# Interpretable Optical Network Fault Detection and Localization with Multi-Task Graph Prototype Learning

XIAOKANG CHEN[1,2], XIAOLIANG CHEN[2,*], AND ZUQING ZHU[2]

[1] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
[2] School of Information Science and Technology, University of Science and Technology of China, Hefei, China
[*] xlichen@ieee.org

Compiled May 30, 2025

**The recent advances in machine learning (ML) have promoted data-driven automated fault management in optical networks. However, existing ML-aided fault management approaches mainly rely on black-box models that lack intrinsical interpretability to secure their trustworthiness in mission-critical operation scenarios. In this paper, we propose an interpretable optical network fault detection and localization design leveraging multi-task graph prototype learning (MT-GPL). MT-GPL models an optical network and the optical performance monitoring data collected in it as graph-structured data and makes use of graph neural networks to learn graph embeddings that capture both topological correlations (for fault localization) and fault discriminative patterns (for root cause analysis). MT-GPL interprets its reasoning by: *i*) introducing a prototype layer that learns physics-aligned prototypes indicative of each fault class using the Monte Carlo tree search method, and *ii*) performing predictions based on the similarities between the embedding of an input graph and the learned prototypes. To enhance the scalability and interpretability of MT-GPL, we develop a multi-task architecture that performs concurrent fault localization and reasoning with node-level and device-level prototype learning and fault predictions. Performance evaluations show our proposal achieves $>6.5\%$ higher prediction accuracy than the multi-layer perceptron model, while the visualizations of its reasoning processes verify the validity of its interpretability.**

http://dx.doi.org/10.1364/ao.XX.XXXXXX

## 1. INTRODUCTION

Driven by the rapid development of 5G/6G, Internet of Things (IoT) and artificial intelligence (AI) technologies, diverse applications have flourished and produced exponentially growing traffic. According to Ericsson's report, the global monthly network traffic increased by 25% from Q1 2023 to Q1 2024, reaching 145 exabytes (EB) [1]. This surge solidifies optical networks as the cornerstone of modern digital infrastructures, and in turn, calls for powerful optical network fault management solutions to secure high service availability.

Unlike fiber cuts or port failures that cause immediate signal losses, soft failures in optical networks have attracted more research interest recently owing to their complex yet covert patterns [2, 3]. In particular, soft failures caused by device malfunctioning, aging or physical-layer attacks can lead to gradual signal degradations or spatially correlated optical performance monitoring (OPM) data anomalies which cannot be easily discriminated from normal fluctuations using simple threshold-based rules. Consequently, machine learning (ML), for its capability of learning complex mappings automatically from data, is emerging as a promising technique for actuating data-driven cognitive fault management in optical networks [4–6]. Previous studies have reported various ML approaches (supervised [4], unsupervised [6] and semi-supervised [7]) applied to fault detection [8], identification [9] and localization tasks [10]. Despite these ML designs have demonstrated superior performance (in terms of accuracy, adaptability, *etc.*) over conventional solutions, they mostly rely on black-box models (*e.g.*, neural networks) that barely provide any interpretability to their reasoning processes. As a result, their trustworthiness [11], and thereby, applicability to mission-critical operation conditions can hardly be secured. Indeed, some ML algorithms like decision trees are intrinsically interpretable, their simple structures restrict their representation capabilities and scalability, rendering them less effective for processing high-dimensional data.

Lately, explainable AI has gained increasing attentions from the optical network community. For instance, researchers have attempted to improve the explainability of existing ML solutions in fault localization [12] and quality-of-transmission (QoT) estimation [13]. Nevertheless, these works unanimously employ post-hoc attribution methods like SHAP (SHapley Additive exPlanations) [14] to explain the models' inferences rather than enhancing the models' intrinsic interpretability itself. In other

words, insufficiency of intrinsic transparency remains a critical limitation of existing post-hoc explainable ML solutions for optical networks.

In this paper, we aim at filling the aforementioned gap by proposing an intrinsically interpretable ML design for fault detection and localization in optical networks. Our design models the OPM data in an optical network as a graph and makes use of graph neural networks (GNNs) to learn graph embeddings that capture both topological correlations and fault discriminative patterns. Then, we introduce a graph prototype layer to identify physics-aligned prototypes (*e.g.*, a subgraph or a subset of features) decisive for classifying samples to each fault class. A multi-task learning mechanism is also applied to enhance the model interpretability and scalability by learning separate prototypes for current node-level (for fault localization) and device-level (for root cause analysis) predictions. Simulation results show the effectiveness of our proposal while verifying the validity of its interpretability.

The rest of this paper is organized as follows. In Section 2, we provide a brief review of the state of the art on fault localization and interpretable ML design in optical networks. In Sections 3 and 4, we describe the multi-task GNN-based framework and the graph prototype learning method, respectively. We provide and discuss the results in Section 5 and finally conclude the paper with Section 6.

## 2. RELATED WORK

### A. ML-aided Fault Localization in Optical Networks

Traditional fault localization methods in optical networks primarily rely on rule-based expert systems and statistical analysis of alarm correlations. For instance, when a device fails in an optical network, conventional methods rely on experienced network administrators to manually analyze alarm logs and locate network faults[15]. These methods have a reliance on expert experience and cannot analyze faults caused by the combined effects of optical power, spectral efficiency, and nonlinear phase noise.

In contrast, deep learning (DL) frameworks automate feature extraction from multi-dimensional telemetry data (*e.g.*, optical spectrum analyzers, BER monitors). Convolutional neural networks (CNNs) significantly improve the identification accuracy of polarization mode dispersion (PMD)-induced waveform distortions by learning spatial patterns from two-dimensional spectrograms, compared to traditional methods [16]. Recurrent architectures, such as long short-term memory (LSTM), have been used to capture temporal dependencies in OPM sequences, enabling accurate detection and identification of soft faults associated with Electro-optic Modulators (ECL), erbium-doped fiber amplifiers (EDFA), and nonlinear interactions (NLI) [17].

Compared with traditional CNN or recurrent neural network (RNN) models, GNNs have better topological perception and dynamic adaptability, making them a powerful tool for fault detection in optical networks. Literature [18] utilized GNNs to achieve a distributed fault management design in optical networks, effectively addressing the issue of neuron expansion in large-scale networks. In literature [19], GNNs and alarm knowledge graphs were used to identify the root causes of alarms by reasoning the relationships between alarms.

While DL models with attention mechanisms [20, 21] demonstrate remarkable accuracy in soft failure detection, their lack of interpretability hinders rapid root cause analysis in practical scenarios – a critical operational imperative for high-reliability optical networks where transparent decision logic is essential to accelerate fault recovery cycles and mitigate cascading misdiagnosis risks [22].

### B. Interpretable ML in Optical Networks

Currently, in optical network tasks, particularly in fault management, there are few research cases on interpretable methods. Most of them involve using the SHAP method to analyze the feature importance of existing model decisions. However, the models themselves still lack interpretability.

The authors of [23] applied the SHAP method to perform feature attribution analysis on the XGBoost model's predictions, which calculated the contribution of each feature to fault detection outcomes. Experimental results demonstrated that this approach achieved a high accuracy of 99.84% on the dataset. Furthermore, it identified the average environmental temperature as the most critical feature influencing model decisions, thereby enabling precise localization of fault causes.

In [24], the authors also employed the SHAP method to uncover the critical feature importance of different fault types, identifying specific timestamp events linked to connector faults, stress faults, shutdown faults, and others. Through the explainer dashboard, the research provided global explanations of the model and delves into local explanations, enhancing transparency and trust in the model's decision-making process.

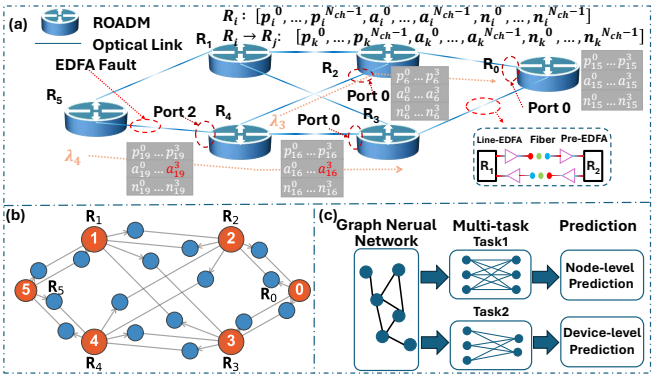


**Fig. 1.** Multi-task GNN-based fault management framework: (a) an example of OPM data correlation in a six-node topology; (b) graph modeling of the optical network in (a); and (c) multi-task GNN structure for joint fault detection and localization.

## 3. MULTI-TASK GNN-BASED FAULT MANAGEMENT

Optical network faults can exhibit diverse and complex characteristics while inducing correlated OPM data anomalies in multiple monitoring positions over signal propagation paths. Fig. 1(a) exemplifies the topological correlations among the OPM data collected at different reconfigurable optical add-drop multiplexer (ROADM) nodes in the presence of an EDFA malfunctioning between nodes $R_5$ and $R_4$. The fault varies the spectral signatures of $\lambda_4$ observed at nodes $R_4$ (port 2) and $R_3$ (port 0), whereas those at $R_2$ (port 0) remains normal because lightpath $R_4 \rightarrow R_2 \rightarrow R_0$ does not traverse the faulty amplifier. Consequently, accurate detection and localization of these faults necessitates not only powerful feature learning capabilities to discriminate normal and abnormal behaviors, but also multi-node data correlation analyses to pinpoint the root causes. Unlike the work in [21] which deploys OPMs at almost all device I/O ports, our method assumes installations of OPMs solely at

ROADM ingress ports of each optical link. By utilizing graph structural characteristics and fault-induced signal variation patterns, this configuration maintains manageable data collection scale while ensuring complete fault detection capability throughout transmission paths.

To this end, we leverage GNNs' speciality of learning graph embedding to develop a multi-task GNN model for joint fault detection and localization in optical networks. In particular, we first model an optical network as a graph as shown in Fig. 1(b), where every ROADM node and fiber link is modeled as a node (denoted by red and blue circles, respectively) while the edges represent the physical connectivity. The multi-dimensional spectral data [*e.g.*, optical power, amplified spontaneous emission (ASE) noise and nonlinear impairment for each wavelength] collected at each position are assigned to the corresponding node as its feature representation. This way, we structure the network-wide OPM data to create a topological representation of the optical network, allowing for exploiting the spatial and spectral features influenced by different network components. The graph representation is fed as input to a multi-task GNN (see Fig. 1(c)) for fault detection and localization.

Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote a graph model mentioned above, where $\mathcal{V}$ denotes the set of ROADMs and optical links, and $\mathcal{E}$ represents the edges. Each node $v_i \in \mathcal{V}$ has a multi-dimensional state $\mathbf{x}_i \in \mathbf{X}$ ($\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$) which is formed as,

$$\mathbf{x}_i = \left[ p^0, \ldots, p^{N_{\text{ch}}-1}, a^0, \ldots, a^{N_{\text{ch}}-1}, n^0, \ldots, n^{N_{\text{ch}}-1} \right], \quad \textbf{(1)}$$

where $p^c$, $a^c$ and $n^c$ represent the optical power, ASE noise level and nonlinear impairment metrics of channel $c$, respectively, and $N_{\text{ch}}$ is the total number of wavelength channels (*i.e.*, $c \in \{1, \ldots, N_{\text{ch}}\}$). Then, the multi-task GNN processes $G$ with iterative message passing and aggregation operations following the graph convolutional network (GCN) scheme [25], *i.e.*,

$$\mathbf{h}_i^{(l+1)} = \sigma\left( \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{1}{\sqrt{\delta_i \delta_j}} \mathbf{h}_j^{(l)} \theta^{(l)} \right), \quad \textbf{(2)}$$

where $\mathbf{h}_i^{(l)} \in \mathbb{R}^{d^{(l)}}$ denotes the state vector of node $v_i$ at layer $l$ (the $l$-th iteration), $\mathcal{N}(i)$ represents its neighbor set in $G$, and $\delta_i = 1 + \sum_{j \in \mathcal{N}(i)} 1$ is a normalization coefficient used for ensuring stable gradient propagation across nodes with diverse degrees. Note that, $\mathbf{h}_i^{(0)} = \mathbf{x}_i$. The trainable parameter matrix $\theta^{(l)}$ and nonlinear activation function $\sigma(\cdot)$ transform the states into higher-level embeddings. Through such multi-layer embedding, the GNN progressively captures local-to-global topological patterns. Ultimately, we employ separate neural network heads to perform hierarchical fault detection and localization (multi-class classifications) with the learned graph embeddings. Specifically, the node-level predictions tell whether a fault exists, and if so, which node (a ROADM or a fiber link) is suspect of the fault, while the device-level predictions further associate the fault with a specific network component on that node. In other words, the node-level task focuses on topological feature analyses, whereas the device-level task pays more attention to fault discriminative features. This hierarchical design secures its scalability to larger networks by constraining the scale of the output space to be $O(|\mathcal{V}|)$ rather than $O(N_{\text{devices}} \cdot |\mathcal{V}|)$, where $N_{\text{devices}}$ is the number of device types involved.

## 4. INTERPRETABILITY ENHANCEMENT WITH GRAPH PROTOTYPE LEARNING

We leverage graph prototype learning to enhance the multi-task GNN framework in Fig. 1 with intrinsical interpretability during fault detection and localization.

### A. Prototype Graph Network

The proposed prototype graph network consists of three core components: learnable prototype vectors that capture discriminative fault patterns, a fixed-weight classification layer for theoretical guarantees, and a similarity computation module that translates embedding distances into interpretable probabilities.

#### *A.1. Prototype Layer Architecture*

Let $\mathbf{e} = f_\theta(G)$ ($\mathbf{e} \in \mathbb{R}^{d_e}$) denote the graph embedding generated by the GNN encoder $f_\theta(\cdot)$, where $d_e$ represents the embedding dimension (empirically set to 128 in our implementation). The prototype layer operates through the following components.

- **Learnable Prototypes**. For each of the $C$ classes, we initialize $m$ prototype vectors $\{\mathbf{p}_k^m\}_{m=1}^M \in \mathbb{R}^{d_e}$ that characterize typical fault patterns. These prototypes are randomly initialized from a uniform distribution and updated during training.

$$\mathbf{p}_k^m \sim \mathcal{U}(0,1)^{d_e}, \quad k \in \{1, \ldots, C\},\ m \in \{1, \ldots, M\}. \quad \textbf{(3)}$$

- **Similarity Activation**. For an input embedding $\mathbf{e}$, we compute its similarity to each prototype through $L_2$-based logarithmic transformation:

$$s_j = log\left(\frac{\exp(\|\mathbf{e} - \mathbf{p}_j\|_2^2) + 1}{\exp(\|\mathbf{e} - \mathbf{p}_j\|_2^2) + \epsilon}\right), \quad \epsilon = 10^{-4}, \quad \textbf{(4)}$$

  where $\epsilon$ ensures numerical stability in division. Here, for the sake of clarity, we combine the subscript and superscript of $\mathbf{p}_k^m$ and unify the notation of a prototype as $\mathbf{p}_j$. We use the two notations interchangeably in the rest of the paper.

- **Phase-Adaptive Classifier**. The classification layer utilizes an initially prototyped weight matrix $\mathbf{W} \in \mathbb{R}^{C \times (C \times M)}$ governed by phased learning:

$$\mathbf{W}_{k,j}^{(t)} = \begin{cases} \begin{cases} 1, & \text{if } \mathbf{p}_j \in \{\mathbf{p}_k^m\}_{m=1}^M, \\ -0.5, & \text{otherwise}, \end{cases} & \text{for } 0 \le e \le E_{\text{warm}}, \\ \underbrace{\mathbf{W}_{k,j}^{(t-1)} - \eta \nabla_{\mathbf{W}_{k,j}} \mathcal{L}}_{\text{learnable}}, & \text{for } e > E_{\text{warm}}, \end{cases} \quad \textbf{(5)}$$

  where $E_{\text{warm}}$ specifies the number of epochs for prototype stabilization, and $\mathcal{L}$ is the total loss. Cross-task consistency is maintained in similarity activation, while classifier weights become trainable post warmup.

#### *A.2. Loss Function*

We employ a cross-entropy loss $\mathcal{L}_{\text{CE}}$ to enhance classification accuracy and a separation loss $\mathcal{L}_{\text{Sep}}$ to enforce that each graph embedding $\mathbf{e}$ maintains distant from prototypes of non-corresponding classes. By combining the two complementary

terms, we jointly optimize classification accuracy [26] and prototype interpretability:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{CE}} + \gamma \mathcal{L}_{\text{Sep}},\\ \mathcal{L}_{\text{CE}} &= \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{C} y_{n,k}\log\left(\frac{\exp(z_{n,k})}{\sum_{k'=1}^{C}\exp(z_{n,k'})}\right),\\ \mathcal{L}_{\text{Sep}} &= -\frac{1}{N}\sum_{n=1}^{N}\min_{\mathbf{p}_{k'}^{m'}\in\{\mathbf{p}_k^m\}_{m=1}^{M}, y_{n,k'}=0}\|f_\theta(G_n)-\mathbf{p}_{k'}^{m'}\|_2^2.\end{aligned} \tag{6}$$

Here, $N$ is the total number of training samples, $G_n$ is the graph input of sample $n$, $z_{n,k}$ is the logit output for sample $n$ and class $k$, and $y_{n,k}$ is the true label. The hyperparameters $\gamma$ control the importance of separation.

## B. Active Learning for Prototype Refinement

The prototypes obtained with the aforementioned training process do not correspond to physical significance. We next employ a prototype refinement process to establish physical interpretability through Monte Carlo tree search (MCTS)-guided subgraph discovery and active sample selection. The key procedures are as follows.

### B.1. Prototype-aware Sample Selection

We first apply active learning to select the most informative samples $\mathcal{S}_k^m$ for each prototype $m$ within class $k$ to reduce the search space of MCTS. Let $\{\mathbf{e}\}_k \subset \{f_\theta(G_n)\}_{n=1}^N$ denote the GNN embeddings of the graphs belonging to class $k$. We calculate $M$ medoids of $\{\mathbf{e}\}_k$ in three phases [27],

**Phase 1: Medoid Initializaiton.** Randomly select $M$initial medoids from $\{\mathbf{e}\}_k$,

$$\{\mathbf{ME}_1^{(0)}, \mathbf{ME}_2^{(0)}, \ldots, \mathbf{ME}_M^{(0)}\} \subseteq \{\mathbf{e}\}_k. \tag{7}$$

**Phase 2: Iterative Optimization.** At iteration $t$, for each $e_p \in \{\mathbf{e}\}_k$, perform nearest-medoid assignment across all $M$ medoids $\{\mathbf{ME}_1^{(0)}, \mathbf{ME}_2^{(0)}, \ldots, \mathbf{ME}_M^{(0)}\}$:

$$\mathbf{CL}_m^{(t)} = \{\mathbf{e_p} : m = \arg\min_{m'\in\{1,2,\ldots,M\}} \|\mathbf{e}_p - \mathbf{ME}_{m'}^{(t)}\|_2\}, \tag{8}$$

Then, compute new medoid for all $M$ cluster $\{\mathbf{CL}_1^{(t)}, \mathbf{CL}_2^{(t)}, \ldots, \mathbf{CL}_M^{(t)}\}$,

$$\mathbf{ME}_m^{(t+1)} = \arg\min_{\mathbf{e}_p\in\mathbf{CL}_m^{(t)}} \sum_{\mathbf{e}_q\in\mathbf{CL}_m^{(t)}} \|\mathbf{e}_p - \mathbf{e}_q\|_2^2 \tag{9}$$

**Phase 3: Convergence.** Terminate when $\mathbf{ME}_l^{(t+1)} = \mathbf{ME}_m^{(t)}$,for all $m = 1, \ldots, M$.

Then, the candidate set $\mathcal{S}_k^m$ is composed by $\varsigma$ ($\varsigma = 10$) graph samples closest to the medoid $\mathbf{ME}_m^*$,

$$\mathcal{S}_k^m = \left\{ G_n \,\middle|\, {}_{y_{n,k}^m=1, \|f_\theta(G_n)-\mathbf{ME}_k^*\|_2^2 \le \underbrace{\psi_\varsigma}_{\text{global}}} \right\}, \tag{10}$$

where $\psi_\varsigma$ is the $\varsigma^{\text{th}}$ shortest distance across $\|f_\theta(G_n) - \mathbf{ME}_m^*\|_2^2$, $\forall y_{n,k} = 1$. As a result, we reduce the search space of MCTS from $\mathcal{O}(N)$ to $\mathcal{O}(1)$.

### B.2. MCTS-based Subgraph Rollout

For each candidate graph $G \in \mathcal{S}_k^m$, MCTS discovers the minimal fault-indicative subgraph $G^{\text{sub}}$ as the refined prototype for class $k$. Specifically, we initialize a coalition $\mathcal{C} = G$ as the root of the Monte Carlo tree and then perform the following four-phase explorations.

**Phase 1: Node Expansion**. Add $K$ nodes with the highest degrees in $\mathcal{C}$ to $\mathcal{V}_{\text{exp}}$ and expand the tree with $K$ child vertices using $\mathcal{V}_{\text{exp}}$.

**Phase 2: State Selection**. For each child vertex, remove from $\mathcal{C}$ a node in $\mathcal{V}_{\text{exp}}$ and preserve the largest connected subgraph $G'$ after removal of the node to maintain structural validity.

**Phase 3: Rollout Simulation**. Calculate the subgraph-prototype similarity as the reward of each child vertex,

$$r(G') = \|f_\theta(G') - \mathbf{p}_k^m\|_2^2, \quad v_{exp} \in \mathcal{V}_{\text{exp}}. \tag{11}$$

**Phase 4: Backpropagation**. Node statistics are updated via the polynomial upper confidence tree (PUCT) algorithm:

$$\begin{aligned} W(v_{exp}) &\leftarrow W(v_{exp}) + r(G'),\\ Count(v_{exp}) &\leftarrow Count(v_{exp}) + 1,\\ Q(v_{exp}) &= \frac{W(v_{exp})}{Count(v_{exp})},\\ U(v_{exp}) &= c_{\text{puct}} \cdot P(v_{exp}) \cdot \frac{\sqrt{Count(\text{parent}(v_{exp}))}}{1 + Count((v_{exp})}. \end{aligned} \tag{12}$$

Here $Count(v)$ denotes the visit count of node $v_{exp}$, $Count(\text{parent}(v_{exp}))$ is the parent node's visit count, $W(v_{exp})$ accumulates the total reward, $Q(v_{exp})$ calculates the average reward per visit (exploitation factor), and $U(v_{exp})$ promotes exploration of under-visited nodes. The hyperparameter $c_{\text{puct}}$ governs the exploration-exploitation tradeoff, where larger values encourage exploration while smaller values emphasize reward exploitation.

Based on the selection citerion $Q(v_{\text{exp}}) + U(v_{\text{exp}})$, selecting node $v_{exp}$, upadating the coalition as $\mathcal{C} = \mathcal{C} \setminus \{v_{\text{exp}}\}$. Then returing to Phase 1 to repeat the four-step process. This iterative pruning continues until the coalition size satisfies $|\mathcal{C}| \le \tau_{\min}$ ($\tau_{\min} = 4$). The final step compares all candidate subgraphs $G'$ generated during the pruning iterations, where the subgraph with maximum $r(G')$ value is selected to update the prototype vector $\mathbf{p}_k^m$ through $\mathbf{p}_k^m \leftarrow f_\theta(G')_{\text{best}}$, effectively aligning prototypes with the most representative fault patterns.

## C. Multi-Task Learning Architecture

Given a training dataset $\{\mathcal{G}_n, y_n^{\text{node}}, y_n^{\text{device}}\}_{n=1}^N$ containing $n$ labeled network snapshots, we formulate two complementary learning objectives:

- **Node-Level Fault Detection**: For each graph $\mathcal{G}_n$, the label $y_n^{\text{node}} \in \{0, 1, \ldots, N_{\text{nodes}} - 1, N_{\text{nodes}}\}$ identifies faulty nodes, where:

$$y_n^{\text{node}} = \begin{cases} k & \text{(Node } v_k \text{ exhibits abnormality)},\\ N_{\text{nodes}} & \text{(No fault detected in } \mathcal{G}_n\text{)}. \end{cases}$$

  Here $N_{\text{nodes}}$ denotes the total number of optical link nodes in the network.

- **Device-Level Fault Detection**: The label $y_n^{\text{device}} \in \{0, 1, \ldots, N_{\text{devices}} - 1, N_{\text{devices}}\}$ specifies the malfunctioning

component type (with predefined device-type mappings, e.g., 0: Fiber, 1: EDFA). The label is defined as:

$$y_n^{\text{device}} = \begin{cases} k & \text{(Type-}k\text{ device fault)}, \\ N_{\text{devices}} & \text{(No device-level fault)}. \end{cases}$$

We design two distinct prototype layers: 1) a node-oriented layer $\mathcal{P}_{\text{node}} \in \mathbb{R}^{((N_{\text{nodes}}+1)\times M)\times d_e}$ for topological fault localization, and 2) a device-oriented layer $\mathcal{P}_{\text{device}} \in \mathbb{R}^{((N_{\text{devices}}+1)\times M)\times d_e}$ for device-level detection, where $M = 1$ is implemented for both detection tasks in practical application. Notably, the device-oriented prototypes adapt the MCTS-based Feature-type Rollout mechanism, where instead of pruning network nodes, we iteratively remove feature dimensions (e.g., Power, ASE noise, nonlinear impairment metrics) while preserving other phases.

**Algorithm 1.** Multi-Task Graph Prototype Learning(MT-GPL)

**Require:** Dataset $\mathcal{D}$, number of tasks $T = 2$, learning rate $\eta$, training epoch $E$
**Ensure:** Trained model parameters $\theta$

```
Initialize model parameters θ with shared and task-specific layers
Initialize prototype sets P_node, P_device for two task, respectively
for E = 1 to E_max do
    Prototype Projection Phase (every 10 epochs):
    if E ≥ E_warm and epoch ≡ 0 (mod 10) then
        for each task t ∈ {1,2} do
            Select candidate samples S_k^m via M-medoids
            Update prototypes P_node(P_device) using subgraph search(feature sampling)
    Multi-Task Training Phase:
    for each batch B ⊂ D do
        Compute task losses in Eq. 6 and Eq. 13
        Gradient Coordination:
        Compute task gradients in Eq. 14
        Parameter Update:
        θ ← θ − η · g
```

### D. Gradient-Coordinated Uncertainty Optimization for Multi-Task Learning

The learning framework integrates multi-task optimization through homoscedastic uncertainty weighting [28] and PCGrad gradient conflict resolution [29]. The composite loss function is defined as:

$$\mathcal{L}(W) = \frac{1}{2\sigma_1^2}\mathcal{L}_{\text{node}} + \frac{1}{2\sigma_2^2}\mathcal{L}_{\text{device}} + \log \sigma_1\sigma_2, \tag{13}$$

where $\mathcal{L}_{\text{node}}$ and $\mathcal{L}_{\text{device}}$ denote fault detection and localization losses computed via Eq. 6. The noise parameters $\sigma_1, \sigma_2$ adaptively reweight tasks based on uncertainty.

Task-specific gradients are processed through PCGrad to resolve conflicts: For shared parameters $\theta_{\text{shared}}$ (e.g., GNN layers), projected gradients are computed as:

$$\begin{aligned} &\text{Projection:} \quad g_c' = g_c - \frac{\langle g_{c'}, g_c\rangle}{\|g_{c'}\|^2} g_{c'} \quad \text{if } \langle g_{c'}, g_c\rangle < 0, \\ &\text{Merging:} \quad g_{\text{shared}} = \sum_{c=1}^{2} \omega_c \cdot g_c', \quad \omega_c = \frac{\mathcal{L}_c}{\mathcal{L}_1 + \mathcal{L}_2}, \end{aligned} \tag{14}$$

where $g_c, g_{c'}$ are task gradients and $\omega_c$ are dynamic weights based on loss magnitudes. Task-specific parameters $\theta_{\text{task}}$ (e.g., prototype vectors) retain original gradients without projection.

The optimization alternates between: 1) Forward-backward passes to compute task losses, 2) PCGrad-based gradient projection for shared parameters, and 3) Parameter updates with the Adam optimizer. This dual mechanism ensures conflict-free feature learning in shared layers while preserving task-specific prototype discriminability.

Furthermore, given the higher complexity of Task 1 compared to Task 2, the network components for Task 1 – including the shared GCN-based encoder and Task 1's prototype layer $\mathcal{P}_{node}$ – undergo dedicated pre-training for 100 epochs before joint optimization. This phased initialization ensures stable feature learning for the more complex task. The complete workflow of Multi-Task Graph Prototype Learning is formalized in Algorithm 1, where the prototype vectors are dynamically updated through our subgraph search and feature-space sampling mechanism.

## 5. PERFORMANCE EVALUATION

### A. Simulation Setup for Data Collection

We evaluated the performance of our interpretable fault detection and localization design using data collected by the GNPy platform [30] following a six-node configuration shown in Fig. 1(a). The parameter configurations for the six-node topology were determined in reference to the 'German_Topology' setup in GNPy. The fiber length follows the actual configuration in the GNPy toolkit. For example, the fiber between nodes Bremen and Hamburg is 114.764 km. We established 16 lightpaths with four wavelengths as detailed in Table. 1. The lightpaths operated at 12.5 GBaud with 16-QAM in dual polarizations, maintaining a net data rate of 60 Gbps. Transceiver output power, OSNR, power fluctuations across channels, fiber nonlinear coefficient $\gamma$ and chromatic dispersion coefficient $D$ were set as 3.0 dBm, 20 dB, 0.0 dB, 0.001270 $\text{W}^{-1}\cdot\text{m}^{-1}$, and 16.7 $\text{ps}/(\text{nm}\cdot\text{km})$, respectively. Four ITU-T G.694.1 compliant wavelengths were allocated at 193.20 THz ($\lambda_1$), 193.25 THz ($\lambda_2$), 193.30 THz ($\lambda_3$), and 193.35 THz ($\lambda_4$) in the C-band.

**Table 1.** Lightpath configurations in the six-node topology.

| Routing | Wavelength | Routing | Wavelength |
|---|---|---|---|
| $R_5 \to R_1 \to R_2 \to R_0$ | $\lambda_1$ | $R_1 \to R_5$ | $\lambda_4$ |
| $R_0 \to R_2 \to R_1 \to R_5$ | $\lambda_1$ | $R_5 \to R_1$ | $\lambda_4$ |
| $R_0 \to R_5 \to R_4 \to R_3$ | $\lambda_1$ | $R_2 \to R_4$ | $\lambda_2$ |
| $R_0 \to R_3 \to R_4 \to R_5$ | $\lambda_1$ | $R_4 \to R_2$ | $\lambda_2$ |
| $R_4 \to R_2 \to R_1$ | $\lambda_3$ | $R_1 \to R_3$ | $\lambda_2$ |
| $R_1 \to R_5 \to R_4$ | $\lambda_3$ | $R_3 \to R_1$ | $\lambda_2$ |
| $R_2 \to R_0 \to R_3$ | $\lambda_4$ | $R_3 \to R_0$ | $\lambda_3$ |
| $R_3 \to R_1 \to R_2$ | $\lambda_4$ | $R_0 \to R_3$ | $\lambda_3$ |

Fiber impairments were modeled with attenuation coefficients $\alpha \in [0.2, 0.3]$ dB/km for normal operations and $\alpha \in [0.3, 0.5]$ dB/km for faulty conditions. Amplifier faults were emulated through first-stage noise figure ($\text{NF}_1$) degradation,

$$\text{NF}_1 = \begin{cases} 7.0\,\text{dB} \le \text{NF}_1 \le 10.0\,\text{dB} & \text{(Faulty)}, \\ 5.0\,\text{dB} \le \text{NF}_1 \le 7.0\,\text{dB} & \text{(Normal)}. \end{cases} \tag{15}$$

For variable-gain amplifiers, the effective noise figure is calculated as,

$$\begin{aligned} \mathrm{NF}_{\mathrm{avg}} &= \mathrm{lin2db}\left(\mathrm{db2lin}(\mathrm{NF}_1) + \frac{\mathrm{db2lin}(\mathrm{NF}_2)}{\mathrm{db2lin}(G_{1a})}\right), \\ G_{1a} &= G_{\mathrm{tgt}} - \Delta P - \Delta G, \\ \mathrm{NF} &= \mathrm{NF}_{\mathrm{avg}} + \max(G_{\mathrm{min}} - G_{\mathrm{tgt}}, 0) + \mathrm{NF}_{\mathrm{ripple}}, \end{aligned} \tag{16}$$

where $G_{\mathrm{tgt}}$ and $G_{\mathrm{min}}$ denote the target and minimum gains set by the EDFA, respectively, and $\Delta P/G$ represents the power/gain offset. The ASE noise power spectral density can be further expressed by,

$$P_{\mathrm{ASE}} = h \cdot B \cdot f_{\mathrm{opt}} \cdot 10^{\frac{\mathrm{NF}}{10}}, \tag{17}$$

where $h$ is the Planck constant ($h = 6.626 \times 10^{-34}$ J $\cdot$ s), $B$ is the baud rate in Hz, $f_{\mathrm{opt}}$ is the optical carrier frequency in Hz ($f_{\mathrm{opt}} = c/\lambda$, with $c = 3 \times 10^8$ m/s).

Overall, we constructed a dataset containing 7,000 optical transmission samples: 6,000 fault instances and 1,000 normal operation records that emulate normal operations with random perturbations introduced by fluctuating the parameters within the normal ranges. The fault samples were equally distributed between two failure modes. Specifically, 3,000 samples exhibit abnormal fiber attenuation coefficients while the other 3,000 samples demonstrate abnormal EDFA noise figures, with all the faults distributed across the 16 fiber links.

### B. Algorithm Configuration and Benchmarks

For our proposed algorithm, we set the hyperparameter $\gamma$ in Eq. 6 as 0.0001. Both task 1 (node-level prediction) and task 2 (device-level prediction) employ a single prototype per fault class. In MCTS for prototype projection, the number of iterations was set to 10. For prototype search for task 1, each vertex in the Monte Carlo tree can expand up to three child vertices with $N_{\mathrm{min}} = 4$, whereas for task 2, each tree vertex can expand up to two child vertices with $N_{\mathrm{min}} = 1$.

We compared our multi-task graph prototype learning approach (denoted as MT-GPL) with three benchmark designs, namely, GCN, GPL, and multi-layer perceptron (MLP). Specifically, GCN adopts the same architecture for graph embedding as in our design (*i.e.*, $f_\theta(\cdot)$) without prototype layers, GPL adds prototype layers but trains a single neural network head for direct fault localization over all the devices in the network, while MLP adopts a simple five-layer neural network structure, each with 128 neurons. For all the algorithms, we split the data set into training, validation, and testing sets with a ratio of 0.8 : 0.1 : 0.1 and performed training for 1,000 epochs using the Adam optimizer (with a learning rate of 0.0007). An early stopping strategy based on the validation accuracy was also adopted to prevent overfitting.

**Table 2.** Comparisons of classification accuracy.

| | MLP | GCN | GPL | MT-GPL |
|---|---|---|---|---|
| Task 1 | 0.992 | 0.995 | 0.994 | 0.999 |
| Task 2 | 0.993 | 0.996 | 0.999 | 0.997 |

**Table 3.** Comparisons under modified EDFA fault boundaries.

| | MLP | GCN | GPL | MT-GPL |
|---|---|---|---|---|
| Task 1 | 0.889 | 0.977 | 0.979 | 0.954 |
| Task 2 | 0.902 | 0.982 | 0.986 | 0.972 |

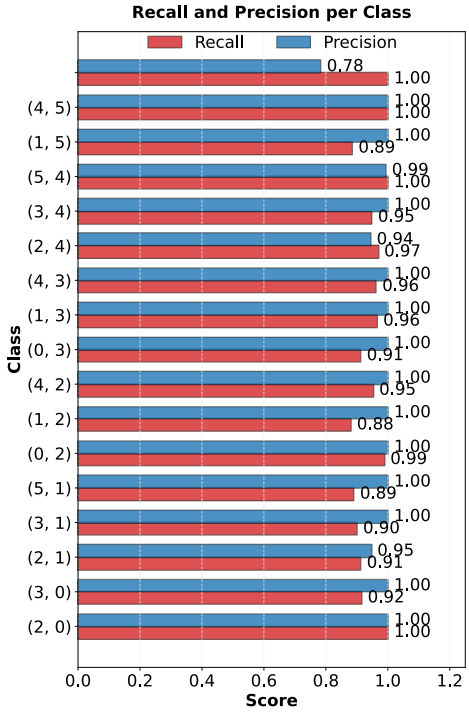


**Fig. 2.** Recall and precision results from our algorithm for all the fault classes.

### C. Comparison with the Benchmarks

Table 2 shows the results of classification accuracy from different algorithms. It can be seen that all the algorithms achieve accuracies of $> 99\%$ for both tasks, indicating that they learn successful features to discriminate normal and faulty data while pinpointing the root causes. In this case, our approach performs equally well to its counterparts, and hence, the distinctions mainly lie in model interpretability, which we will discuss later. Next, to evaluate the performance of the algorithms in handling more complex tasks, we deliberately blurred the fault boundaries. In particular, although the first-stage EDFA noise figure $\mathrm{NF}_1$ typically ranges between $5-7$ dB in practical engineering, our six-node topology simulation environment exhibits $\mathrm{NF}_1 \in [6.0, 6.3)$ dB under normal operations. Thus, we defined anomalies as $\mathrm{NF}_1 \in [6.3, 8.0]$ dB instead of the range given by Eq. 15. The results after boundary modification are shown in Table 3. We can see that in this case, MLP can no longer provide accurate predictions, whereas the three GCN-based designs still achieve desirable accuracies. Our algorithm excels MLP by $> 6.5\%$ and $> 7.0\%$ accuracy on tasks 1 and 2, respectively, but slightly underperforms GCN and GPL. This can be attributed to the error accumulation effect from our hierarchical multi-task arrangement. We will provide more intrinsical interpretation to this performance difference in Section 5. D.3. However, we should note that the multi-task design allows for better scalability by requiring much fewer trainable parameters for classification. Specifically, our algorithm learns 20 prototypes (17 for task 1

and 3 for task 2), but GPL needs to maintain 33 prototypes. Consequently, the numbers of trainable parameters from our design and GPL are $17 \times 17 + 3 \times 3 = 298$ and $33 \times 33 = 1,089$, respectively. This gap will further expand when we consider a larger topology and more abundant failure scenarios.

To further examine our algorithm's predictions for the results in Table 3, we calculate class-specific recall and precision metrics for task 1 and plot the results in Fig. 2. Here, recall and precision are defined as,

$$\text{Recall} = \frac{TP}{TP+FN}, \quad \text{Precision} = \frac{TP}{TP+FP}, \tag{18}$$

where true positives (TP) are the number of correctly detected fault samples, false negatives (FN) are the number of undetected true faults, and false positives (FP) are the number of normal samples mistakenly classified as faults. In the figure, we enumerate all the fault positions in our data set, for instance item (0,3) represents the case where failures occur on the link from ROADM $R_0$ to $R_3$. We can see that for all the fault classes, the proposed design achieves $\geq 95\%$ recall and nearly 100% precision (close to zero false alarms).

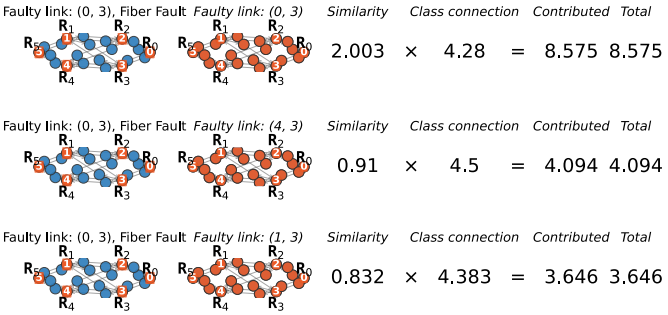


**Fig. 3.** Visualization of prototype-based reasoning in detecting a fault between ROADMs $R_0$ and $R_3$. Prototypes for task 1 focus on the graph structures.

## D. Interpretability Analysis

Next, we analyzed the interpretability of our design through visualizations of its prototype-based reasoning and learned embeddings.

### *D.1. Visualization of Prototype-based Reasoning for Task 1*

Both GPL and MT-GPL enable direct interpretability of classification decisions through their prototype-based architectures. Fig. 3 interprets MT-GPL's decisions by visualizing the prototypes and the learned embedding in predictions. The input data represents a fault on the fiber link between ROADM $R_0$ and $R_3$ (first column). The second column displays MT-GPL's fiber link fault prototypes targeting distinct fault locations, *i.e.*, faults on $0 \to 3$, $2 \to 3$, and $4 \to 3$, which were generated by applying the GCN encoder $f_\theta(\cdot)$ outputs of raw training data. The third column quantifies the similarities between the input data's embeddings and the prototypes via Eq. 4. The fourth column lists the fully-connected layer weights, and the fifth column calculates prototype contributions. The prototype for the class associated with a fault on link $0 \to 3$ makes the highest contribution (*i.e.*, 8.575), and consequently assists in correct localization of the fault. The results demonstrate that prototype-driven attributions align well with physical topology.

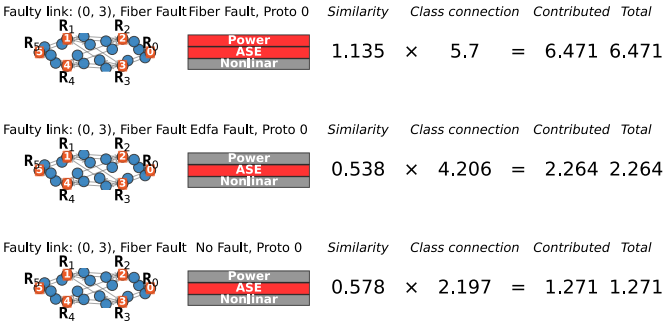


**Fig. 4.** Visualization of prototype-based reasoning in locating a fiber fault. Prototypes for task 2 focus on the decisive features.

### *D.2. Visualization of Prototype-based Reasoning for Task 2*

While GPL maintains equivalent interpretability across the two tasks, MT-GPL provides enhanced explanatory capabilities through its multi-task prototype alignment mechanism, particularly enabling detailed failure component analysis in task 2 via disentangled feature representations. As shown in Fig. 4, the feature contribution analysis of the prototype layer reveals discriminant mechanisms for different failure modes. The red-highlighted features define the core decision criteria for the three prototypes: the fiber fault prototype depends on joint analysis of ASE noise and optical power features; in contrast, the EDFA fault prototype relies solely on ASE characteristics; the normal operation prototype also depends on ASE characteristics. Finally, MT-GPL determines the classification result by identifying the maximum similarity score and correctly detects the fiber fault due to the highest similarity score of 1.135 between the projected embedding and the fiber fault prototype. Notably, ASE features maintain significant contribution values across all prototypes, which aligns precisely with the physical principle. According to Eq. 17, when the baud rate and optical carrier frequency are fixed, EDFA-induced ASE is exclusively determined by the noise figure, while fiber attenuation coefficient variations simultaneously alter both optical power and ASE features, which is accurately captured by the prototype layer's feature coupling analysis.

### *D.3. Visualization of Prototype Distribution*

Fig. 5 presents t-SNE visualizations [31] of graph embeddings and their corresponding prototypes for task 1 and task 2, respectively. Here, for the sake of clarity, we only show the prototypes for faults on three links. The plots show that distinct fault types form tightly clustered distributions in the latent space, with prototype vectors (marked as red pentagrams) positioned in the central regions of their respective fault clusters. This geometric configuration confirms that the learned prototypes effectively capture the patterns of each failure mode. Furthermore, the distributions demonstrate that using 6 unified prototypes may achieve stronger feature regularity compared with the multi-task setting that produces 3 and 2 prototypes individually. This intrinsically explains the slight performance degradation by MT-GPL in Table 3. Again, although GPL achieves stronger pattern consistency, MT-GPL's hierarchical arrangement offers enhanced interpretability and scalability. Meanwhile, the visualizations reveal distinct cluster geometries, with fiber fault data exhibiting cluster-like distributions, whereas EDFA faults present linear or band-like structures. This suggests EDFA faults manifest approximately continuous variations along a dominant feature dimension, while fiber faults demonstrate discrete multi-modal

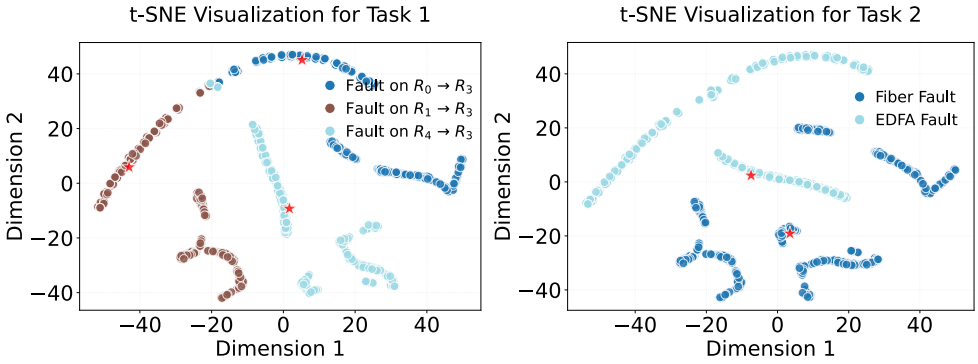


**Fig. 5.** Distributions of the projected embbeding for (a) task 1 and (b) task 2. Colored dots represent the t-SNE visualizations for the embeddings of different fault classes. The corresponding prototypes are marked by red pentagrams.

patterns with heterogeneous feature interactions, which is in line with the observations drawn from Fig. 4.

### E. Scalability and Generalization Ability Studies

Finally, we experimented the models in a larger topology (17-node, 52-link complete German topology) considering broader fault scenarios to test their scalability and generalization ability. In particular, we configured 23 lightpaths and introduced also connector attenuation faults (*i.e.*, attenuation of $(0.5, 1.5]$ dB) in addition to fiber and EDFA faults. Following similar setups as in the previous evaluations, a total of $20,800$ samples were collected, with each of the EDFA, fiber and connector faults and the normal class accounting for $1/4$ of the data. The accuracy comparisons of the models are presented in Table. 4. We can see that the GCN-based models still perform comparably but excels MLP by larger margins (over 8% and 3% on tasks 1 and 2, respectively) in this larger topology setting. In line with the results in Tables 2 and 3, incorporating interpretability and multi-task learning does not evidently compromise the model performance. Due to the more complex topology structure and more failure scenarios introduced, the accuracy of MT-GPL for task 1 drops slightly to 94.4%. To shed light on this performance disparity, we plotted its recall and precision results with respect to 17 (out of 53) representative classes for task 1 in Fig. 6. Notably, the model undergoes significant declines in recall for certain links, such as links $2 \rightarrow 4$ and $9 \rightarrow 3$. The results suggest that the model fails to capture accurate topological characteristics for diagnosing the faults on these links. We presume this is due to the insufficient coverage and correlation that the limited set of lightpaths can provide. For instance, a fiber fault on $1 \rightarrow 2$ may not be correctly identified with data from two lightpaths $1 \rightarrow 2 \rightarrow 3$ and $4 \rightarrow 5 \rightarrow 0 \rightarrow 1$ if the pattern of $0 \rightarrow 1$ resemble that of $1 \rightarrow 2$ due to the cumulative loss over a longer path, while an additional lightpath $0 \rightarrow 1 \rightarrow 2$ can assist in pinpointing the fault. Nevertheless, determining the right set of lightpaths allowing for accurate fault management remains a challenging task.

**Table 4.** Comparisons of classification accuracy under the 17-node German Topology.

| | MLP | GCN | GPL | MT-GPL |
|---|---|---|---|---|
| Task 1 | 0.863 | 0.957 | 0.955 | 0.944 |
| Task 2 | 0.967 | 0.998 | 0.998 | 0.999 |

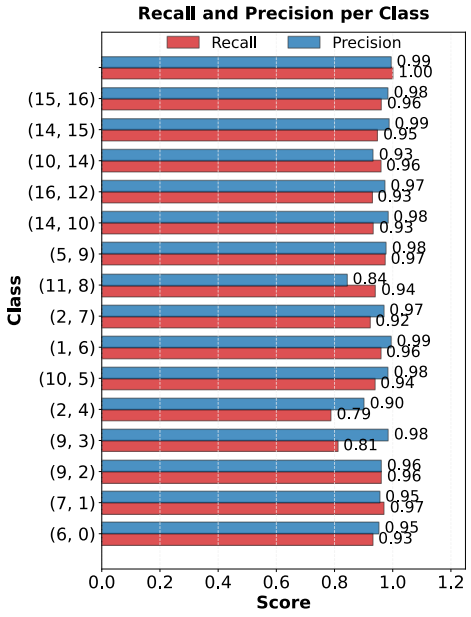


**Fig. 6.** Recall and precision results for 17 selected link-fault classes evaluated under the 17-node German topology.

## 6. CONCLUSION

In this paper, we proposed an interpretable fault detection and localization design named MT-GPL for optical networks. MT-GPL processes OPM data as graph samples and learns embeddings that capture both topological correlations for fault localization and fault discriminative patterns for pinpointing the root causes. MT-GPL interprets its reasoning by identifying physics-aligned prototypes for each fault class and comparing the similarity between an input graph's embedding and the learned prototypes. Simulation results verify the effectiveness of MT-GPL.

Potential future research directions include: *1*) studying how the configuration of lightpaths (number and routing paths) would affect the accuracy of our GNN-based design and how to optimize this configuration given an optical network topology; *2*) improving the scalability of our approach to ultra-scale networks, for instance, by optimizing the time complexity of MCTS leveraging optimized pruning heuristics [32], neural network ensemble integration [33], or post-hoc Grad-CAM [34] masking strategies; and *3*) refining prototype learning to minimize the adverse effect of interpretability on the raw performance, and investigating the trade-offs between accuracy and interpretability for mission-critical applications where predictive accuracy is

paramount.

## ACKNOWLEDGMENTS

This work was supported by the NSFC projects 62201627 and 62371432, and the Guangdong Program under Grant 2021QN02X039.

## REFERENCES

1. Ericsson, "Ericsson Mobility Report June 2024," Tech. rep., Ericsson, Stockholm, Sweden (2024).
2. A. Vela, B. Shariati, M. Ruiz, F. Cugini, A. Castro, H. Lu, R. Proietti, J. Comellas, P. Castoldi, S. J. B. Yoo, and L. Velasco, "Soft failure localization during commissioning testing and lightpath operation," J. Opt. Commun. Netw. **10**, A27–A36 (2018).
3. D. Rafique, T. Szyrkowiec, H. Grießer, A. Autenrieth, and J. Elbers, "Cognitive assurance architecture for optical network fault management," J. Light. Technol. **36**, 1443–1450 (2018).
4. S. Shahkarami, F. Musumeci, F. Cugini, and M. Tornatore, "Machine-learning-based soft-failure detection and identification in optical networks," in *Proc. Conf. Opt. Fiber Commun.*, (2018). Paper M3A.5.
5. M. Furdek, C. Natalino, A. Di Giglio, and M. Schiano, "Optical network security management: requirements, architecture, and efficient machine learning models for detection of evolving threats [invited]," J. Opt. Commun. Netw. **13**, A144–A155 (2021).
6. X. Chen, C. Liu, R. Proietti, Z. Li, and S. J. B. Yoo, "Automating optical network fault management with machine learning," IEEE Commun. Mag. **60**, 88–94 (2022).
7. S. Liu, D. Wang, C. Zhang, L. Wang, and M. Zhang, "Semi-supervised anomaly detection with imbalanced data for failure detection in optical networks," in *Proc. Conf. Opt. Fiber Commun.*, (2021). Paper Th1A.24.
8. H. Lun, X. Liu, M. Cai, Y. Wu, M. Fu, L. Yi, W. Hu, and Q. Zhuge, "GAN based soft failure detection and identification for long-haul coherent transmission systems," in *Proc. Conf. Opt. Fiber Commun.*, (2021). Paper Th4J.2.
9. L. Shu, Z. Yu, Z. Wan, J. Zhang, S. Hu, and K. Xu, "Dual-stage soft failure detection and identification for low-margin elastic optical network by exploiting digital spectrum information," J. Light. Technol. **38**, 2669–2679 (2020).
10. H. Lun, X. Liu, M. Cai, M. Fu, Y. Wu, L. Yi, W. Hu, and Q. Zhuge, "Anomaly localization in optical transmissions based on receiver DSP and artificial neural network," in *Proc. Conf. Opt. Fiber Commun.*, (2020). Paper W1K.4.
11. F. N. Khan, "Non-technological barriers: the last frontier towards AI-powered intelligent optical networks," Nat. Commun. **15**, 5995 (2024).
12. O. Ayoub, S. Troia, D. Andreoletti, A. Bianco, M. Tornatore, S. Giordano, and C. Rottondi, "Towards explainable artificial intelligence in optical networks: the use case of lightpath qot estimation," J. Opt. Commun. Netw. **15**, A26–A38 (2023).
13. O. Karandin, O. Ayoub, F. Musumeci, Y. Hirota, Y. Awaji, and M. Tornatore, "If not here, there. explaining machine learning models for fault localization in optical networks," in *Proc. Int. Conf. Opt. Netw. Design Model.*, (2022), pp. 1–3.
14. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Adv. Neural Inf. Process. Syst. **30** (2017).
15. Z. Li, Y. Zhao, Y. Li, S. Rahman, F. Wang, X. Xin, and J. Zhang, "Fault localization based on knowledge graph in software-defined optical networks," J. Light. Technol. **39**, 4236–4246 (2021).
16. D. Wang, M. Zhang, Z. Zhang, J. Li, H. Gao, F. Zhang, and X. Chen, "Machine learning-based multifunctional optical spectrum analysis technique," IEEE Access **7**, 19726–19737 (2019).
17. S. Ghosh and A. Adhya, "Soft failure detection and identification in optical networks using cascaded deep learning model," Authorea Prepr. (2024).
18. Q. Lin, X. Chen, Z. Ouyang, H. Gao, X. Chen, and Z. Li, "Scaling Optical Network Fault Management with Decentralized Graph Learning," in *Proc. Conf. Opt. Fiber Commun.*, (2024), pp. 1–3.
19. Z. Li, Y. Zhao, Y. Li, S. Rahman, X. Yu, and J. Zhang, "Demonstration of fault localization in optical networks based on knowledge graph and graph neural network," in *Proc. Conf. Opt. Fiber Commun.*, (IEEE, 2020), pp. 1–3.
20. C. Zhang, D. Wang, J. Jia, L. Wang, K. Chen, L. Guan, Z. Liu, Z. Zhang, X. Chen, and M. Zhang, "Potential failure cause identification for optical networks using deep learning with an attention mechanism," J. opt. Commun. Netw. **14**, A122–A133 (2022).
21. C. Zeng, J. Zhang, R. Wang, B. Zhang, and Y. Ji, "Multiple attention mechanisms-driven component fault location in optical networks with network-wide monitoring data," J. Opt. Commun. Netw. **15**, C9 (2023).
22. D. Wang, C. Zhang, W. Chen, H. Yang, M. Zhang, and A. P. T. Lau, "A review of machine learning-based failure management in optical networks," Sci. China Inf. Sci. **65**, 211302 (2022).
23. C. Zhang, D. Wang, L. Wang, L. Guan, H. Yang, Z. Zhang, X. Chen, and M. Zhang, "Cause-aware failure detection using an interpretable xgboost for optical networks," Opt. Express **29**, 31974–31992 (2021).
24. G. Theodorou, S. Karagiorgou, A. Fulignoli, and R. Magri, "On explaining and reasoning about optical fiber link problems," in *World Conference on Explainable Artificial Intelligence,* (Springer, 2024), pp. 268–289.
25. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907 (2016).
26. O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," (2018).
27. L. Rdusseeun and P. Kaufman, "Clustering by means of medoids," in *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland,* vol. 31 (1987), p. 28.
28. A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* (2018), pp. 7482–7491.
29. F. Meng, Z. Xiao, Y. Zhang, and J. Wang, "Ri-pcgrad: Optimizing multi-task learning with rescaling and impartial projecting conflict gradients," Appl. Intell. **54**, 12009–12019 (2024).
30. V. Curri, "Gnpy model of the physical layer for open and disaggregated optical networking," J. Opt. Commun. Netw. **14**, C92–C104 (2022).
31. L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." J. machine learning research **9** (2008).
32. X. Ma, K. Driggs-Campbell, Z. Zhang, and M. J. Kochenderfer, "Monte-carlo tree search for policy optimization," arXiv preprint arXiv:1912.10648 (2019).
33. Z. Xing and S. Tu, "A graph neural network assisted monte carlo tree search approach to traveling salesman problem," Ieee Access **8**, 108418–108428 (2020).
34. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision,* (2017), pp. 618–626.