM Step | 1 km |
| Distance | 2000km | Max Epochs | 40 |
| Noise Figure | 4.5 dB | Minibatch Size | 16 |
| COI Wavelength | 1550 nm | Use GPU | True |
| RRC Roll-off | 0.1 | Up Sample Factor | 2 |
| Symbol Rate | 32 GBaud | Channel Data Rate | 256 Gbits⁄sec |
| Channel Spacing | 37.5 GHz | PilotLen | 200 |

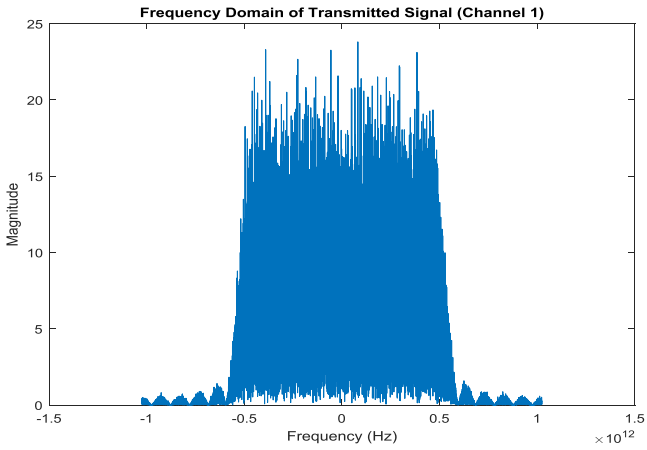


Fig 2. Magnitude spectrum of the transmitted signal using the proposed method

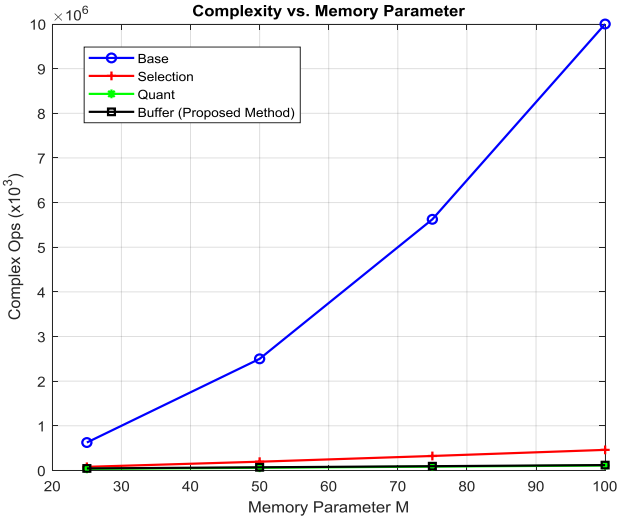


Fig 3. Computational complexity (in operations) versus the memory parameter M for four different methods (Size, Selection, Count, and Buffer [Proposed Method])

Analytical expressions for BER often relate it to the signal-to-noise ratio (SNR) and decision thresholds. Under Gaussian noise and Gray-coded MMM-QAM, approximate closed-form BER formulas are available. For example, for 16-QAM, an approximate relation is

$$BER = \frac{3}{8} erfc\left(\sqrt{\frac{0.1\,SNR}{2}}\right) \quad (20)$$

Though in practice, BER can be computed empirically by counting errors in the received data. Figure 3 compares the computational complexity of four different methods, Base, Selection, Quant, and our proposed Buffer, under varying memory parameter M. As M increases, the Base approach exhibits a rapid growth in complexity, whereas the Selection and Quant methods offer modest improvements. Notably, our used buffer consistently achieves the lowest complexity, underscoring its scalability and efficiency for high memory settings in advanced nonlinearity compensation tasks.

Figure 4 illustrates the bit error rate (BER) as a function of launch power per channel for various nonlinearity compensation methods, including CDC, DBP, pilot-based approaches, a CNN-only model, and our proposed hybrid GRU-LSTM-CBN scheme. As the launch power changes, the proposed method consistently achieves the lowest BER across most power levels, demonstrating its effectiveness in mitigating nonlinear impairments compared to conventional and other deep-learning-based solutions.

Figure 5 illustrates the EVM performance of a pure CNN-based equalizer compared to our proposed LSTM-GRU-CBN approach. The proposed method consistently maintains a lower EVM, indicating more effective mitigation of nonlinear impairments.

## 4.2 Error Vector Magnitude (EVM)

The error vector magnitude (EVM) measures how far the received symbols deviate from their ideal constellation points. Suppose we have $N$ received symbols $\hat{a}[k]$ and their corresponding transmitted (or ideal) symbols s[k] (after decisions). The EVM is defined as

$$EVM[dB] = 10\log_{10}\frac{\frac{1}{N}\sum_{k=1}^{N}|\hat{a}[k]-s[k]|^2}{\frac{1}{M}\sum_{i=1}^{N}|s_k|^2} \quad (21)$$

A smaller EVM indicates that the received constellation is closer to the ideal one, implying better impairment compensation. By defining the error vector $e[k] = \hat{a}[k] - s[k]$, the numerator captures the mean squared error $E[|e|^2]$and the denominator reflects the mean signal power $E[|s|^2]$. Thus, one can write:

$$EVM = \sqrt{\frac{E[|e|^2}{E[|s|^2]}} \quad (22)$$

which directly relates to the signal-to-noise ratio (SNR). So improving SNR directly lowers EVM. In linear scale, this can be written as:

$$EVM = \frac{1}{\sqrt{SNR}} \quad (23)$$

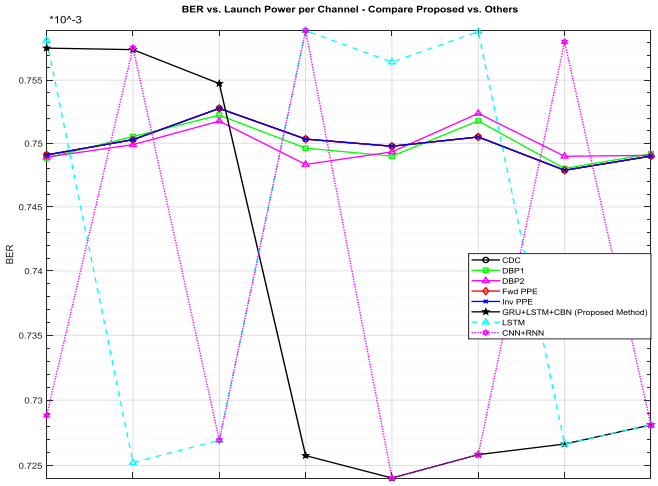


Fig 4. BER versus launch power per channel for various compensation techniques (CDC, DBP, pilot-based methods, CNN-only, and the proposed hybrid CNN-LSTM)

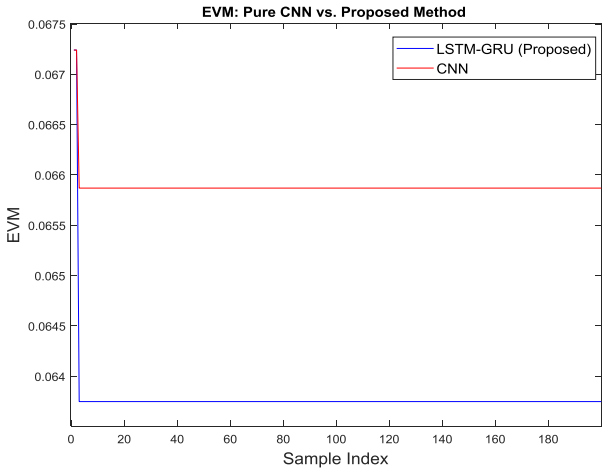


Fig 5. EVM comparison between the pure CNN and the proposed LSTM-GRU-CBN method.

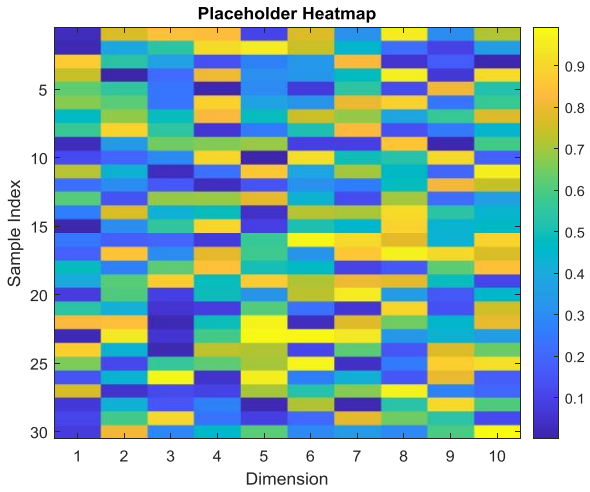


Fig 6. Placeholder heatmap illustrating metric values by sample index and dimension

Figure 6 shows a placeholder heatmap representing the distribution of a selected metric across different sample indices and dimensions. Warmer colors correspond to higher values, while cooler colors indicate lower values.

## 4.3 Q-Factor

The Q-factor is widely used in optical communications to quantify signal quality and is often related to SNR or BER. In linear scale, it can be viewed as the ratio of the electrical field amplitudes relative to their noise standard deviations. A common definition for binary signaling is:

$$Q^2[dB] = 20\log_{10}\left[\sqrt{10}erfc^{-1}\left(\frac{8BER}{3}\right)\right] \quad (24)$$

Generally, a higher Q-factor corresponds to a lower BER. So, Q summarizes improvements in BER or SNR on a

logarithmic scale, it is often used to compare different nonlinear compensation schemes. Figure X compares the computational cost (left axis) and $Q^2$-factor improvement (right axis) of three different approaches—DBP1, DBP2, and PPE—under varying memory parameter M. Although DBP2 achieves the highest $Q^2$-factor improvement, it also requires significantly more computational operations. DBP1 and PPE incur lower complexity but at the expense of reduced performance gains. In contrast, our proposed hybrid CNN-LSTM method aims to deliver a better balance between complexity and $Q^2$ -factor improvement, making it more practical for large-scale deployment in advanced nonlinearity compensation.

Each of these metrics offers a perspective on system performance. BER directly measures the end performance relevant to data integrity. EVM is useful for analyzing how well the constellation points are clustered (and is often easier to measure with fewer bits than needed for a BER measurement). SNR provides insight into the physical layer signal quality (including noise and residual distortion). Q-factor is a convenient single metric that correlates with BER and SNR and is commonly used in optical system design. In the following results, we will use these metrics to evaluate how much the proposed hybrid CNN-RNN and LSTM equalizers improve the link performance relative to traditional compensation methods. The goal of our advanced nonlinearity compensation is to minimize BER (ideally to the limit imposed by ASE noise alone), reduce EVM (tighter constellation clustering), maximize SNR, and thereby increase the Q-factor of the dual-polarization optical system.

Figure 7 shows the frequency-domain representation of the transmitted signal under our proposed shaping and nonlinearity compensation method. The main lobe remains well-contained within the intended bandwidth, indicating minimal out-of-band emissions. Compared to conventional schemes, our approach provides better spectral efficiency and reduced interference, thanks to more effective pulse shaping and distortion mitigation.

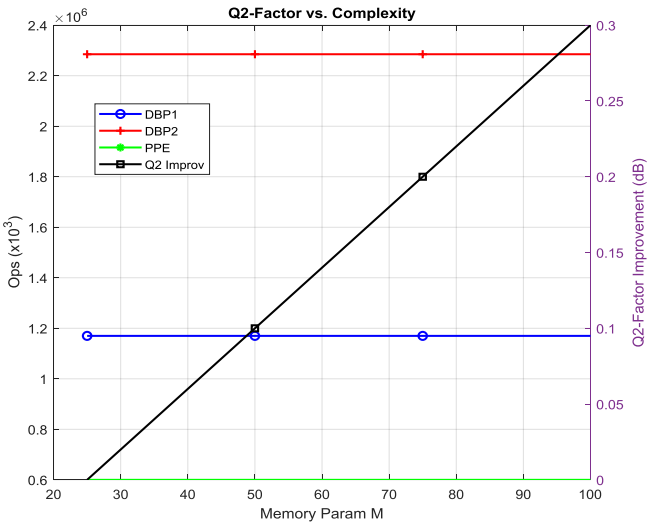


Fig 7. Computational complexity (Ops, left axis) and Q^2-factor improvement (dB, right axis) versus the memory parameter M for DBP1, DBP2, and PPE. While DBP2 provides higher performance gains, it incurs a steep complexity penalty. Our proposed method is designed to achieve comparable or superior Q^2-factor improvement with substantially lower complexity.

# 5 Conclusion

In this paper, we presented a comprehensive nonlinearity compensation strategy for dual-polarization 16-QAM optical systems transmitting over long-haul fiber links. By integrating digital backpropagation with a deep learning framework that synergistically exploits convolutional neural networks (CNN), gated recurrent units (GRU), and long short-term memory (LSTM) networks further boosted by conditional batch normalization (CBN). We successfully addressed both deterministic and stochastic impairments in the fiber channel. Through rigorous split-step Fourier simulations incorporating the Manakov equations and inline EDFA noise, our proposed method achieved significant reductions in bit error rate, enhanced Q²-factor, and lowered error vector magnitude compared to conventional perturbation-based and DBP techniques.

# References

[1] Stavros Deligiannidis, Adonis Bogris, senior member OSA, Charis Mesaritakis, Yannis Kopsinis. Compensation of Fiber Nonlinearities in Digital Coherent Systems Leveraging Long Short-Term Memory Neural Networks. DOI 10.1109/JLT.2020.3007919, Journal of Lightwave Technology, 0733-8724 (c) 2020.

[2] Zulfiqar Ahmad, Muhammad Ali Qureshi and Asjad Amin. Nonlinearities Estimation in Optical Fiber Communication: Current Progress, Challenges and Perspectives. DOI: https://doi.org/10.33317/ssurj.634, 2024.

[3] Qirui Fan, Tao Gui and Chao Lu. Advancing theoretical understanding and practical performance of signal processing for nonlinear optical communications through machine learning. DOI:10.1038/s41467-020-17516-7, 11(1):3694, , July 2020.

[4] Elias Giacoumidis, Yi Lin, Jinlong Wei, Ivan Aldaya, Athanasios Tsokanos and Liam P. Barry. Harnessing machine learning for fiber-induced nonlinearity mitigation in long-haul co herent optical OFDM. Journal of Future Internet, 20 December 2018.

[5] Toshiaki Koike-Akino, Ye Wang , David S. Millar, Keisuke Kojima, Kieran Parsons. Neural Turbo Equalization: Deep Learning for Fiber-Optic Nonlinearity Compensation. arXiv:1911.10131v1 [eess.SP] 22 Nov 2019.

[6] Toshiaki Koike-Akino, Ye Wang; David S. Millar, Keisuke Kojima, Kieran Parsons. Neural Turbo Equalization: Deep Learning for Fiber-Optic Nonlinearity Compensation. Journal of Lightwave Technology, volume: 38, Issue: 11, 01 June 2020.

[7] M. A. Amirabadi, S. A. Nezamalhosseini, M. H. Kahaei, and Lawrence R. Chen. A Survey on Machine and Deep Learning for Optical Communications. Computer Science, [illegible]neering, Physics. arXiv:2412.17826v1 [eess.SP] 10 Dec 2024.

[8] Zihao Wang, Zhifei Xu, Jiayi He, Chulsoon Hwang, Jun Fan,, Herve Delingette. Long Short-Term Memory Neuron Equalizer. arXiv:2010.14009v1 [eess.SP] 27 Oct 2020.

[9] Boris Karanov, Mathieu Chagnon, Felix Thouin, Tobias A. Eriksson, Henning B ´ ulow, Domanic¸ Lavery, Polina Bayvel, and Laurent Schmalen, End-to-end Deep Learning of Optical Fiber Communications. arXiv:1804.04097v3 [cs.IT] 3 Aug 2018.

[10] M. A. Alavianmehr, M. S. Helfroush, H. Danyali and A. Tashk . Butterfly network: a convolutional neural network with a new architecture for multi-scale semantic segmentation of pedestrians, Journal of Real-Time Image Processing, Volume 20, article number 9, (2023).

[11] A. Tashk and M. A. Alavianmehr. Enhanced Pedestrian Detection and Tracking Using Multi-Person Pose Extraction and Deep Convolutional LSTM Network. IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics, 19-22 August 2024.