Auto-DTWave: Digital Twin-Aided Autonomous Optical Network Operation with Continuous Wavelength Loading

Xiaomin Liu, Qizhi Qiu, Yihao Zhang, Meng Cai, Yichen Liu, Lilin Yi, Weisheng Hu, and Qunbi Zhuge
State Key Laboratory of Advanced Optical Communication Systems and Networks,
Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China
**Corresponding author e-mail address: qunbi.zhuge@sjtu.edu.cn*

Abstract: We develop joint online digital twin (DT) construction and amplifier configuration with continuous wavelength loading in a commercial testbed. The DT achieves an RMSE of 0.37dB, assisting near-optimal amplifier configuration with <0.1dB average Q-factor deviation. © 2024 The Author(s)

1. Introduction

To cope with fast-growing network traffic and reduce operational expenditure, autonomous driving optical network (ADON) is being pursued to achieve highly efficient self-management in a zero-touch manner [1]. Nevertheless, complex physical effects such as fiber nonlinearities coexist with numerous uncertain link parameters, posing a significant challenge in building ADON that requires extremely high levels of reliability.

To enable ADON, two key building blocks are needed. First, a digital twin (DT) of the physical layer should be constructed to assist network operations [2]. Today, physics-based and data-driven DT construction schemes are reported [3-8], but they require to either accurately measure all link parameters or pre-collect extensive training data, which can be impractical or costly in real networks. To synchronize DT with the real system, transfer learning (TL) is commonly employed with online collected data [9-11]. However, the size of all the data over a lifecycle of optical networks (e.g., >10 years) could introduce prohibitively huge storage burdens and computational complexities.

Second, ADON requires trustworthy network operations to optimize system performance. In particular, controlling optical power through configuring cascaded optical amplifiers (OAs) is essential to maximize link capacity. To date, offline or *in-situ* methods have been investigated [12-16]. However, the OA configuration is typically considered in a specific wavelength loading scenario, and the historical operation knowledge in previous loading scenarios is underutilized. This limits the potential of ADON to enhance operational performance and improve reliability. Moreover, how to leverage network operation to assist the online construction of DT also needs further study.

In this paper, we propose Auto-DTWave, a framework for ADON to efficiently operate over its lifecycle. We report the demonstration of joint online DT construction and OA configuration with continuous wavelength loading in a real-time commercial testbed. We propose a knowledge-accumulating DT (KA-DT), which can assimilate and accumulate historical network knowledge without saving the collected data. Without pre-collecting field data, we show the KA-DT's accuracy can improve as loading status varies, reducing the root-mean-squared error (RMSE) from 1.07 dB to 0.37 dB. Cooperated with KA-DT, a reliable OA configuration method is designed. Without interrupting the live traffic, it achieves near-optimal Q-factors with less than 0.1 dB deviation.

2. Principle

Fig. 1(a) presents the workflow of Auto-DTWave from the beginning of life (BoL) to a series of wavelength loading status. The schematic diagram and pseudo code are shown in Fig. 1(b) and 2(a), respectively.

KA-DT construction (stage ①): For the considered link, a synthetic dataset considering diverse wavelength loading status and OA configurations is generated using the Gaussian noise (GN) model [17] with the parameters from the given datasheets (line 1~2). Then, a neural network (NN) is trained with these data as the initial KA-DT (line 4). All

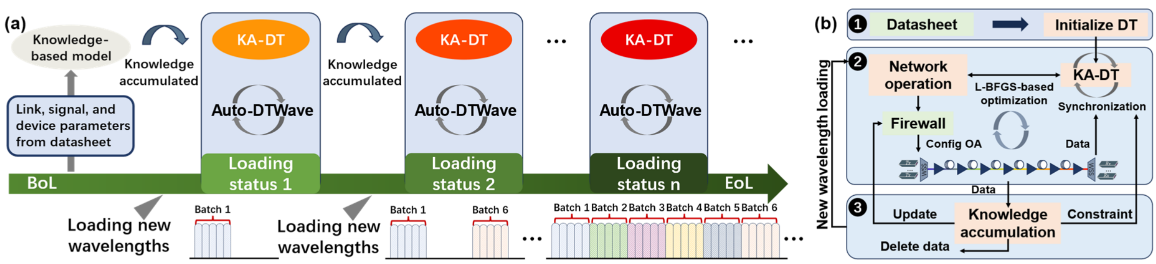


Fig. 1. (a) The workflow of Auto-DTWave with continuous wavelength loading from the BoL to end of life (EoL). (b) The schematic diagram of Auto-DTWave.

Algorithm: Auto-DTWave for continuous wavelength loading and knowledge accumulation

1: Generate diverse channel loading conditions $\boldsymbol{X} = \{\boldsymbol{X}_i\}$ and OA configurations $\boldsymbol{G} = \{\boldsymbol{G}_i\}$ and $\boldsymbol{T} = \{\boldsymbol{T}_i\}$

2: Compute Q-factor based on GN model with given datasheets
$$\boldsymbol{Q}_{\mathrm{GN}} = \{\boldsymbol{Q}_i\} = \mathrm{GN}(\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{T}), \boldsymbol{Q}_i = [Q_{\lambda 1}, Q_{\lambda 2}, \ldots, Q_{\lambda n}]$$

3: Gaussian process regression imputation for idle channel
$$\boldsymbol{Q}_{i_{\mathrm{idle}}} = \mathrm{GPR}\left(\boldsymbol{x}_{i_{\mathrm{idle}}} \middle| \boldsymbol{Q}_{i_{\mathrm{loaded}}}, \boldsymbol{x}_{i_{\mathrm{loaded}}}\right)$$

4: Initialize NN-based DT with $\{\boldsymbol{Q}_{\mathrm{GN}}, \boldsymbol{X}, \boldsymbol{G}, \boldsymbol{T}\}$
$$\boldsymbol{Q}_{\mathrm{NN}} = [Q_{\lambda 1\_\mathrm{NN}}, Q_{\lambda 2\_\mathrm{NN}}, \ldots, Q_{\lambda n\_\mathrm{NN}}] = \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{T})$$

5: For i-th service loading status $\boldsymbol{X}_{\mathrm{status}_i}$

6: While not converged

7: $\boldsymbol{G}, \boldsymbol{T} = \mathrm{argmax}_{\boldsymbol{G},\boldsymbol{T}} \frac{1}{N} \sum Q_{\lambda i\_NN} \quad s.t.\ \boldsymbol{X}_{\mathrm{status}_i}[i] \neq 0$

8: Evaluate through firewall and config OAs

9: Read real Q-factor $\boldsymbol{Q}_{\mathrm{real}}$ and impute idle channel

10: Add $\{\boldsymbol{Q}_{\mathrm{real}}, \boldsymbol{X}_{\mathrm{status}_i}, \boldsymbol{G}, \boldsymbol{T}\}$ to current learning dataset

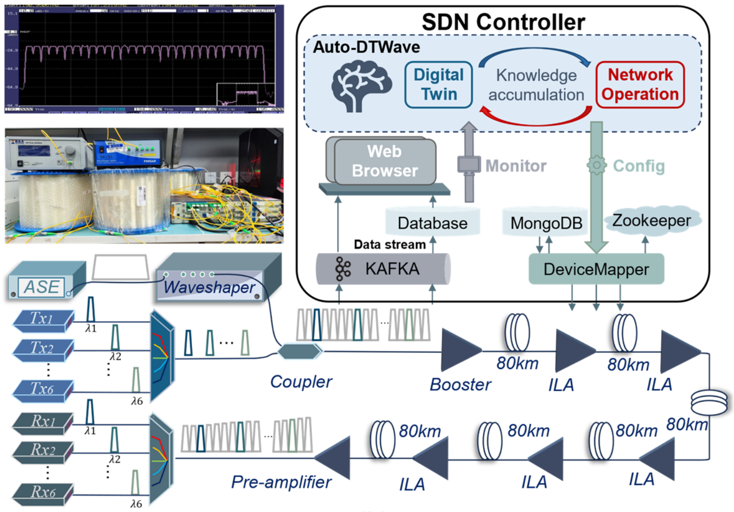
11: Synchronize KA-DT $\boldsymbol{f}_{\mathrm{NN}}$ from $\boldsymbol{X}_{\mathrm{status}_{i-1}}$ to $\boldsymbol{X}_{\mathrm{status}_i}$ by minimizing the estimation error of the loaded signals and maintain the previous knowledge
$$\mathrm{Loss}(\theta) = \mathrm{Loss}_{\mathrm{mse\_loaded}}(\theta) + \lambda \sum_{i,j} \Omega_{ij} \left(\theta_{ij} - \theta_{ij}^*\right)^2$$

12: End while

13: Accumulate knowledge through MAS
$$\Omega_{ij,\mathrm{status}_i} \leftarrow \Omega_{ij,\mathrm{status}_{i-1}} + \frac{1}{N} \sum_{k=1}^{N} \left\| \frac{\partial [\ell_2^2(f_{\mathrm{NN}}(x_k;\theta))]}{\partial \theta_{ij}} \right\|$$

14: Update firewall with Bayesian analysis with $\boldsymbol{Q}_{\mathrm{real}}$
$$\boldsymbol{F}_{firewall_{i+1}}(\boldsymbol{Q}_{\mathrm{GN}}, \boldsymbol{X}_{\mathrm{status}_i}, \boldsymbol{G}, \boldsymbol{T})$$

(a)



(b)

Fig. 2. (a) The pseudo code of Auto-DTWave. (b) The commercial testbed with SDN controller.

adjustable parameters of the link for network operations are used as the input. In our work, the input includes the launch signal power of each channel, the gain $\boldsymbol{G}$ and tilt $\boldsymbol{T}$ of all OAs. The output of the KA-DT includes the Q-factors of all channels, denoted as $\boldsymbol{Q}_{\mathrm{NN}}$. For the idle channel, a proxy label is filled through Gaussian process regression (GPR) imputation (line 3).

KA-DT-cooperated OA configuration (stage ②): When a new batch of wavelengths is loaded, OAs are configured to optimize the Q-factors of all signals without interrupting existing live traffic. Firstly, the average Q-factor of all loaded signals is maximized based on the estimation of the KA-DT through L-BFGS algorithm [18] to obtain the optimized $\boldsymbol{G}$ and $\boldsymbol{T}$ (line 7). To ensure that the live traffic would not be interrupted, the obtained $\boldsymbol{G}$ and $\boldsymbol{T}$ are evaluated by a firewall jointly built with the GN model and previous operation knowledge using the Bayesian analysis (line 8,14) [19]. Then, the reliable $\boldsymbol{G}$ and $\boldsymbol{T}$ are applied to real links, and the resulted Q-factor $\boldsymbol{Q}_{\mathrm{real}}$ can be collected and used for TL-based KA-DT synchronization (lines 9~11). The L-BFGS-based optimization is repeated based on the updated KA-DT until $\boldsymbol{Q}_{\mathrm{real}}$ converges.

Data assimilation and knowledge accumulation (stage ③): After completing the adaptive OA configurations in the current loading status, we utilize memory-aware synapses (MAS) [20] to compute the importance weights of each neuron based on the data collected in this status (line 13). Then, the importance weights are integrated to a regularization function to assist the TL-based synchronization in the next loading status (line 11). Such regularization can ensure that the NN's accuracy would not degrade on the previously-seen status, thus avoiding catastrophic forgetting [21] in a sequential TL scheme. Furthermore, the collected data can be deleted after knowledge accumulation, thus eliminating the storage of massive data for optical links that endure >10 years.

3. Demonstration in a commercial testbed

The proposed Auto-DTWave is evaluated in a testbed with six commercial transponders operating at 400 Gbps with 63.9-Gbaud 16QAM signals and a FEC threshold of 2.3E-2. As shown in Fig. 2(b), at the transceiver sides, 75GHz-grid MUX/DEMUX are used. The transmission link includes six 80-km G.652 fibers. Five in-line amplifiers, one booster, and one pre-amplifier are adopted to compensate for fiber loss and control optical power. All devices are controlled by a software-defined network (SDN) controller through NETCONF protocol and YANG models.

To emulate diverse loading scenarios, an amplified spontaneous emission (ASE) noise source is filtered by a programmable spectrum processor to generate multiple 63.9-GHz dummy signals from 193.1 to 195.3 THz. These signals are coupled with the real signals after MUX. As shown in the bottom line of Fig. 1(a), for each batch of wavelengths, one real signal from the commercial transponder is positioned in the middle and surrounded by four dummy signals. In the experiment, these six batches of signals are sequentially loaded following an order of 1, 6, 4, 2, 5, and 3 to emulate the loading status with 5, 10, 15, 20, 25 and 30 wavelengths. For each loading status, Q-factors of the loaded real signals under 100 types of OA settings are measured. In total, 600 data are collected to test model accuracy. At the BoL, the DT is initialized with the incoherent GN model using the parameters from the datasheet, such as the noise figure table, the standard G.652 fiber parameters, the loss of the MUX/DEMUX (5.5dB), connectors (0.7dB), couplers (0.5dB), and the transponder back-to-back noise (SNRb2b=17dB). Then, the Auto-DTWave starts to construct the KA-DT and optimize the OA configurations when new wavelengths are loaded.

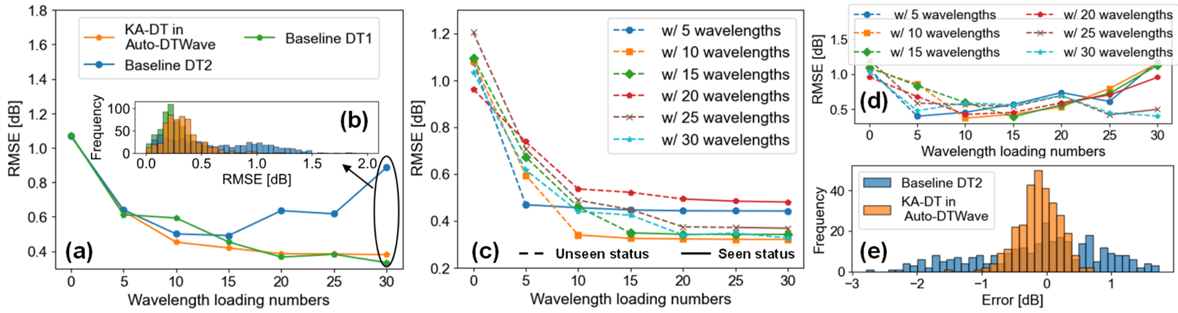


Fig. 3. (a) The accuracy of DTs in different loading status. (b) The histograms of the RMSEs of three DTs. The accuracy of (c) the KA-DT and (d) baseline DT2 on the testing datasets with different loading conditions. (e) The KA-DT and baseline DT2 synchronized with 30-wavelengths-loaded status are tested on 15-wavelengths-loaded status.

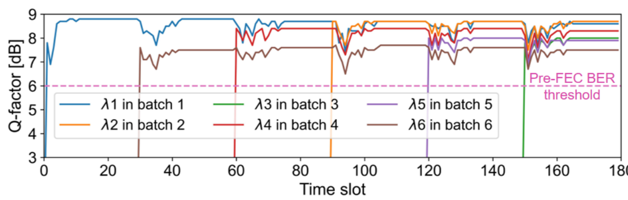


Table. 1: Optimized Q-factors (Auto-DTWave / Brute-force search).

	$\lambda 1$	$\lambda 2$	$\lambda 3$	$\lambda 4$	$\lambda 5$	$\lambda 6$
Status 1	8.8/8.8	N/A	N/A	N/A	N/A	N/A
Status 2	8.8/8.8	N/A	N/A	N/A	N/A	7.5/7.5
Status 3	8.7/8.8	N/A	N/A	8.4/8.4	N/A	7.6/7.7
Status 4	8.7/8.7	8.7/8.8	N/A	8.4/8.4	N/A	7.7/7.7
Status 5	8.7/8.7	8.7/8.7	N/A	8.4/8.4	8.0/8.1	7.6/7.7
Status 6	8.6/8.7	8.7/8.7	8.0/8.1	8.3/8.3	7.9/8.0	7.5/7.7

Fig. 4. The changes of Q-factors on each loading status.

After synchronizing with each loading status, we evaluate the accuracy of the KA-DT on the pre-collected testing dataset. For comparison, two baseline DTs are considered: 1) baseline DT1, a DT storing all historically collected data for TL; 2) baseline DT2, a storage-free DT without knowledge accumulation. In Fig.3(a) and 3(b), results show that the accuracy of the KA-DT gradually increases with an RMSE reduced from 1.07 dB to 0.37 dB. Additionally, without storing historical data, KA-DT can achieve similar accuracy as the baseline DT1 which stores all the historical data for TL. For the baseline DT2, the accuracy initially improves but then decreases due to catastrophic forgetting.

To further illustrate the knowledge accumulation ability, Fig. 3(c) shows the KA-DT's performance on the testing dataset with different loading status. For the previously-learned status, the KA-DT can maintain its performance when the loading status changes. For the baseline DT2 (Fig. 3(d)), it only targets to achieve high accuracy on the current loading status. As a result, overfitting could easily occur, resulting in catastrophic forgetting on the previously-learned knowledge. In Fig. 3(e), after synchronizing with the 30-wavelengths-loaded status, we test the KA-DT and baseline DT2 on the dataset with 15 loaded wavelengths. The result further proves that the proposed KA-DT shows a strong memory of the previously-acquired knowledge. Finally, the performance of the KA-DT-coupled OA configuration in Auto-DTWave is evaluated. The Q-factors of the loaded real signals during the experimental lifecycle are plotted in Fig. 4. Results demonstrate that during the lifecycle, OA configurations can be conducted without interrupting the live traffic. For all loading cases, the Q-factors converge in about 15 rounds. Table 1 compares the Q-factors maximized by Auto-DTWave and 200-times brute-force search, showing that Auto-DTWave can achieve near-optimal OA configuration with <0.1 dB average Q-factor deviation.

4. Conclusions

We propose Auto-DTWave for ADON to demonstrate the joint online DT construction and OA configuration with continuous wavelength loading in a real-time commercial testbed. In our experiment, the DT's accuracy gradually increases from 1.07 dB to 0.37 dB as channel number grows from 5 to 30. For the DT-aided OA configuration, near-optimal average Q-factors with <0.1dB deviation can be achieved without interrupting the live traffic.

Acknowledgements

This work was supported by Shanghai Pilot Program for Basic Research - Shanghai Jiao Tong University (21TQ1400213) and National Natural Science Foundation of China (62175145).

References

[1] A. Richard, TM Forum white paper, 2019.
[2] Q. Zhuge, *et al*, JOCN, 15(8), C242–C262, 2023.
[3] Y. Pointurier, JOCN, 13(4), B60, 2021.
[4] R. Yang, *et al*, OFC, W4G.5, 2023.
[5] V. Curri, JOCN, 14(6), C92, 2022.
[6] A. Ferrari, *et al*, OFC, M3Z.14, 2022.
[7] X. Yang, *et al*, JOCN, 15(7), C20, 2023.
[8] J. Müller, *et al*, OFC, Tu2F.2, 2023.
[9] J. Yu, *et al*, JOCN, 11(10), C48–C57, 2019.
[10] J. Pesic, *et al*, ECOC, Mo2K-2, 2020.
[11] D. Azzimonti, *et al*, OFC, M4E.1, 2020
[12] E. de A. Barboza, *et al*, JLT, 37(16), 3994, 2019.
[13] A. Mahajan, *et al*, JOCN, 13(4), B45, 2021.
[14] Z. Zhong, *et al*, OFC, F3B.1, 2021.
[15] H. Gao, *et al*, OFC, Th4C.2, 2023.
[16] G. Borraccini, *et al*, JOCN, 15(2), 111–118,2023.
[17] P. Poggiolini, *et al*, JLT, 32(4), 694–721, 2014.
[18] C. Zhu, *et al*, ACM Transactions on mathematical software, 23(4), 550–560, 1997.
[19] C. M. Bishop, Pattern recognition and machine learning. New York: Springer, 2006.
[20] R. Aljundi, *et al*, ECCV, 2018.
[21] J. Kirkpatrick, *et al*, Proceedings of the National Academy of Sciences of the United States of America, 114(13), 3521–3526, 2017.