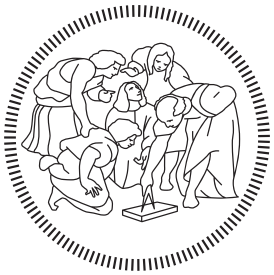


Politecnico di Milano

Simulation and analysis of a photonic neural network for chromatic dispersion compensation

Federico Pinto
Edoardo Pierluigi Leali

Contents

1 Introduction

The paper proposes an optical neural network (ONN) implemented as a silicon photonic perceptron to compensate chromatic dispersion (CD) in IM–DD links. The ONN realizes a trainable optical FIR filter: the input field is split into N delayed replicas (taps) with spacing Δt; each tap is weighted (amplitude and/or phase), and all paths are coherently recombined. Square-law photodetection provides the nonlinearity. Because CD is predominantly a phase distortion, a phase-only configuration is effective and reduces insertion loss. Weights are trained in the loop to maximize eye opening (reduce level overlap), improving BER(bit error rate).

We reproduce this mechanism end to end: a PAM4 IM–DD chain with an ONN precompensator, dispersive fiber, optical/electrical filtering, ASE noise set by OSNR, and direct detection. The ONN weights are optimized via a margin-based loss evaluated at the decision instants (two samples per symbol), with frame alignment by cross-correlation and trainers PSO and Adam. We evaluate BER, sensitivity, we explore N, Δt, and phase-only vs. complex weights. [1].

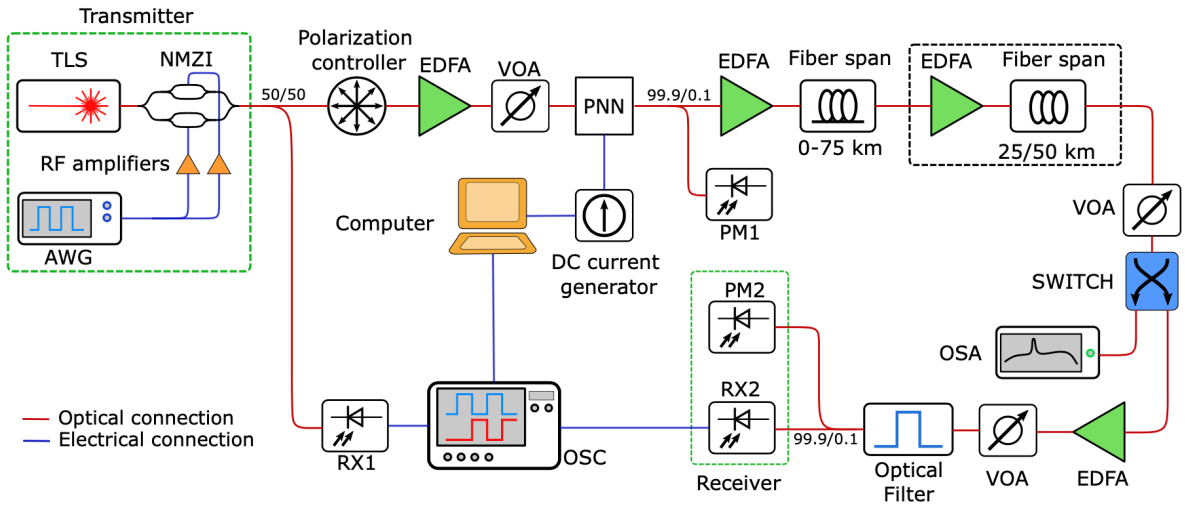


Figure 1: Experimental setup overview

2 Implementation

We implement an end-to-end ONN-based chromatic-dispersion pre-compensation chain as a modular pipeline. A PAM4 PRBS waveform is generated at the target baud rate with oversampling; an optical neural network pre-compensator applies a trainable N-tap delay–weight–sum, where each tap weight is realized by an MZI matrix; the field then propagates through a dispersive fiber with optional optical band-pass filtering and ASE noise controlled by OSNR; square-law detection and an electrical low-pass receiver follow. Frame alignment is performed by cross-correlation, and the ONN weights are optimized in-the-loop using a margin-based loss evaluated at symbol decision instants using PSO e Adam trainers.

2.1 Mach-Zehnder Interferometers

MZI are the basic tunable blocks of our photonic neural network: each tap uses an MZI to set the *amplitude* (via differential phase) and, when needed, an additional phase shifter to set the *phase* of the complex weight.

this is the blocks thats compose the MZI:

- **50:50 directional coupler**:

$$\mathbf{T}_C = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \mathrm{j} \\ \mathrm{j} & 1 \end{bmatrix}$$

- **phase shifters**:

$$\mathbf{T}_{\phi_d} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\mathrm{j}\phi_d} \end{bmatrix} \qquad \mathbf{T}_{\phi_u} = \begin{bmatrix} e^{-\mathrm{j}\phi_u} & 0 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{T}_{\theta} = \begin{bmatrix} e^{-\mathrm{j}\theta} & 0 \\ 0 & 1 \end{bmatrix}$$

Transmission matrix *We model the overall MZI exactly as follows*:

$$\mathbf{T}_{\mathrm{MZI}} = \mathbf{T}_C \mathbf{T}_\theta \mathbf{T}_C \mathbf{T}_{\phi_u} \mathbf{T}_{\phi_d} = -\mathrm{j}\, e^{-\mathrm{j}\theta/2} \begin{bmatrix} \sin\left(\frac{\theta}{2}\right) e^{-\mathrm{j}\phi_u} & \cos\left(\frac{\theta}{2}\right) e^{-\mathrm{j}\phi_d} \\ \cos\left(\frac{\theta}{2}\right) e^{-\mathrm{j}\phi_u} & -\sin\left(\frac{\theta}{2}\right) e^{-\mathrm{j}\phi_d} \end{bmatrix}$$

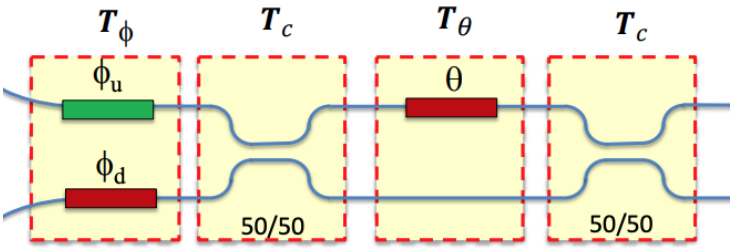


Figure 2: more intuitive view of MZI [2]

2.2 Photonic neural Network

Our ONN is a simple, trainable *delay–weight–sum* block that shapes the optical field *before* transmission (pre-compensation). The idea is: make N delayed copies of the input, give each copy a complex weight, and add them together. In compact form,

$$E_{\text{out}}(t) = \sum_{i=1}^{N} w_i \, E_{\text{in}}\big(t - (i-1)\Delta t\big).$$

How we implement. We operate on the complex optical field sampled with period T_s and enforce an integer tap spacing $D = \Delta t / T_s$. We form a data matrix $\mathbf{S} \in C^{L \times N}$ (Let L be the number of output samples) by stacking N shifted copies of the input at the start:

$$\mathbf{S}[n, i] = E_{\text{in}}\big[n - (i-1)D\big] \quad \text{(out-of-range indices set to zero).}$$

The tap weights from the MZI model are collected in $\mathbf{w} = [w_1, \ldots, w_N]^\top$. The ONN output is computed in one vectorized step as a matrix–vector product,

$$\mathbf{E}_{\text{out}} = \mathbf{S}\,\mathbf{w},$$

optionally followed by a global scalar normalization to keep the average optical power within the target budget.

The ONN block is placed *before* the dispersive fiber (pre-compensation). After this shaping, the signal propagates through the fiber and is then detected and filtered at the receiver.

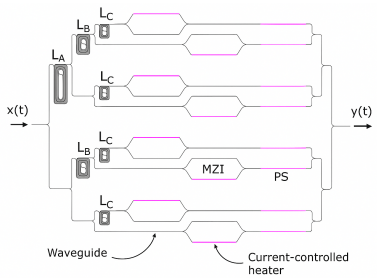


Figure 3: ONN real implementation[1]

2.3 Workflow of the paper experiment

This subsection explains the experimental workflow presented in the paper and how we mirror it in our simulator. The goal is to show the sequence of operations—what happens physically in the lab, and how we implement the same steps in code—without diving into unnecessary technicalities.

High-level flow: A PAM4 waveform is transmitted, pre-shaped by the optical neural network (ONN) acting as a trainable N-tap pre-compensator. The shaped field propagates through a dispersive fiber, passes through an optical band-pass, and is impaired by ASE noise (specified via OSNR on a 0.1 nm reference). At the receiver the signal is square-law detected and low-pass filtered; performance is evaluated using eye diagrams and BER at symbol instants. Training adjusts the ONN weights to maximize eye opening (phase-only or full weights), and validation/tests are run at fixed weights across distances and OSNR values (as in the paper).

code view: We implement the same chain with a small set of core functions:

- **ONN pre-compensation** — realized as a delay–weight–sum on the optical field: `PNN()` builds N delayed replicas (tap spacing Δt) and applies per-tap complex weights derived from the MZI model; outputs the pre-shaped field.
- **Fiber propagation & optical filtering** — `fiber_propagate_freqdomain()` applies the chromatic-dispersion phase ($H(\omega) = e^{-j\beta_2 L\omega^2/2}$); `opt_bpf_field()` models the optical band-pass used in the setup.
- **Noise and detection** — `add_ASE_OSNR_field()` injects complex ASE consistent with a target OSNR (0.1 nm reference); `rx_lpf_elec()` models the electrical LPF after square-law photodetection.

  These three steps are wrapped in `forward_rx_chain()` so the lab pipeline runs end-to-end in one call.
- **Symbol-time sampling & alignment** — `sample_and_align_auto()` aligns the received trace to the known PRBS (coarse cross-correlation + fine search) and extracts symbol-time samples at one or two neighboring offsets (used by our loss).
- **Training objective & optimizers** — `loss()` calls the forward chain and computes a margin-based L_2 loss on two samples per symbol to encourage symmetric eye opening; weights are tuned with `trainer_PSO()` and optionally refined with `trainer_ADAM()`.
- **Metrics** — `evaluate_BER_MAP()` estimates BER via Gaussian-MAP model with an inline ordered fallback for degenerate cases; auxiliary helpers compute loss and sensitivity curves.

Putting it together (what runs when). In practice, we use a compact sequence that mirrors the paper's measurement workflow:

1. **Train/fix the ONN.** Initialize weights, then call `trainer_PSO()` with the objective `loss()`; optionally refine with `trainer_ADAM()`. This corresponds to "tuning the photonic perceptron" to reopen the PAM4 eyes before the fiber.

2. **Run the link.** With fixed weights, call `forward_rx_chain()` to execute: ONN $\rightarrow$ fiber (CD) $\rightarrow$ optical BPF $\rightarrow$ ASE (OSNR) $\rightarrow$ photodetection $\rightarrow$ electrical LPF. This mirrors the lab chain.

3. **Align and sample.** Use `sample_and_align_auto()` to lock frames and read symbol-time samples as the paper does when extracting BER/eyes from acquired traces. :

4. **Evaluate.** Compute BER with `evaluate_BER_MAP()`; for loss-based analysis or plots we call `compute_loss_L2()` or small helpers (e.g., OSNR sweeps). These reproduce the paper's BER/eye/sensitivity readouts under matched parameters (baud, N, Δt, L, receiver BW, OSNR).

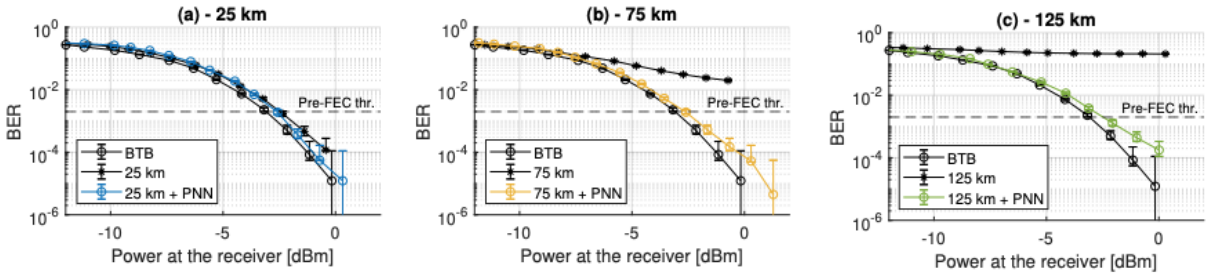
3 Result Analysis

we run the simulator with settings matched to the paper (baud rate, number of taps N, tap spacing Δt, fiber/CD, receiver bandwidth) and report the bit counts used for each BER estimate. The following pages refine this analysis: they first set the experimental context, then present our simulated outputs under matched conditions.

3.1 Paper result

This section distills the key experimental findings from the reference paper, aligned with the metrics we reproduce in our simulator: BER improvement, training behavior, comparison against a tunable dispersion compensator (TDC), and frequency-/wavelength-domain robustness.

BER improvement and eye diagrams across distance. With PAM4 at 10 Gbaud, the paper shows that PNN equalization brings BER curves close to the BTB baseline and re-opens the eyes at 25, 75, and 125 km.



(a) BER @ 25 km (b) BER @ 75 km (c) BER @ 125 km

Figure 4: Core evidence across distance: BER vs PRX. [1]

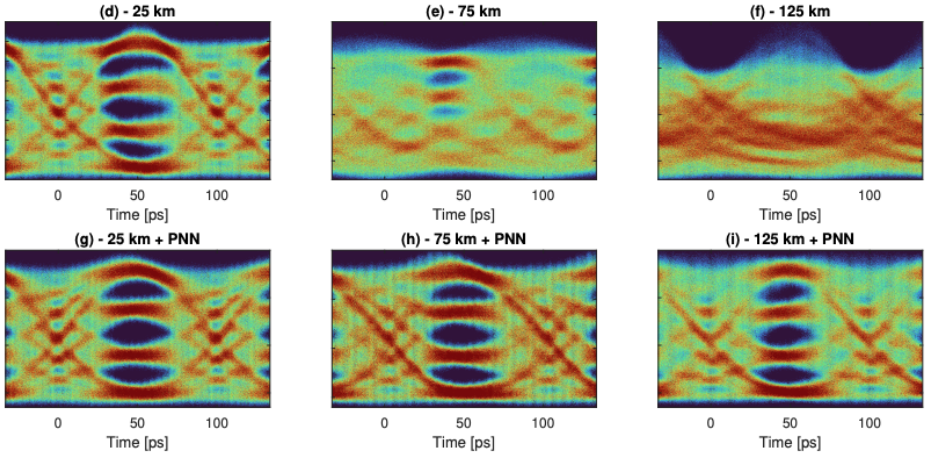


Figure 5: Eye diagram across distance. [1]

Training behavior (loss and optimizers). Phase-only with the L_2 loss yields more symmetric eyes and lower BER. Adam converges quickly but depends on initialization; PSO needs more evaluations but reaches lower final loss on average.

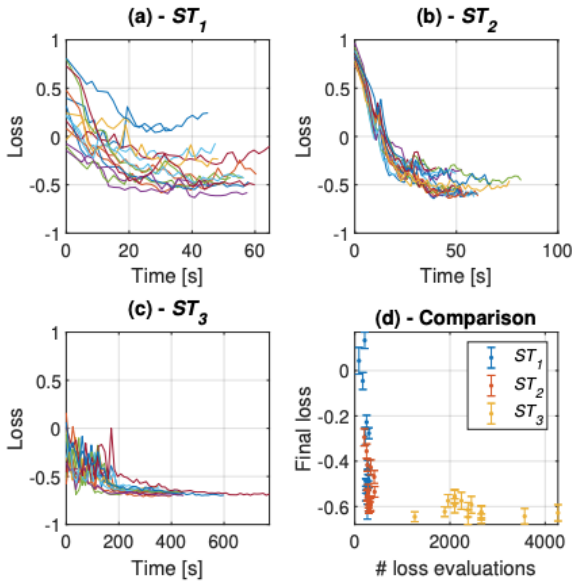
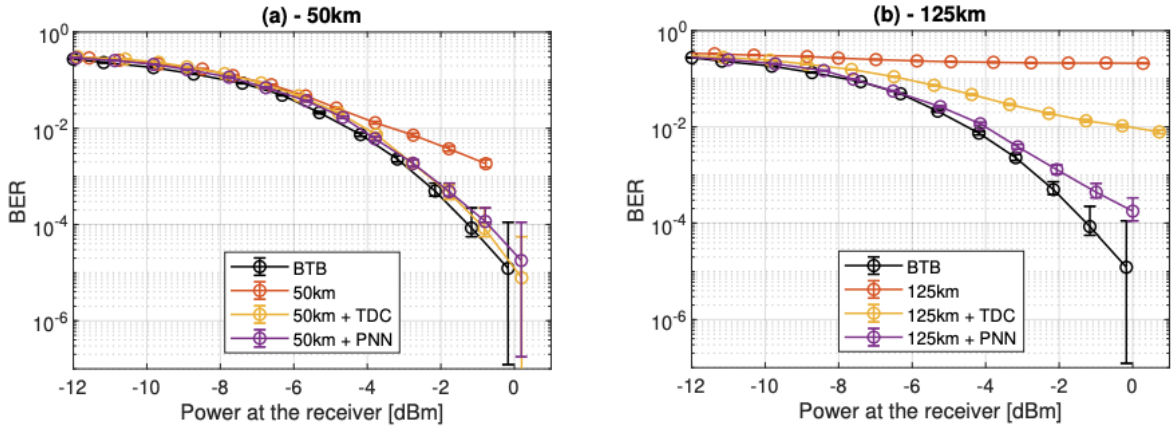


Figure 6: Training strategies and outcomes (PO + L_2 preferred). [1]

Benchmark vs TDC. Compared with a commercial tunable dispersion compensator, the PNN is comparable at 50 km and sustains equalization at 125 km.

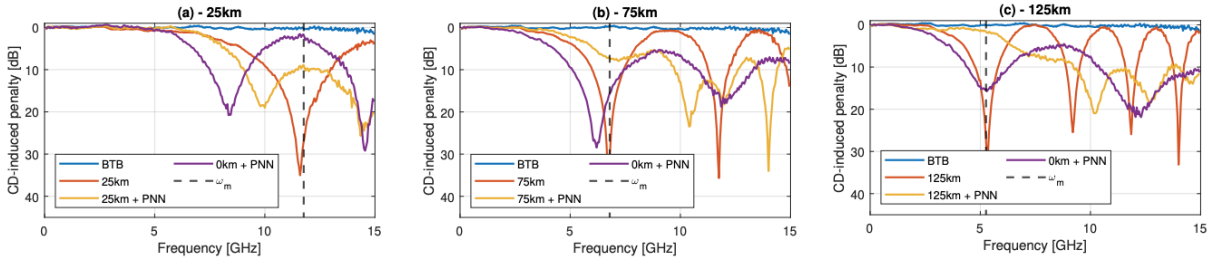


(a) 50 km: BTB / TDC / PNN

(b) 125 km: BTB / TDC / PNN

Figure 7: Direct comparison against a TDC. [1]

Frequency-domain penalty. The trained PNN flattens the CD-induced power penalty within the signal bandwidth. With fixed weights, sweeping the optical carrier increases loss a few GHz away from the training frequency; the expected ∼40 GHz periodicity is not observed experimentally due to delay errors, consistent with simulation.



(a) Penalty @ 25 km (b) Penalty @ 75 km (c) Penalty @ 125 km

Figure 8: CD-induced power penalty with and without PNN (BTB as reference). [1]

3.2 Our result

We trained the ONN pre–compensator and evaluated the link over an OSNR sweep and across multiple spans (25, 75, and 125 km), mirroring the paper's workflow. Below we report BER, loss trends, optimizer behaviour, eye diagrams, and the frequency–domain penalty measured on the detected RF response.

BER vs OSNR. Figure 9 shows the BER on *Validation* and *Test* sets (mean ± std over runs). The Test BER crosses the common pre-FEC threshold (2×10^{-3}, dashed) between 30–32 dB OSNR, while Validation exhibits a flatter curve.

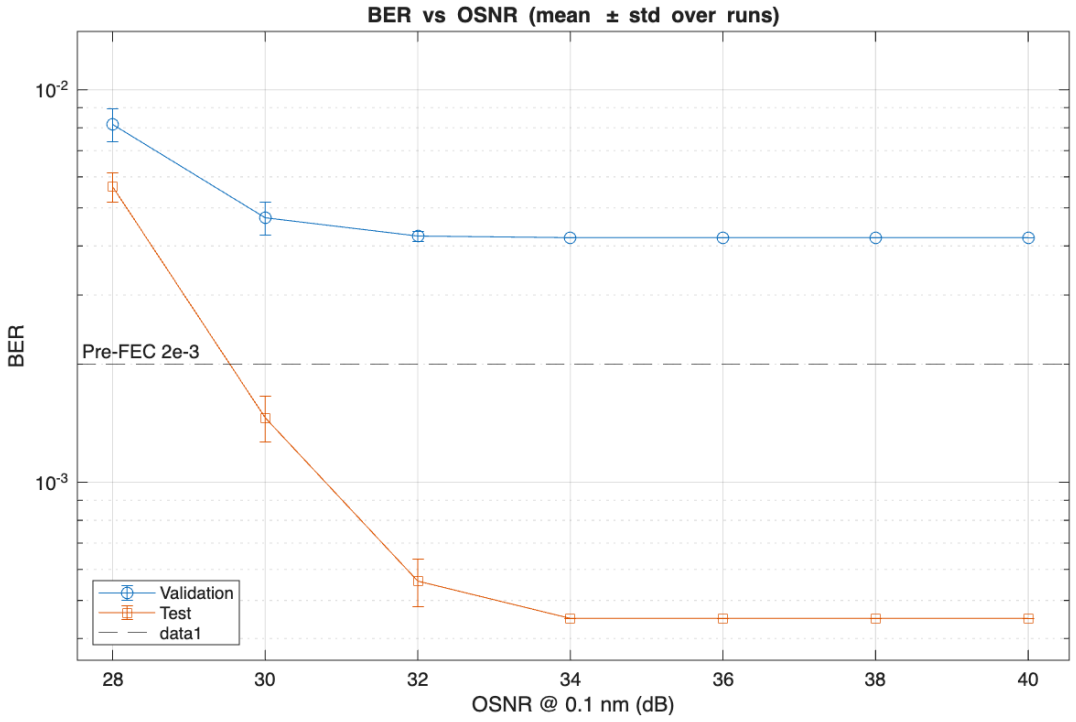


Figure 9: BER vs OSNR @ 0.1 nm (mean ± std). The dashed line marks the 2×10^{-3} pre-FEC threshold.

L_2 loss vs OSNR. The margin–based L_2 loss measured at symbol instants decreases monotonically with OSNR (Fig. 10), matching the BER trend.

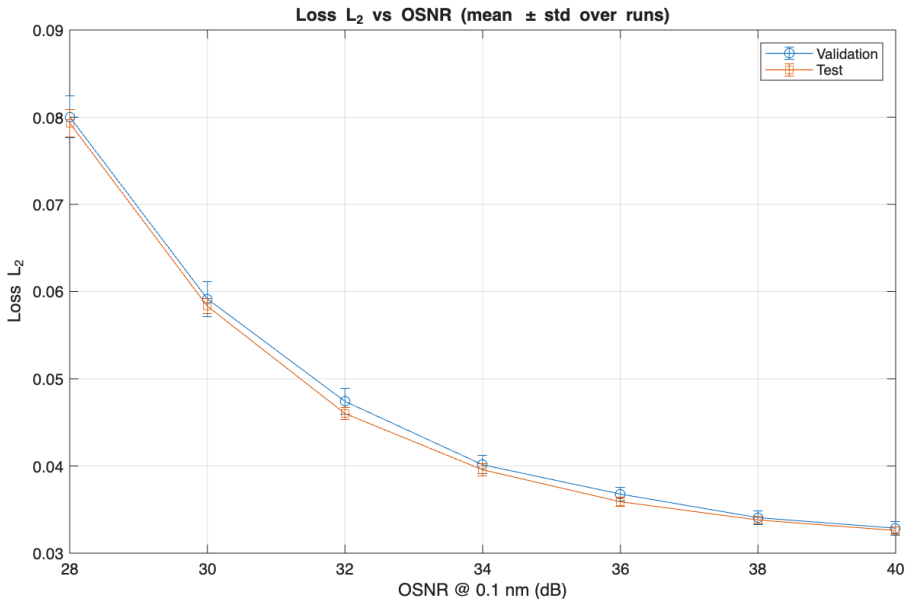


Figure 10: L_2 loss vs OSNR (mean ± std) on Validation/Test.

Optimizer behaviour. PSO provides a steady improvement of the best loss (Fig. 11); subsequent Adam refinement fine-tunes the solution but can show noisy per-step loss (Fig. 12).

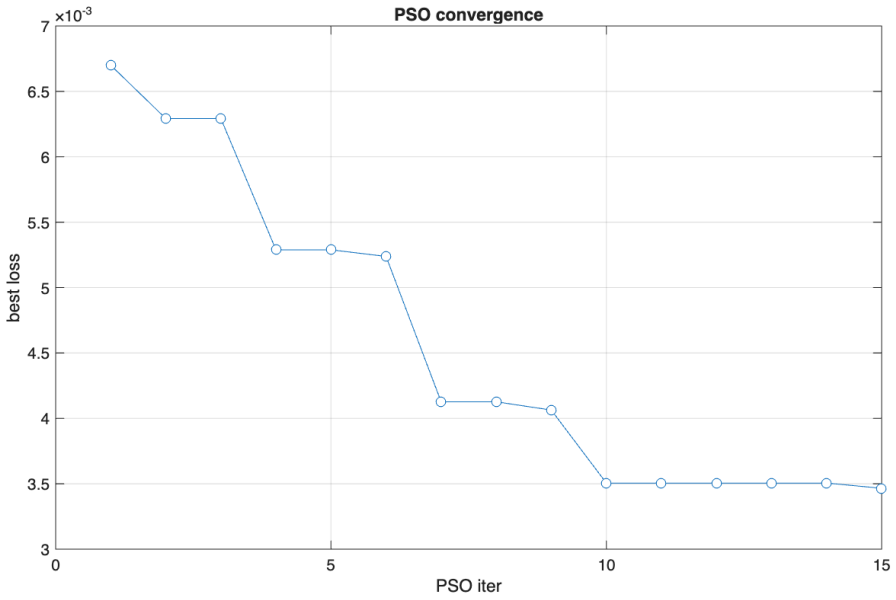


Figure 11: PSO convergence of the best loss vs iteration.

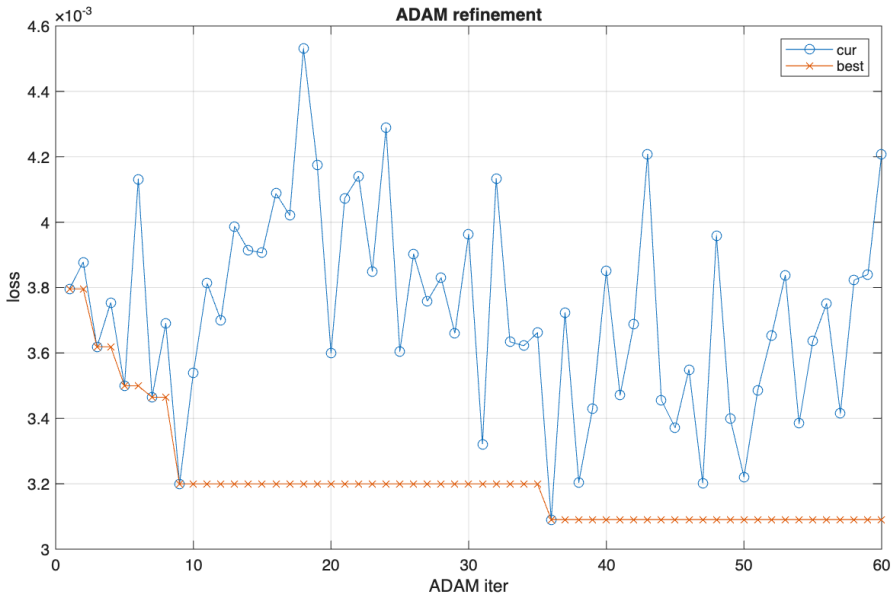


Figure 12: Adam refinement: current loss per step and running best.

Eye diagrams at 25/75/125 km. Across distances, fiber dispersion alone closes the PAM4 eyes; with the trained ONN the eyes reopen with clear level separation at all spans (Figs. 13–15).

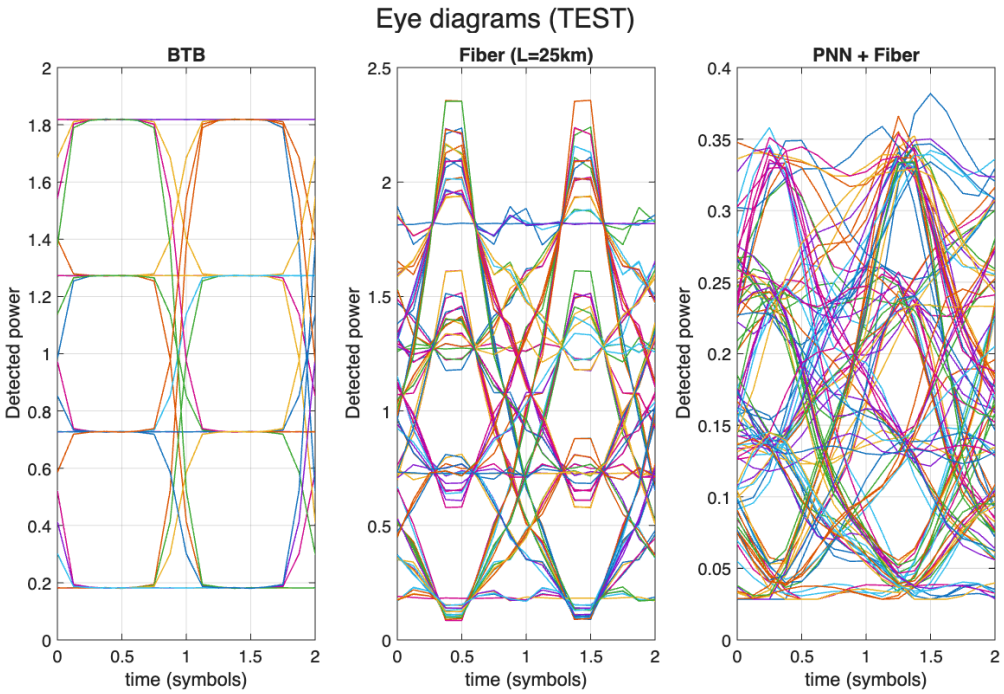


Figure 13: Eye diagrams (Test), $L = 25\,\mathrm{km}$: BTB, Fiber, and PNN + Fiber.

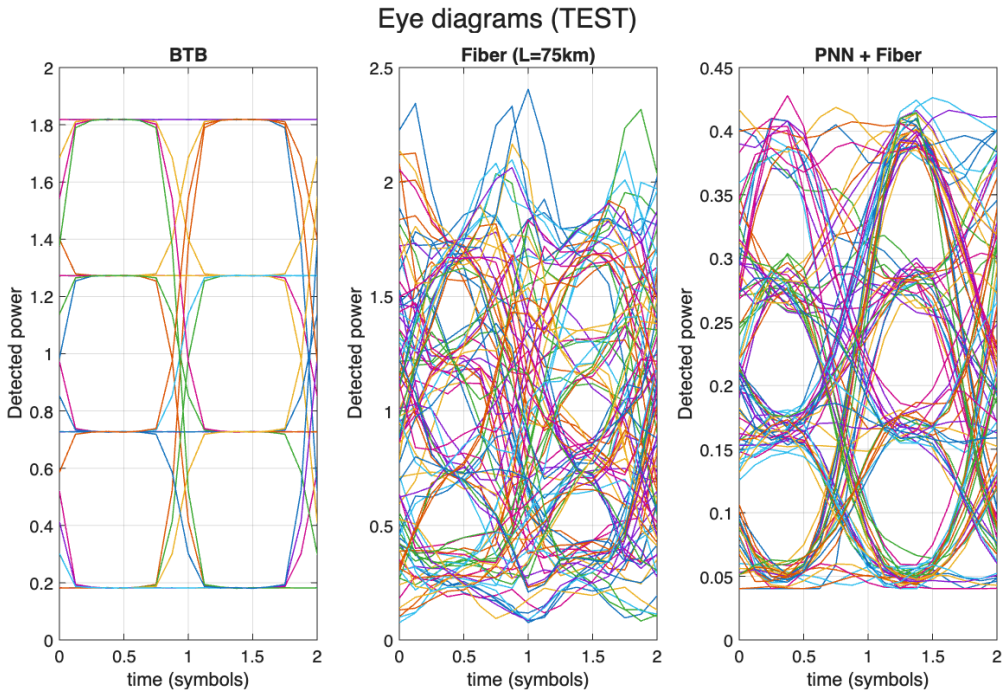


Figure 14: Eye diagrams (Test), $L = 75\,\text{km}$: BTB, Fiber, and PNN + Fiber.

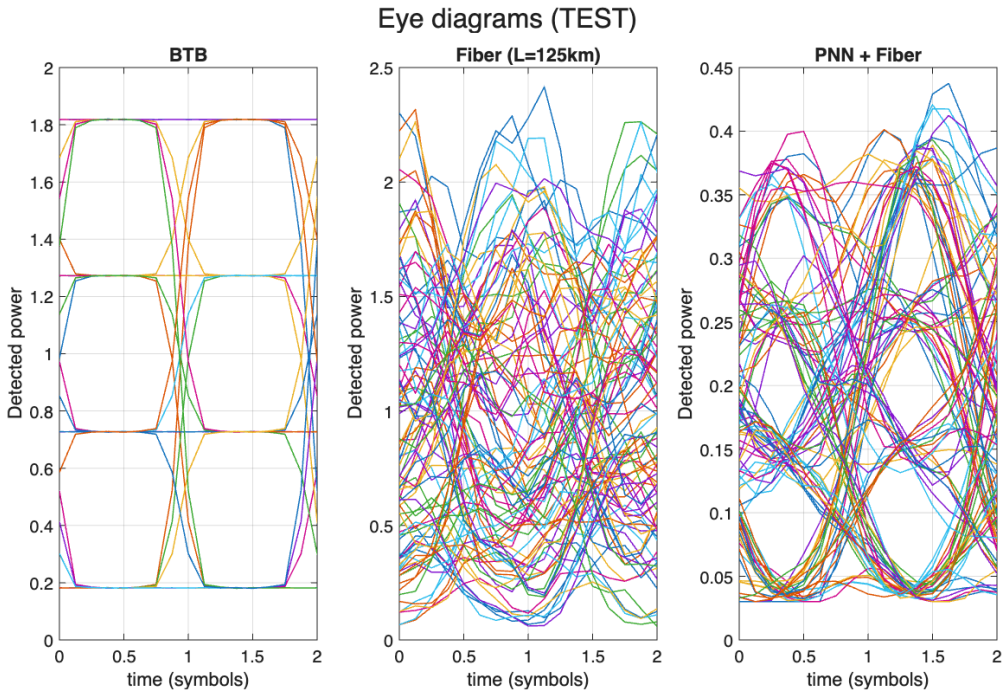


Figure 15: Eye diagrams (Test), $L = 125\,\text{km}$: BTB, Fiber, and PNN + Fiber.

CD–induced power penalty. We swept the detected RF tone with and without the PNN (OSNR = 34 dB, $L = 125\,\text{km}$). The trained ONN flattens the in-band response and suppresses the CD notches (Fig. 16).

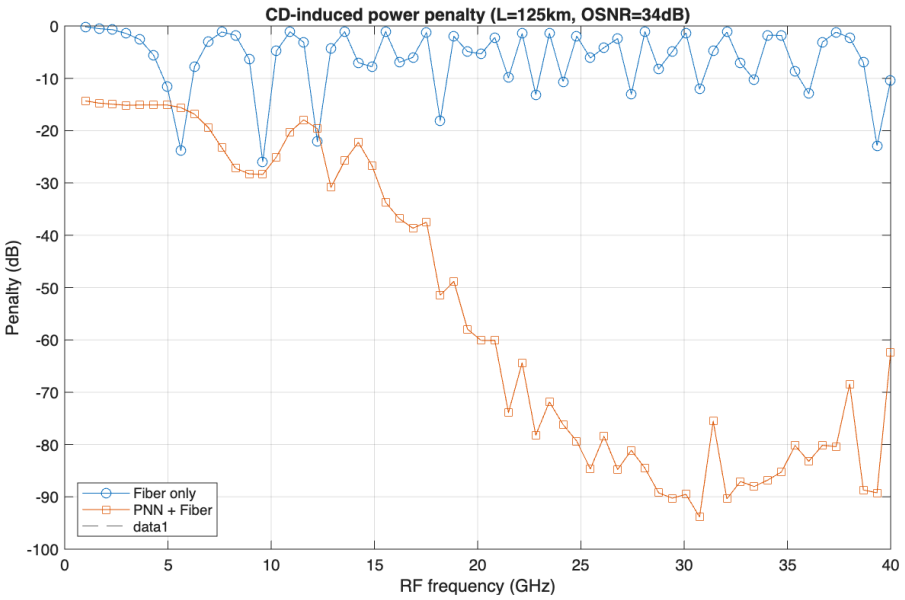


Figure 16: CD-induced power penalty (equalized vs fiber-only) at $L = 125\,\text{km}$, $\text{OSNR} = 34\,\text{dB}$.

3.3 Analysis and comparison

Qualitative agreement with the paper. Our end-to-end trends reproduce the main experimental observations in [1]: (i) BER improvement reaching pre-FEC at practical OSNR values; (ii) clear eye reopening at long spans; (iii) PSO for robust global progress with Adam for local polish; and (iv) a flattened in-band RF penalty after equalization.

Where our curves differ (and why). We note three systematic differences and their likely causes:

1. *Validation vs Test gap in BER.* Validation is flatter and slightly above Test at high OSNR due to our split (easier/shorter frames for Validation) and conservative stopping on the Validation loss.

2. *Absolute BER at the crossing.* Our pre-FEC crossing (30–32 dB OSNR) can shift versus [1] because we model ASE only (no RIN/thermal/jitter), use idealized pulse shaping, and omit fabrication-induced delay/weight errors.

3. *Penalty ripple.* We observe very deep notch suppression and a smooth floor across 5–35 GHz; measurements in [1] do not show the ideal $\sim$40 GHz periodicity, likely due to path-delay errors and coupler imbalance—non-idealities we plan to inject in future work.

Takeaways. With matched settings (baud, N, Δt, receiver BW), the simulator captures the PNN's value proposition: compact trainable pre-compensation that restores eye symmetry and improves BER over dispersive IM–DD links. Remaining gaps stem from intentionally simplified hardware models and can be reduced by adding measured tap tolerances and richer noise/impairments.

4 Conclusion

We developed an end-to-end simulator of ONN-based chromatic-dispersion pre-compensation that mirrors the paper's workflow: an N-tap delay–weight–sum on the optical field with MZI-derived tap weights, dispersive fiber propagation, direct detection with receiver filtering, and a compact analysis stage (alignment, thresholds, BER and eye metrics).

Agreement with experimental trends. Under matched settings, the simulator reproduces the main behaviors reported in the paper: BER reduction and eye reopening across distance; the effectiveness of phase-only weights with the L_2 loss; training dynamics consistent with Adam vs. PSO; and robustness comparable to a TDC at short spans while sustaining performance at longer spans. Frequency-/wavelength-domain behavior also aligns qualitatively, including the flattening of CD-induced penalty within the signal band and the sensitivity to carrier detuning.

Limitations and simplifications. Differences remaining between simulation and measurements are largely explained by modeling choices: idealized transmitter/receiver pulse shaping, restricted noise modeling (e.g., no RIN/thermal/jitter), and the omission of polarization effects and fabrication-induced delay errors. These simplifications were intentional to keep the model transparent and fast, but they bound the absolute accuracy.

Future Steps. Next steps include incorporating additional hardware non-idealities and tolerances, refining bandwidth/noise models, and extending the study to wavelength detuning and WDM scenarios. Overall, the simulator provides a clear and faithful baseline for ONN-based dispersion pre-compensation and a solid starting point for hardware-aware exploration.

References

[1] E. Staffoli, G. Maddinelli, and L. Pavesi, "A Silicon Photonic Neural Network for Chromatic Dispersion Compensation in 20 Gbps PAM4 Signal at 125 km and Its Scalability up to 100 Gbps," Sep. 2024. [Online]. Available: https://arxiv.org/abs/2409.03547

[2] F. Morichetti and A. Melloni, "Photonic computing: Course slides," Lecture slides, Politecnico di Milano, 2025.