# Cross-Task and Cross-Lightpath Failure Detection and Localization in Optical Networks using Transfer Learning

Francesco Musumeci*, Giacomo G. Marchionni, Massimo Tornatore
Politecnico di Milano, Milan, Italy. *Corresponding author: francesco.musumeci@polimi.it

***Abstract*—Practical deployments of Machine-Learning(ML)-based solutions for failure management in optical networks often suffer from limited data availability, due to, especially, scarcity of labelled data describing different failure scenarios. Transfer Learning (TL) is regarded as a promising direction in cases of data scarcity, thanks to its ability to transfer knowledge from a Source Domain (SD) (e.g, SD could be a digital twin or a laboratory testbed) to a Target Domain (TD) (e.g., the in-field network). In this paper, we focus on cross-lightpath and cross-task application of TL for failure localization and failure detection in optical networks. We found that, depending on the number of retrained parameters in the ML model, cross-lightpath TL for failure localization provides satisfactory accuracy (higher than 90%, in some cases) with limited amounts of TD data, and is also convenient in terms of TD retraining duration with respect to cases where TL is not used. Moreover, we found that cross-task failure detection/localization reaches up to 12% or 25% improvement in TD accuracy when considering failure localization and detection as TD task, respectively.**

***Index Terms*—Transfer learning, failure detection and localization, optical networks, OSNR**

## I. Introduction

During lightpath lifetime, it is fundamental to guarantee proper transmission quality and, if needed, quick lightpath restoration to comply with the stringent availability requirements of 5G and beyond services supported by modern optical networks. In most situations, when dealing with *soft-failures* (that in this paper we simply refer to as *failures*), consisting of gradual optical signal degradation until unacceptable quality threshold, continuous monitoring of transmission quality parameters, such as Bit Error Rate (BER), Optical Signal to Noise Ratio (OSNR) etc.. Machine Learning (ML) has been already demonstrated to be an effective tool to extract useful information from field network data to address different failure-management tasks, such as failure detection, identification, localization and magnitude estimation [1]. These ML-based failure management solutions have also been applied to different types of failures, e.g., involving filters and/or amplifiers malfunctioning [1], [2], as well as to cyberthreats at the optical layer [3].

However, in practical deployments, since ML-assisted optical network failure management is usually modeled as a supervised ML problem, it is often difficult to collect sufficient failure labeled data for different failure conditions (e.g., type, location, severity, etc.) to train ML models. For this reason, Transfer Learning (TL) is being investigated to address data scarcity in failure management problems. TL consists of training a ML model using data retrieved from a source domain (SD) and applying (i.e., *transferring*) this knowledge to a target domain (TD), i.e., the network domain where the model is deployed, where a limited amount of data is available and is used to fine-tune the original model. In the context of failure management in optical networks, the SD can be, e.g., an emulated *digital twin*, a laboratory testbed, or even a network segment dedicated to data collection, where failures are purposely injected with to generate a sufficiently-large training dataset. As an example, in a previous work [4], we have trained a failure-identification model on a SD lightpath with certain characteristics (e.g., in terms of number of links, number and type of devices, etc.) and adopted TL to fine-tune the model with a limited amount of data coming from the TD lightpath with different characteristics. In this paper we refer to this scenario as cross-lightpath TL.

TL can also be beneficial in other scenarios where knowledge transfer is performed between different tasks, i.e., different problems (note that in this case, SD and TD use datasets might not be necessarily distinct). For example, a ML classifier can be trained for failure detection, i.e., to assess whether a lightpath is affected by a failure, representing the SD (or *source task*), and then fine-tuned with a limited amount of failure-location data to perform failure localization, that represents the TD (or *target task*). In this paper we refer to this scenario as cross-task TL. Cross-task TL may become useful, e.g., when only part of the data is available also with labels used for the target task (e.g., information on failure location), while most data is labeled only with source task labels (e.g., information on absence/presence of failure). Note that the opposite scenario, i.e., performing TL from failure localization to failure detection, is also possible, and it is expected to have good transferability, as information on failure localization also includes information on failure detection[1].

### A. Related Work

Different applications of TL in optical networking have been investigated, mostly focusing on Quality of Transmission (QoT) estimation. In [5], authors train Artificial Neural Networks (ANNs) using data from two different topologies and reuse the models on a different topology to estimate

[1]Note that, even though a failure localization model can be also used, as is, to perform failure detection, the failure localization model might not have the same accuracy of a failure detection model. So transfer learning from localization to detection is still meaningful.

the generalized optical-signal-to-noise-ratio (G-OSNR) for unestablished lightpaths. A similar approach has been used also in [6], where authors estimate QoT of unestablished lightpaths with the objective of reducing the G-SNR uncertainty and hence the design margins. The work in [7] proposes a genetic algorithm that optimizes the ANN architecture and the set of weights to be transferred when applying TL to multi-domain elastic optical networks, considering lightpaths with different characteristics in terms of path length, modulation format and devices conditions. The effectiveness of TL for QoT estimation is also evaluated in [8] and compared with active learning, an approach that addresses data scarcity by providing information on the portions of the features space where collecting new data is more informative to improve model performance.

Considering other optical-networking applications, TL is used in [9] to transfer knowledge between different ML algorithms, namely, Recurrent and Feed-forward Neural Networks, to address nonlinear equalization in short-reach optical links. Authors of [10] adopt TL for impairments mitigation in long-haul coherent optical transmission systems with different characteristics, including launch power, modulation format, symbol rate, and fiber plants and considering different types of fibers. Ref. [11] uses TL to predict spectrum defragmentation and optimize resources utilization in space-division-multiplexed elastic optical networks, leveraging knowledge acquired on a simple source topology and applying the pre-trained model on a larger topology where a limited amount of data is available. To the best of our knowledge, the application of TL to failure management in optical networks is almost unexplored. In one of our previous work [4], we considered cross lightpaths TL for failure identification across lightpaths with different lengths, but no cross-lightpath TL for localization, and, more importantly, no cross-task TL were ever considered.

### B. Paper Contribution

For the first time to the best of our knowledge, we use real OSNR data collected at lightpaths' receiver of an optical network testbed, and evaluate TL for failure localization across different lightpaths, which mainly differ in their length and number/type of traversed devices. We also evaluate the impact of cross-task TL between failure localization and detection, i.e., we reuse ML models pre-trained to solve failure detection (respectively, localization) and apply TL to solve localization (resp., detection) on the same lightpath.

The paper is organized as follows. In Sec. II, we define the problem of ML-based failure detection and localization in optical networks. Sec. III describes the steps constituting the cross-task and cross-lightpaths failure detection and localization based on TL. Finally, we discuss numerical results in Sec. IV, and conclude the paper in Sec. V.

## II. ML-Based Failure Detection and Localization

### A. Problem Definition

We model the failure-detection and failure-localization problems in optical networks as two supervised classification problems. On the one hand, failure detection is a binary classification problem, where the two classes indicate the presence or absence of a failure along the observed lightpath. On the other hand, failure localization is a multi-class classification problem, where each class corresponds to a failure location, i.e., in this study, one of the fiber links traversed by the lightpath (also including the case with no failure along the lightpath). Hence, failure detection can be considered as a generalization of failure localization, where all classes indicating a failure on a specific link correspond to a same class (i.e., presence of failure in *any* link). [2]

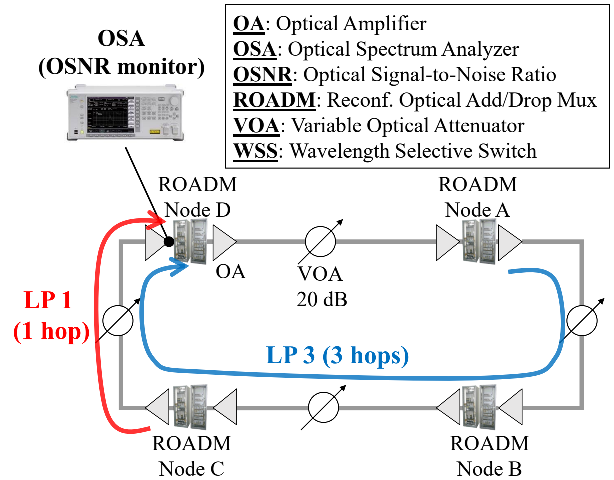


Fig. 1: Testbed setup.

In both problems, we concentrate on a given lightpath for which we are given the OSNR at the receiver, monitored with sampling period of $T_{OSNR}$ seconds (e.g., $T_{OSNR} = 1$). Classification is performed considering OSNR "windows" of duration $W$ seconds, each one including a sequence of $1+W/T_{OSNR}$ consecutive OSNR observations (also including the OSNR samples at the start and end times of the window).

### B. Testbed description

We consider TL-based failure detection and localization using real data obtained on a testbed of the National Institute of Information and Communications Technology (NICT) located in Sendai, Japan. The testbed is shown in Fig. 1 and consists of 4 ROADMs, identified as Nodes A, B, C and D, interconnected through optical fibers, and equipped with one pre-amplifier and one booster (OA in the figure) at their input and output, respectively, whose gain is set to recover from link or node loss. Each fiber link can emulate fiber spans of up to 80 km using a Variable Optical Attenuator (VOA) with up to 20 dB attenuation. Failure scenarios are mimicked by enforcing an extra-attenuation of 11 dB in one of the traversed links by means of a Wavelength Selective Switch (WSS). Figure 1 also shows the two lightpaths considered in our dataset, namely LP1 and LP3, traversing 1 and 3 links, respectively, both terminated at ROADM node D, where an OSNR monitor is installed after node pre-amplifier for OSNR data collection.

[2] As we consider only failures occurring in fiber links (i.e., not in nodes), failure detection and localization coincide when the observed lightpath traverses only one link, although the two models are not necessarily the same.

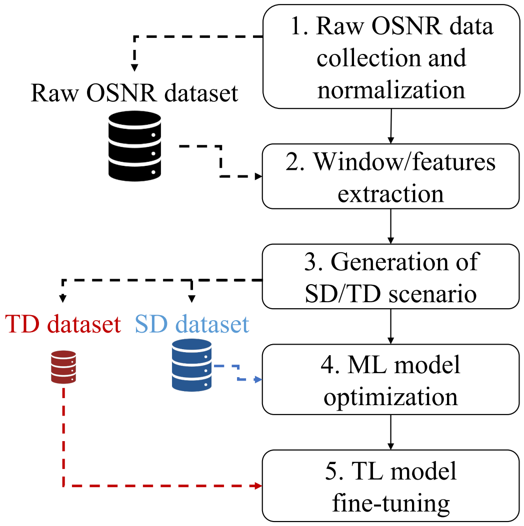



Fig. 2: Block diagram of the proposed TL framework for failure detection and localization.

To perform data collection, each lightpath is set-up in the testbed separately, i.e., no other lightpath is simultaneously set-up in the testbed. For each lightpath, the same 100 GHz bandwidth with central frequency 194.8 THz is used to transmit a 10 Gbit/s signal using OOK modulation format.

## III. TL Framework for Cross-Task and Cross-Lightpath Failure Detection and Localization

In this section we describe the framework used to perform cross-task and cross-lightpath TL for failure detection and localization. Figure 2 shows the main steps of this framework, which are detailed in the following subsections.

### A. OSNR Data Collection and Normalization

To generate the initial dataset, raw OSNR data is collected for two lightpaths, i.e., LP1 and LP3 shown in Fig. 1.

For each lightpath, besides the normal case of operation with no failure, we consider different failure scenarios (i.e., failure classes) where extra-attenuation failure is induced in one of the links traversed by the lightpath. For each scenario, either normal or failure, OSNR data is collected for a total duration of 6 hours, at a sampling period of $T_{OSNR} = 1$ second, so the entire dataset consists of around 36 hours of OSNR monitoring at lightpaths receivers.

After data collection, raw OSNR data is processed and, for each lightpath, OSNR samples are standardized to obtain OSNR distributions with 0 mean and unit standard deviation. This operation is performed as we are more interested in the relative behaviour of lightpaths' OSNR rather than its specific absolute values, that may be very different also for lightpaths that have similar failure characteristics, due to the possible difference in lightpaths settings (also in non-failure states of operation), such as span length, number of hops, wavelength, types/number/gain of the traversed optical amplifiers, etc.

### B. OSNR Window Formation and Features Extraction

To train the ML models, we further pre-process raw OSNR to generate *OSNR windows* containing a sequence of OSNR samples for a duration of $W$ seconds. Once windows are formed, they are treated independently one from another at train/test phases, as our goal is to train failure detection and localization models that can work with a single "snapshot"of the OSNR windows. Following the approach in [1], [4], for each OSNR window we consider the following 16 features:

- $x_1 - x_{10}$: the ten strongest spectral components in the window, extracted by applying Fast Fourier Transform (FFT) on the OSNR window;
- $x_{11} = min$: minimum OSNR value;
- $x_{12} = max$: maximum OSNR value;
- $x_{13} = mean$: mean OSNR value;
- $x_{14} = std$: OSNR standard deviation;
- $x_{15} = p2p$: peak-to-peak OSNR, i.e., $p2p = max - min$;
- $x_{16} = RMS$: OSNR root mean square.

We then perform features normalization to obtain features ranging between -1 and 1 (normalization is regularly used to avoid that specific features provide different impact on ML algorithm training due to their different scale, and to reduce the duration of model training).

Finally, we also assign a label to each window, that depends on the specific lightpath scenario (i.e., presence/absence of failure and its location), and problem considered (i.e., detection or localization). As an example, in the case of failure localization on LP3 with 3 hops, we have 4 possible labels, corresponding to the cases of no failure and failure at each one of the three links traversed by the lightpath.

### C. Generation of SD/TD Scenarios

We run various experiments to evaluate TL for cross-lightpath and cross-task failure detection and localization. For each experiment, we consider subsets of the original dataset and generate distinct SD and TD datasets as follows:

- *Cross-lightpath failure localization*: here we evaluate TL performance when a ML model is pre-trained and optimized to perform **failure localization** on a SD lightpath and is then fine-tuned with a limited amount of data from the TD lightpath, where failure localization is performed. We consider two alternatives, i.e., when we consider LP1 and LP3 as SD and TD, respectively, and vice-versa.
- *Cross-task failure detection/localization*: we evaluate TL performance when the ML model is pre-trained to solve a problem and then retrained and fine-tuned to solve another problem. We consider both detection-to-localization and localization-to-detection TL, **considering LP3 in all scenarios**. Note that, differently from cross-lightpath TL, in these cross-task TL use-cases, the number of possible outputs in the two problems (localization and detection) is different, so proper ML model adaptation is necessary when fine-tuning the model, as explained later. More specifically, since we consider a 3-hop lightpath, we have 4 classes for failure localization (corresponding

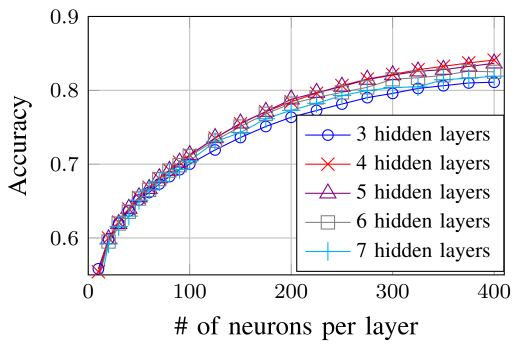



Fig. 3: Example of ANN model optimization for failure localization in LP3. Impact of ANN hyperparameters on classification accuracy (dropout = 0.1, $W = 50$ seconds).

to the cases of no failure and failure in one of the 3 links traversed by the lightpath) and 1 class for failure detection, (representing the presence of failure along the lightpath, regardless its location). Note that, for this TL use-case, we do not consider LP1 data, as detection and localization problems coincide.

### *D. ML Model Optimization*

After SD and TD datasets have been identified, we consider SD data only for ML model optimization. In this paper we consider feed-forward ANNs as ML models, as they are well recognized as practical models to apply TL. We optimize the ANN hyperparameters (i.e., number of hidden layers, number of hidden neurons per layer, and dropout rate) by means of k-fold cross-validation. Moreover, we consider ML model optimization for different values of window duration $W$, ranging from 10 to 100 seconds.

Note that, given the high number of ANN hyperparameters and values of window duration that can be used to optimize the various ML models, we here report a sensitivity analysis showing the impact of the number of neurons and number of hidden layers of the ANN in model performance, expressed in terms of classification accuracy. This analysis is shown in Fig. 3, where we consider the problem of failure localization in lightpath LP3 and set the window duration as $W = 50$ seconds. An exhaustive evaluation of all the combinations of parameters is not the main focus of our paper.

This analysis suggests that, for the set of tested hyperparameters, the most reasonable choice is to set the number of hidden layers to 4, as it represents the model with lowest complexity (less trainable parameters) providing the highest accuracy. In fact, we can observe from the figure that with lower (i.e., 3) or higher (especially 7) hidden layers, model performance deteriorates, due to underfitting and overfitting, respectively. On the other hand, an increasing number of neurons per layer shows an improvement in model performance compared to lower values, as shown in the figure, where we obtain a plateau in accuracy with 400 neurons per layer.

### *E. TL-based Model Fine-Tuning*

After ML model has been optimized and trained using SD data, fine-tuning can be applied by retraining the model with few TD data. Feed-forward ANNs, as those used in this paper, are known as universal function approximators, i.e., they are able to represent raw data as non-linear combinations of the input features at the hidden layers, representing *latent* features, and theoretically learn any non-linear function that maps the input features in the output labels. For this reason, ANNs are particularly suitable for TL, due to the fact that model weights learned during training on SD might be sufficiently informative also for the TD data, and so ANN model structure (i.e., the hyperparameters) and weights can be reused in the TD. In this context, retraining the ANN by applying backpropagation algorithm with limited amounts of TD data starting from a pre-trained model aims at maintaining the ability, acquired in the SD, in representing data with latent features, i.e., the activations of neurons at a certain hidden layer, and adjusting the weights in the subsequent layers to generate an output that is adapted to the TD.

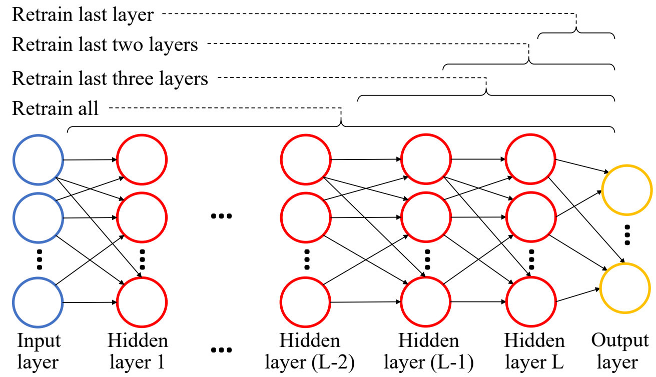


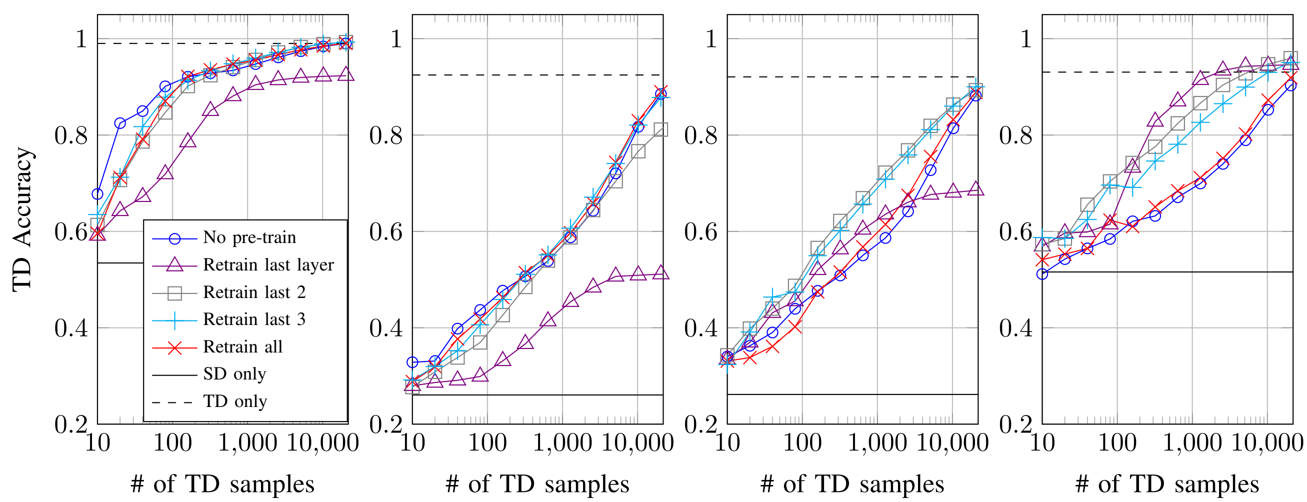

Fig. 4: Retraining options in the TL-based model fine-tuning.

We perform fine-tuning of pre-trained models in different ways, i.e., by considering different layers in the ANN to be retrainable with TD data. More specifically, we consider retraining of 1) *the last (output) layer*, 2) *the last two layers*, 3) *the last three layers* or 4) *all layers*, as graphically shown in Fig. 4. Note that, when considering cross-task TL between failure detection and localization considering a three-hop lightpath (LP3), the last layer of the ANN (i.e., the output layer) has to be retrained, as the SD and TD consist of two classification problems (tasks) with different number of classes.

## IV. Numerical Results

In this section, we discuss TL results obtained for cross-lightpath and cross-task failure detection and localization under different SD/TD scenarios. We concentrate on the impact of an increasing number of TD samples used for retraining, and evaluate model performance in terms of classification accuracy and TD training duration, considering, for each case, a fixed amount of 20% of the specific TD data as test set. Specifically, for all cases we evaluate the impact of retraining subsets of the ANNs with TD data, i.e., retraining only the last one/two/three layer(s) or the entire ANN, and compare these cases with the following benchmark scenarios:

- *SD only*: we train the ML model using only SD data and then test on TD data without any fine-tuning with TD




(a) Localization: LP3→LP1 (b) Localization: LP1→LP3 (c) LP3: Detection→Localization (d) LP3: Localization→Detection

Fig. 5: Accuracy vs. # of TD samples used for retraining in the various TL scenarios and window size W=50 seconds. (a-b) Cross-lightpath failure localization: (a) SD=LP3, TD=LP1; (b) SD=LP1, TD=LP3. (c-d) Cross-task failure detection/localization for LP3: (c) SD=detection, TD=localization; (d) SD=localization, TD=detection.

data[3]. *SD only* is a lower bound on model performance.

- *TD only*: we perform training with all available TD data (excluding the portion used for testing). *TD only* provides an upper bound on model performance;
- *No pre-train*: the model is not pre-trained with SD data, but instead it is trained from scratch with the same TD data used in the proposed FL approach.

Moreover, in all cases, the number of SD windows used to optimize/train the initial ML model (step 4 in Fig. 2) is around 21000, and each window has duration W=50 seconds.

### A. Cross-lightpath TL

We now consider the application of TL for failure-localization across lightpaths LP1 and LP3, shown in Fig. 1. We are interested in observing whether changing lightpath length between SD and TD has an impact in TL performance. For the various strategies, we show TD accuracy for increasing number of TD samples used for retraining for the two cases where {SD=LP3, TD=LP1} (LP3→LP1 in Fig. 5(a)) and {SD=LP1, TD=LP3} (LP1→LP3 in Fig. 5(b)).

First, we observe that failure localization on LP3 is more difficult compared to LP1, as intuition would suggest, due to the fact that more failure locations must be discriminated. In fact, higher accuracy is obtained in *TD only* scenario (horizontal dashed lines), when LP3 is the TD (Fig. 5(b)), compared to the case where LP1 is the TD (Fig. 5(a)). For both LP3→LP1 and LP1→LP3 cases, performing TL by retraining only the last layer (*Retrain last layer*) is the worst choice in terms of accuracy, which is lower compared to other approaches and even saturates to a maximum value (around 0.92 and 0.5 for LP3→LP1 and LP1→LP3, respectively) for

[3]Note that, in case the SD and TD problems have different number of outputs (e.g., different possible failure locations as happens for lightpaths with different number of hops), after training with SD data, the weights of the last layer in the ANN are randomly chosen.

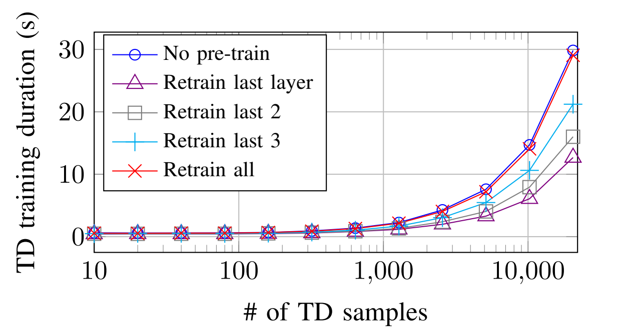



Fig. 6: Failure localization LP3→LP1: Training duration vs. # of TD samples used for retraining for window size W=50 s.

higher TD samples used to retrain. This suggests that, for cross-lightpath localization, fine-tuning weights only at the last layer is not sufficient, as there is a strong bias from the SD training phase, especially when transferring model knowledge from a shorter to a longer path (as in Fig. 5(b)). Instead, retraining two or more layers provides similar accuracy, that, for the LP3→LP1 case, is around 90% or higher when using 100 TD data points or more, and reaches the performance of *TD only* for about 5k data points used for retraining. For the LP1→LP3, TL accuracy when retraining two or more layers is satisfactory (i.e., higher than 80%) only after using 10k points for retraining, due to the fact that the TD lightpath has more candidate failure location compared to the SD lightpath.

Another interesting observation is that, for both LP3→LP1 and LP1→LP3 cases, retraining two or more layers has similar performance than *No pre-train* case, suggesting that the effort produced by TL during fine-tuning of trainable parameters is spent to compensate the bias acquired during training with SD data and maintained in the ANN layers that are kept unchanged when passing from SD to TD. This result, however, does not directly suggests that TL is a no-go option for this problem, for a reason related to training duration with TD data. In Fig. 6, we show training duration with TD data for the TL cases compared to *No pre-train*. For a fixed amount of TD data points used for retraining (x axis), fine-tuning (*Retrain all*) or training from scratch (*No pre-train*) all weights in the ANN are not scalable as they require in general much higher training time w.r.t. retraining less layers. It is true that, when adopting TL, an initial training is also performed on the SD, which is not due in *No pre-train* case. However, it is fair to assume that training on a SD is not an issue both from data collection and training duration perspectives. Therefore, TL (and in particular, for the LP3→LP1 localization under analysis, *Retrain last 2*) is a good compromise between TD accuracy, needed amount of TD data and TD training duration.

### B. Cross-task TL

Moving to cross-task TL, we now show results for detection→localization and localization→detection use cases in Figs. 5(c)-(d), considering LP3. We observe that, similarly to cross-lightpath cases, retraining only the last layer is not sufficient in detection→localization (see Fig. 5(c)), but retraining two or three layers outperforms the other strategies especially for medium-low amounts of TD data (between 100 and 1000 data points), when accuracy is up to around 12% higher than the other cases. On the other hand, retraining only the last layer is sufficient (and even more convenient, due to the reduced number of trainable ANN parameters, hence reduced training duration) for localization→detection (see Fig. 5(d)). This is due to the fact that the TD task (i.e., failure detection) is a more general version of failure localization, where less level of detail is needed for the classification, considering that failure localization distinguishes between no failure and all failures at all possible links, while failure detection only distinguishes between no failure and failures at *any* link. The advantage of *Retrain last layer* in this case is more evident if compared to *Retrain all* and *No pre-train* scenarios, where accuracy is up to 25% higher for a number of TD samples around 1000.

## V. Conclusion

We focus on the application of TL considering cross-lightpaths and cross-task failure detection and localization, considering ANN models and using real OSNR traces collected at lightpaths receivers on a laboratory testbed. For cross-lightpath failure localization, we found that retraining at least two ANN layers is necessary to have relatively high accuracy, in the order of 90% (respectively, 80%) for the cases when the TD lightpath is shorter (respectively, longer) than the SD lightpath and with at least 100 (respectively, 10k) data points. In general, retraining at least two layers outperforms other cases (including, in particular, *No pre-train*, *Retrain all* and *Retrain last layer*) in terms of accuracy, training duration or both. For cross-task failure detection/localization, we found that, considering failure detection as TD task allows significant improvement in TD accuracy, also for limited amounts of TD data points (around 100-1000) and also in cases where few ANNs parameters are fine-tuned with TL (i.e., *Retrain last layer* case), mainly due to the fact that less level of detail is needed for failure detection, compared to failure localization. Conversely, for detection-to-localization TL, relatively high number of TD data (in the order of few thousands) is necessary to have 80% or higher accuracy.

As future work, we plan to extend this study by considering different ML algorithms, such as tree-based models. We also aim at addressing TD data scarcity by means of synthetic data generation leveraging few real TD data.

## Acknowledgements

This work was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART"). The authors would like to thank Dr. Yusuke Hirota, Dr. Yoshinari Awaji and Dr. Masaki Shiraiwa from the National Institute of Information and Communications (NICT), Japan, for providing data and guidance on the testbed settings.

## References

[1] F. Musumeci, C. Rottondi, G. Corani, S. Shahkarami, F. Cugini, and M. Tornatore, "A tutorial on machine learning for failure management in optical networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4125–4139, 2019.

[2] X. Chen, C.-Y. Liu, R. Proietti, Z. Li, and S. J. B. Yoo, "Automating Optical Network Fault Management with Machine Learning," *IEEE Communications Magazine*, pp. 1–7, 2022.

[3] 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]," *Journal of Optical Communications and Networking*, vol. 13, no. 2, pp. A144–A155, 2021.

[4] F. Musumeci, V. G. Venkata, Y. Hirota, Y. Awaji, S. Xu, M. Shiraiwa, B. Mukherjee, and M. Tornatore, "Domain adaptation and transfer learning for failure detection and failure-cause identification in optical networks across different lightpaths [Invited]," *Journal of Optical Communications and Networking*, vol. 14, no. 2, pp. A91–A100, 2022.

[5] J. Pesic, M. Lonardi, E. Seve, N. Rossi, and T. Zami, "Transfer Learning from Unbiased Training Data Sets for QoT Estimation in WDM Networks," in *2020 European Conference on Optical Communications (ECOC)*, 2020, pp. 1–4.

[6] I. Khan, M. Bilal, M. Umar Masood, A. D'Amico, and V. Curri, "Lightpath QoT computation in optical networks assisted by transfer learning," *Journal of Optical Communications and Networking*, vol. 13, no. 4, pp. B72–B82, 2021.

[7] C.-Y. Liu, X. Chen, R. Proietti, and S. J. B. Yoo, "Performance studies of evolutionary transfer learning for end-to-end QoT estimation in multi-domain optical networks [Invited]," *Journal of Optical Communications and Networking*, vol. 13, no. 4, pp. B1–B11, 2021.

[8] D. Azzimonti, C. Rottondi, A. Giusti, M. Tornatore, and A. Bianco, "Comparison of domain adaptation and active learning techniques for quality of transmission estimation with small-sized training datasets [Invited]," *Journal of Optical Communications and Networking*, vol. 13, no. 1, pp. A56–A66, 2021.

[9] Z. Xu, C. Sun, T. Ji, J. H. Manton, and W. Shieh, "Feedforward and Recurrent Neural Network-Based Transfer Learning for Nonlinear Equalization in Short-Reach Optical Links," *Journal of Lightwave Technology*, vol. 39, no. 2, pp. 475–480, 2021.

[10] P. J. Freire, D. Abode, J. E. Prilepsky, N. Costa, B. Spinnler, A. Napoli, and S. K. Turitsyn, "Transfer Learning for Neural Networks-Based Equalizers in Coherent Optical Systems," *Journal of Lightwave Technology*, vol. 39, no. 21, pp. 6733–6745, 2021.

[11] Q. Yao, H. Yang, A. Yu, and J. Zhang, "Transductive Transfer Learning-Based Spectrum Optimization for Resource Reservation in Seven-Core Elastic Optical Networks," *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4164–4172, 2019.