OEA | Opto-Electronic Advances

Article

July 2026, Vol. 9, No. 7

Open Access

Digital twin optical computing system

Run Sun[1,2†], Yuemin Li[1,2†], Tingzhao Fu[3*], Wencan Liu[1,2], Sigang Yang[1,2] and Hongwei Chen[1,2*]

Abstract: Optical computing systems (OCS) are promising accelerators for artificial intelligence due to their high bandwidth, low latency, and inherent parallelism. However, in existing OCS implementations, task development often requires direct participation of physical hardware during training and optimization, tightly coupling development workflows to device access and limiting offline design, reproducible benchmarking, and parallel exploration. Here, we propose a Digital Twin Optical Computing System (DT-OCS), a system-level, measurement-driven digital surrogate that emulates the end-to-end input-output behavior of a specific physical OCS under different operating configurations. DT-OCS is implemented as a differentiable software module, enabling fully offline task training and configuration optimization. The optimized configuration parameters can be directly transferred to the physical OCS without hardware-in-the-loop retraining. We validate DT-OCS on a high-speed optical computing system operating at 10 GHz and equipped with a silicon-based integrated computing chip. Representative tasks, including image classification and temporal strategy generation, are evaluated. Across all tasks, the transferred models closely match the performance of their digital-twin counterparts after direct parameter transfer. These results demonstrate that DT-OCS enables a hardware-decoupled and fully offline development paradigm for optical computing systems. The DT-OCS implementation is released as open-source code to facilitate reuse and further development.

Keywords: digital twin system; high-speed optical computing; transfer training method

DOI: 10.29026/oea.2026.250254 | CSTR: 32247.14.oea.2026.250254

Citation: Sun R, Li YM, Fu TZ et al. Digital twin optical computing system. *Opto-Electron Adv* **9**, 250254 (2026).

1 Introduction

Optical computing systems (OCS) have become a promising architecture for next-generation artificial intelligence applications, owing to their ultra-high speed[1–4], low latency[5,6], and parallel computing capability[7–11]. As OCS is increasingly explored as a shared artificial intelligence (AI) accelerator supporting diverse task objectives and data modalities, its development workflow must support offline model design[12,13], reproducible benchmarking[14], and parallel exploration[15,16], consistent with established practices in the digital neural-network ecosystem. This requirement arises from the fact that task development on OCS can be driven by multiple users or research threads iterating on different tasks along independent development trajectories. To support such trajectories, task iteration must be decoupled from exclusive access to the physical OCS device, thereby enabling fair and reusable benchmarking across tasks and users. At the same time, this requirement calls for parallel mechanisms that accommodate asynchronous development cycles and prevent resource contention, without forcing them into a single serialized workflow.

However, in many state-of-the-art OCS implementations, the effective computation is governed by system-level behavior of the assembled hardware, rather than an idealized optical core[17,18]. The system-level effects introduce nonidealities that are difficult to capture in offline models without repeated reliance on the physical system. Consequently, as shown in the left diagram in Fig. 1, task development in practice commonly relies on hardware-coupled workflows such as in-situ training[19–21], hardware-aware fine-tuning[22,23], and measurement-driven evaluation[8,24,25] where model updates, parameter adjustments, and task iteration must be executed with exclusive access to the physical OCS through

Received: 23 September 2025

Accepted: 9 March 2026

Published online: 22 April 2026

[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China; [2]Beijing National Research Center for Information Science and Technology, Beijing 100084, China; [3]College of Advanced Interdisciplinary Studies & Hunan Provincial Key Laboratory of Novel Nano-Optoelectronic Information Materials and Devices, National University of Defense Technology, Changsha 410073, China.

[†]These authors contributed equally to this work.

*Correspondence: TZ Fu, E-mail: futingzhao@nudt.edu.cn; HW Chen, E-mail: chenhw@tsinghua.edu.cn

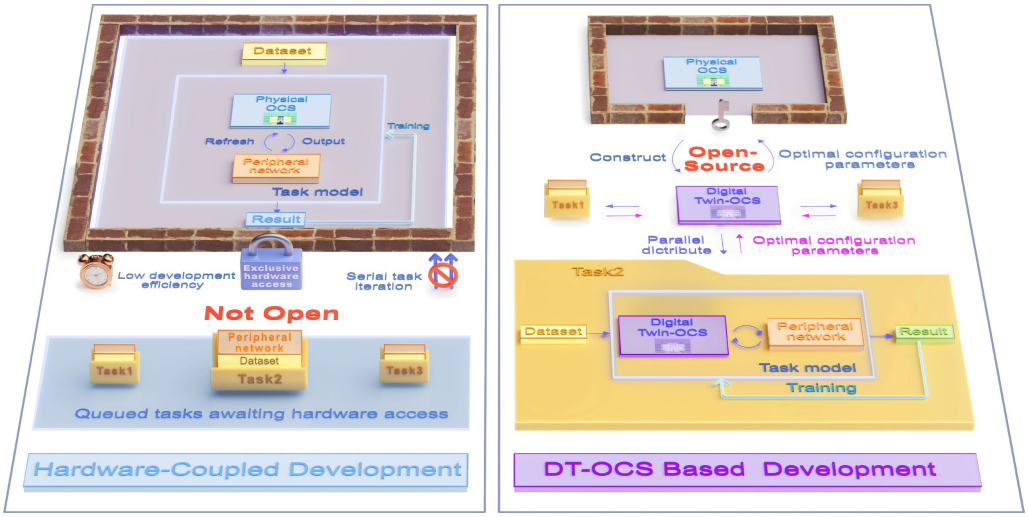


Fig. 1 | Comparison between hardware-coupled and DT-OCS based development workflows for optical computing systems. The left diagram illustrates a conventional hardware-coupled workflow, in which task development must be carried out with direct and exclusive access to the physical OCS. Task optimization relies on repeated hardware configuration and measurement, forcing different tasks to proceed in a serialized manner and limiting offline design, reproducible benchmarking, and parallel exploration. The right diagram shows the proposed DT-OCS based development workflow. By learning from measured data through supervised training, DT-OCS serves as a digital surrogate of the physical OCS. With reproducible DT-OCS incorporated into the task models, different tasks can be trained in parallel in the digital domain. The resulting optimized configuration parameters can be transferred back to the physical OCS while preserving its computational characteristics. This workflow decouples task iteration from hardware access and enables efficient, reproducible, and parallel task development on optical computing systems.

repeated configuration and measurement. Such hardware-coupled workflows fundamentally limit the ability to decouple task development from physical OCS.

These limitations originate from system-level nonidealities manifesting across a wide range of OCS platforms including on-chip diffractive processors, Mach-Zehnder interferometer (MZI)-based meshes, and micro-ring processors. For on-chip diffractive processors, the intended optical field distribution is implemented through nanostructured diffractive features[8,24], optically[26,27] or thermally[28] driven optical modulation elements, whose responses are affected by fabrication variability, thermal crosstalk, and bias-dependent nonidealities, leading to deviations between the nominal design and the actual realization. For MZI-based meshes, computation emerges from the collective interference of cascaded interferometers[29], where small errors in splitting ratios[17], phase efficiencies[30], and control electronics accumulate into a system-level response that is difficult to predict from component-level models alone. For micro-ring processors, strong bias-dependent sensitivity[31,32] further exacerbates model-hardware mismatch and device-to-device variability. Beyond the optical core, nonideal responses and coupling effects introduced by peripheral optoelectronic components[18] such as modulators, amplifiers, photodetectors, and electronic control and readout circuits further contribute to system-level nonidealities.

To resolve this conflict between hardware-coupled workflow and the requirements for offline model design, reproducible benchmarking, and parallel exploration, we propose a digital twin optical computing system (DT-OCS) that serves as a transferable, system-level digital surrogate of a specific physical OCS. DT-OCS is constructed by supervised learning from experimentally measured end-to-end input-output data of the physical OCS, enabling reproduction of the system's computational behavior in the digital domain.

To validate the proposed DT-OCS framework, we adopt a physical OCS equipped with a high-speed silicon-based integrated optical computing chip as a representative experimental platform. The system integrates this optical computing chip with a complete optoelectronic signal chain, including signal generation, amplification, modulation, photodetection, and electronic sampling. The system processes temporal input sequences at a symbol rate of 10 GHz. Based on this platform, we construct DT-OCS in a measurement-driven manner and implement it as a reproducible and callable Python module, as shown in the right diagram in Fig. 1. During task development, the DT-OCS replaces the physical OCS, allowing task models to be fully optimized offline. The resulting optimized configuration parameters are then directly transferred to the physical OCS for deployment, without additional hardware-in-the-loop training or post-deployment fine-tuning. We demonstrate the effectiveness of this workflow on representative tasks spanning different data modalities and model structures, including image classification and temporal decision-making. Across all tasks, the performance of transferred task models obtained with the physical OCS closely matches that

obtained during DT-OCS-based training, confirming the fidelity and transferability of DT-OCS.

Beyond task-level validation, DT-OCS enables the optical computing system to be exposed to the research community as a reproducible software-accessible resource. By decoupling task development from physical hardware availability, DT-OCS eliminates the data-intensive and time-consuming hardware-coupled optimization procedures, while supporting general optimization strategies such as gradient-based methods and grid search. From a system-level perspective, DT-OCS further provides a convenient framework for analyzing nonlinear response characteristics and noise behavior of the physical OCS. Together, these capabilities establish DT-OCS as a practical paradigm for transforming optical computing systems from hardware constrained experimental platforms into accessible, reproducible, and extensible computing resources.

2 Digital twin optical computing system

2.1 Principles and methods

DT-OCS is conceived as a system-level digital twin framework that models the end-to-end input-output behavior of a physical OCS. To illustrate the principles and construction methodology of DT-OCS, we instantiate the framework on a representative high-speed silicon-based integrated optical computing system, shown in Fig. 2(a), as the physical platform for validation. The optical computing chip in this system has high-speed temporal signal processing capabilities and supports programmable computing functions. Our previous work has validated that this type of physical OCS can compute the feature information of high-speed signals[33,34]. As shown in Fig. 2(c–e), the optical computing chip within the system consists of two core modules. 1) Data preprocessing module: This module performs sampling, serial-to-parallel conversion, and phase adjustment on the time-sequenced input signal $i(t)$, generating spatially parallel signal

$$\boldsymbol{I}(t) = \left[i(t-\Delta_3)\mathrm{e}^{\mathrm{i}\varphi_3}, i(t-\Delta_2)\mathrm{e}^{\mathrm{i}\varphi_2}, i(t-\Delta_1)\mathrm{e}^{\mathrm{i}\varphi_1}, i(t)\right], \quad (1)$$

where $\mathrm{e}^{\mathrm{i}\varphi_n}$ and Δ_n represent the added phase and delay time, respectively. By designing the lengths of the delay waveguides, the sampling rate is set to 10 GHz. 2) Feature extraction core: This core performs matrix multiplication. Based on the input vector $\boldsymbol{I}(t)$, the feature extraction core computes the output $O(t) = \boldsymbol{S}\boldsymbol{I}(t)$, where $\boldsymbol{S}$ represents the port scattering matrix of the feature extraction core. Under this operating condition, the physical OCS performs computation directly at the symbol rate, such that the computation itself does not introduce additional clocked digital latency. Using this optical computing chip package, the estimated throughput[25,28,35] and equivalent energy efficiency[36] reach 100 GOPS and 0.8 TOPS/W, respectively. Details of the optical computing chip initialization and the estimation of system-level metrics of the physical OCS are provided in Supplementary Section 1 and Section 2, respectively.

However, the actual output deviates from the expected behavior: firstly, parasitic dispersion, reflection, and resonance in the feature extraction core and other components in the waveguide optical path introduce wavelength-

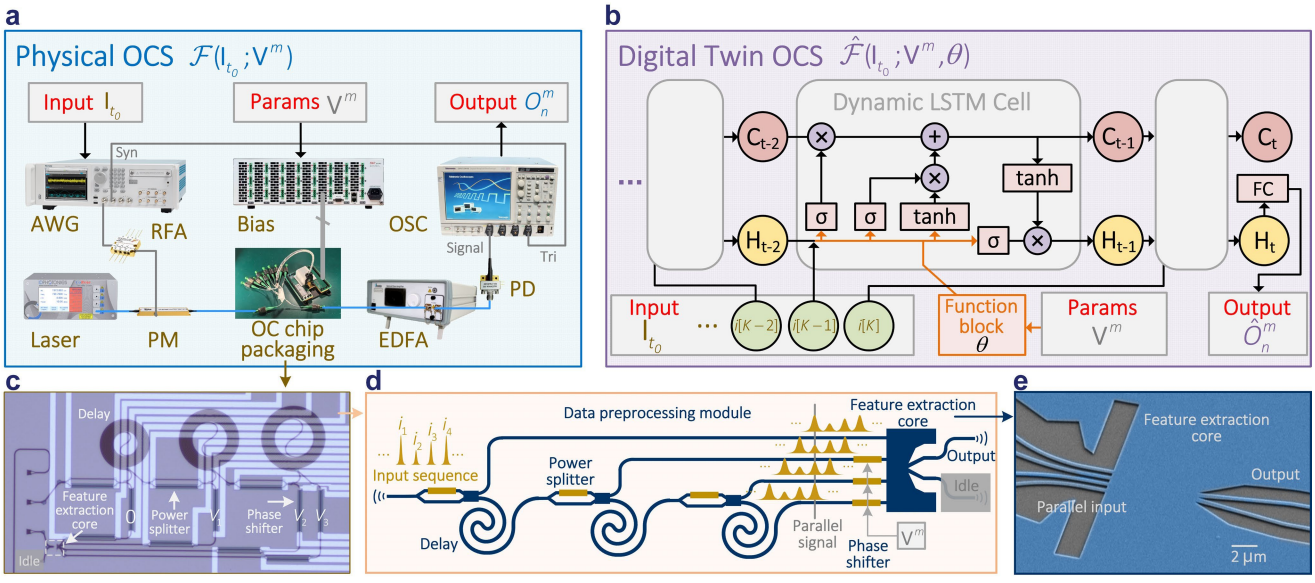


Fig. 2 | Physical OCS and DT-OCS. (a) Composition of the physical OCS. Black lines indicate signal paths, blue lines represent optical fibers, and gray lines denote microwave cables. AWG, arbitrary waveform generator; Bias, bias voltage source; OC, optical component; OSC, oscilloscope; PD, photodetector; PM, phase modulator; RFA, radio frequency amplifier; Syn, synchronization cable; Tri, trigger cable. (b) Structure of DT-OCS. FC, fully connected layer. (c) Optical microscopic image of the chip. (d) Schematic of the chip. (e) Electron microscope image of the diffraction structure and parallel input and output ports.

dependent responses. As a result, the output of the physical OCS is influenced by an extended input receptive field, which not only samples more discrete points but also incorporates adjacent time ranges. Similarly, as shown in Fig. 2(a), signal-related devices in the physical OCS further complicate this issue. These devices include the signal source device (AWG), modulation amplifier, phase modulator (PM), optical amplifier (EDFA), PD, and sampling device (OSC), all of which have limited bandwidth or non-ideal responses. Together, these factors result in a complex input-output mapping of the physical OCS that deviates from what can be described by simple linear matrix multiplication, necessitating a more generalized and systematic modeling approach to study the physical OCS. Detailed information on the system's components can be found in Supplementary Section 2.

The physical OCS is a finite memory system, whose output at a given time t depends on the input from a past time period, i.e.,

$$O(t) = \mathcal{F}(i(\xi \in [t-\tau, t]); \boldsymbol{V}) \ . \tag{2}$$

The operator $\mathcal{F}$ represents the computational functionality of the physical OCS and captures the overall system response resulting from the optical processor and the associated components, thereby defining the mapping rule between input and output data. The input $i(\xi)$ over a time period $[t-\tau, t]$ determines the system's output, where τ represents the time length of the receptive field. $\boldsymbol{V}$ represents the bias voltage of the phase shifters in the data preparation module of the optical computing chip in the physical OCS. By changing the bias voltage (configuration parameter) $\boldsymbol{V} = [V_p]$, the additional phase in the branch is altered, thus adjusting the computational functionality of the physical OCS. Here p represents the number of voltage configuration parameters, which corresponds to the number of actively controlled phase shifters in the OCS. The computational functionality $\mathcal{F}$ of the physical OCS is realized only through direct hardware involvement, and becomes inaccessible when the physical system is offline. By constructing DT-OCS, we create an open-source and callable operator $\hat{\mathcal{F}}$ as a surrogate for the physical operator $\mathcal{F}$, enabling DT-OCS to participate in task design and training while the physical OCS is offline.

The input signal to the physical OCS is a piecewise constant signal generated from the data to be computed. Therefore, the vector $\boldsymbol{I}_{t_0} = [i(t_0-(K-1)\Delta), \cdots, i(t_0-k\Delta), \cdots, i(t_0)]$ is able to encapsulate the input information for the physical OCS within the receptive field $[t_0-\tau, t_0]$. The number of elements in vector $\boldsymbol{I}_{t_0}$ is denoted as

$$K = \mathrm{ceil}[\tau/\Delta] \ , \tag{3}$$

where $\mathrm{ceil}[\cdot]$ is the ceiling function, Δ represents the modulation symbol duration, τ denotes the receptive field length, and t_0 refers to the sampling completion time point. The response of the physical OCS is similarly sampled at the same rate to form O_{t_0}. In this way, the computational function of the physical OCS can be described as follows: under configuration parameter $\boldsymbol{V}$, the physical OCS generates an output $O_{t_0} = \mathcal{F}(\boldsymbol{I}_{t_0}; \boldsymbol{V})$ based on the input $\boldsymbol{I}_{t_0}$.

According to the finite memory and causal characteristics of the physical OCS, we adopt a long short-term memory (LSTM) architecture[37] as the digital foundation of DT-OCS. To embed the configuration-dependent computational function of the physical OCS, the standard LSTM is modified such that its computational behavior is governed by the same configuration parameters $\boldsymbol{V}$. Specifically, as shown in Fig. 2(b), the internal parameters, including the weight matrices and bias vectors associated with the LSTM gates, are not fixed but are generated by a function block that takes the physical configuration parameters $\boldsymbol{V}$ as input. This function block is realized as a simple linear (affine) mapping from $\boldsymbol{V}$ to the gate parameters of the LSTM, while all matrices and bias in this mapping are trainable and collectively denoted as θ. In this manner, a standard LSTM cell is transformed into a Dynamic LSTM cell, whose recurrent dynamics explicitly depend on the configuration parameters of the OCS. The number of cells in DT-OCS is also set to K, ensuring that DT-OCS shares the same receptive field as the physical OCS. The final output of DT-OCS is $\hat{O}_{t_0} = \hat{\mathcal{F}}(\boldsymbol{I}_{t_0}; \boldsymbol{V}, \theta)$, where $\hat{\mathcal{F}}$ is realized with the DT-OCS network. DT-OCS is expected to be equivalent to the physical OCS, meaning that $\hat{\mathcal{F}}$ closely approximates $\mathcal{F}$. This equivalence is reflected in the fact that, given the same input $\boldsymbol{I}$ and configuration parameters $\boldsymbol{V}$, the output of DT-OCS $\hat{O}$ is sufficiently close to the output of the physical OCS O. This closeness is quantified by a similarity function (SF), which evaluates the mean squared error (MSE) between the two output sequences $\hat{O}$ and O in both time and frequency domains:

$$SF(\hat{O}, O) = 0.5MSE(\hat{O}, O) + 0.5MSE(|FFT(\hat{O})|, |FFT(O)|) \ . \tag{4}$$

Joint evaluation of errors in both time and frequency domains helps avoid degenerate similarity evaluations. Such degeneracy may arise when relying on a single representation, which tends to emphasize either local waveform details or global structural characteristics alone. Similar joint loss functions have been widely adopted in signal processing research[38,39]. Detailed implementation of DT-OCS is available in the open-source code linked at the end of the paper.

2.2 Results

We constructed a learning dataset for DT-OCS under the supervised learning framework. The learning objective of DT-OCS is to accurately approximate the input-output mapping of the physical OCS across different configuration parameters, as shown in Fig. 3(a). The dataset is organized in a triplet structure, where each training sample consists of a voltage configuration $\boldsymbol{V}^m$, an input segment $\boldsymbol{I}_{\{t_n\}}$, and the

measured physical OCS output $O^m_{\{t_n\}}$ under that configuration parameter, i.e., mapping pairs $\boldsymbol{V}^m - \boldsymbol{I}_{\{t_n\}} - O^m_{\{t_n\}}$. Specifically, 4000 distinct voltage configurations

$$\boldsymbol{V}^m = [V^m_1, V^m_2, V^m_3] , \tag{5}$$

are loaded onto the phase shifters shown in Fig. 2(c) of the optical computing chip in the physical OCS, where m denotes the configuration parameter index. These configurations $\boldsymbol{V}^m$ randomly sampled from a range spanning twice the half-wave voltage of the phase shifters. Under each configuration parameter, a fixed probing sequence is input into the physical OCS. This sequence consists of multimodal data, including image information, time series, and random noise, with a total length of 2500 points. Details of this sequence construction are provided in Supplementary Section 4. For a given configuration parameter, the physical OCS produces a corresponding output sequence in response to the input sequence. Representative segments of the input and the corresponding output sequence are shown in Fig. 3(f) and the blue markers in Fig. 3(g), respectively. Based on the each input-output sequence pair under each configuration parameter, a sliding time window is applied to the input sequence to extract input segments $\boldsymbol{I}_{t_n}$ of length K (K is the element number of the input vector, as defined in *Principles and methods*). Each input segment $\boldsymbol{I}_{t_n}$ is combined with the current configuration $\boldsymbol{V}^m$ and the corresponding output $O^m_{t_n}$ to form a complete mapping pair. Here, $t_n \in \{K, K+1, \cdots, 2500\}$ denotes the sampling time. To determine the system's receptive field τ and accordingly compute $K = \mathrm{ceil}[\tau/\Delta]$, a full-amplitude step function, shown in Fig. 3(b), is used to excite the physical OCS. The decay of the output response is recorded in Fig. 3(c). We define the receptive field τ as the time interval over which the effect output response (the DC component 0.5 subtracted) decays from its maximum to 15%, which is approximately 0.73 ns. The 15% level is adopted as an operational criterion corresponding to a point at which the output response has essentially converged, leaving only a weak tail contribution. Given the system's 10 GHz sampling rate ($\Delta = 0.1$ ns), we determined $K = 8$. Under this configuration, each voltage setting yields $2500 - K = 2492$ mapping pairs, resulting in a total of $2492 \times 4000 = 9.968$M mapping pairs, which form the supervised learning dataset for DT-OCS: $\boldsymbol{V}^m - \boldsymbol{I}_{\{t_n\}} - O^m_{\{t_n\}}$. The range of the half-wave voltage is provided in Supplementary Section 1.

The learning of DT-OCS is then conducted. The learning objective is for the computational function of DT-OCS

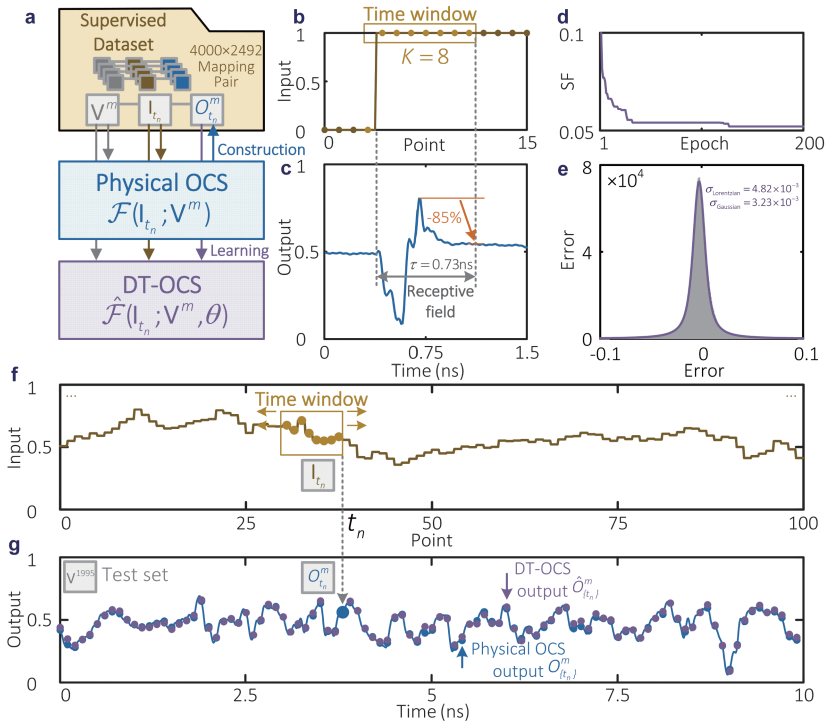


Fig. 3 | Construction process of DT-OCS. (a) Schematic of supervised learning for DT-OCS. (b) Step signal used to excite the physical OCS. The light yellow box highlights the time window and its corresponding input vector $\boldsymbol{I}_{t_0}$. (c) Response of the physical OCS to the input step signal. The gray lines mark the receptive field range and the orange line indicates the point where the amplitude decays to 15% of its maximum value. (d) Convergence process of the similarity function (SF) during the supervised learning of DT-OCS. (e) Residual distribution between DT-OCS and the physical OCS on the test set, along with the fitted function. (f) A segment of the probing sequence. The light yellow box highlights the time window and its corresponding input vector. (g) Output comparison between the physical OCS and DT-OCS. The blue line represents the unsampled response of the physical OCS, blue dots represent the output of the physical OCS, and purple dots represent the output of DT-OCS.

$\hat{\mathcal{F}}(\boldsymbol{I};\boldsymbol{V},\theta)$ to accurately approximate the function of the physical OCS $\mathcal{F}(\boldsymbol{I};\boldsymbol{V})$. The previously proposed time-frequency composite similarity function (SF) is selected as the loss function and optimization criterion for training the DT-OCS. By minimizing SF via gradient descent, DT-OCS is guided to progressively approximate the input-output mapping of the physical OCS across the entire configuration parameter space. During the learning process, the configuration parameters $\boldsymbol{V}$ are first passed into the function block defined by parameters θ, which then generates the internal parameters of the Dynamic LSTM cells. The learning process thus achieves the optimization of the function block parameters:

$$\theta^* = \underset{\theta \text{ in}\hat{\mathcal{F}}(\boldsymbol{I};\boldsymbol{V},\theta)}{\operatorname{argmin}} \sum_{\text{Mapping pair}} SF(\hat{O}^m_{\{t_n\}} - O^m_{\{t_n\}}) \,. \tag{6}$$

The 4000 voltage configurations are randomly divided into a train set and a test set in a 4∶1 ratio. As shown in Fig. 3(d), after 200 epochs of training, *SF* converges sufficiently. From the test set, we select one configuration and compare the outputs of DT-OCS and the physical OCS, as shown in Fig. 3(g). DT-OCS accurately fits the output of the physical OCS. Across the entire test set, the residuals between $\hat{O}_n$ and O_n are fitted using a Voigt profile. The standard deviation of its Gaussian component is $\sigma_{\text{Gaussian}} = 3.24 \times 10^{-3}$, and that of its Lorentzian component is $\sigma_{\text{Lorentzian}} = 4.82 \times 10^{-3}$. Both are below 0.5‰ of the maximum output amplitude, demonstrating the accuracy of the fit. At this point, the learning process of DT-OCS is complete, and the trained DT-OCS is used for subsequent training and testing tasks.

3 Task involvement of the digital twin system

In this section, the task-level application capability of DT-OCS is demonstrated. The trained DT-OCS is integrated with peripheral networks, which serve as task-dependent digital modules interfacing with the output of the OCS, to form a complete DT-task model. This DT-task model operates entirely in the digital domain and completes the full inference process from input data to final output while the physical OCS remains offline. After training the DT-task model, the optimal configuration parameters $\boldsymbol{V}^*$ for the OCS along with the parameters of the peripheral network are obtained. These parameters are then transferred to construct a transferred task model composed of the physical OCS configured with the optimized parameters and the peripheral network with the unchanged parameters. The performance of this transferred task model is directly validated on the test set without any further fine-tuning of the peripheral network. To illustrate the generality of this paradigm across different application requirements, we evaluate DT-OCS on several representative tasks, including single-channel handwritten digit recognition, multi-channel color clothing and footwear classification, and gold trading strategy generation. These tasks are selected to span diverse data modalities and task complexity, thereby demonstrating how DT-OCS supports hardware-decoupled task development and naturally enables parallel exploration of multiple applications.

As shown in Fig. 4(a), the MNIST handwritten digit recognition task is used as a representative example. The input data consists of single-channel grayscale images, and the goal is to accurately classify the digits 0 through 9 in the dataset. As shown in Fig. 4(b), the complete task model is composed of an OCS followed by a convolutional network. The input grayscale images are first reshaped into one-dimensional sequence through line scanning and fed into the OCS, which performs high-speed feature extraction on the input sequence. The resulting output sequence is then reshaped back into two-dimensional feature maps and processed by a peripheral convolutional network, which serves as a digital readout module that maps the feature representation generated by the OCS to the final task labels. The detailed procedures for data processing and reshaping are provided in Supplementary Section 5. As shown in Fig. 4(c), after 10 epochs of training, the DT-task network converges on the train set. On the test set, the optimized DT-task model achieves a classification accuracy of 98.09%. Subsequently, as shown in Fig. 4(d), the optimal configuration parameters of the DT-OCS $\boldsymbol{V}^* = [0.2530\ \text{V}, 1.3768\ \text{V}, 0.3330\ \text{V}]$ are transferred to the physical OCS. The peripheral network parameters are similarly transferred and kept fixed, forming a transferred task model to perform inference. Fig. 4(g) shows the segment of an input sequence corresponding to the digit "4" as shown in Fig. 4(i). In Fig. 4(h), the blue dots and purple dots represent the outputs from the physical OCS and DT-OCS, respectively, under the same optimal configuration parameters. The two outputs exhibit remarkable similarity. Figure 4(j) and 4(k) displays the feature maps reconstructed from the outputs of the physical OCS and DT-OCS, which show high consistency in both spatial structure and brightness distribution. The difference between the two feature maps is shown in Fig. 4(l). Across the entire test set, the average peak signal-to-noise ratio (PSNR) between the output feature maps of the physical OCS and DT-OCS is 29.01 dB, and the structure similarity index measure (SSIM) is 0.93. On the same test set, the transferred task model achieves a classification accuracy of 98.11%, with a difference of less than 0.02% compared to the DT-task model. Figure 4(m–o) presents the confusion matrices for classification with the DT-task model and the transferred task model, along with their element-wise differences. As shown in Fig. 4(o), the most significant discrepancy occurs for the digit “8”, with a per-class accuracy difference still below 1.3%. These results demonstrate the high-fidelity approximation of DT-OCS to the behavior of the physical OCS. This fidelity enables DT-OCS to serve as a substitute platform for training when the physical OCS

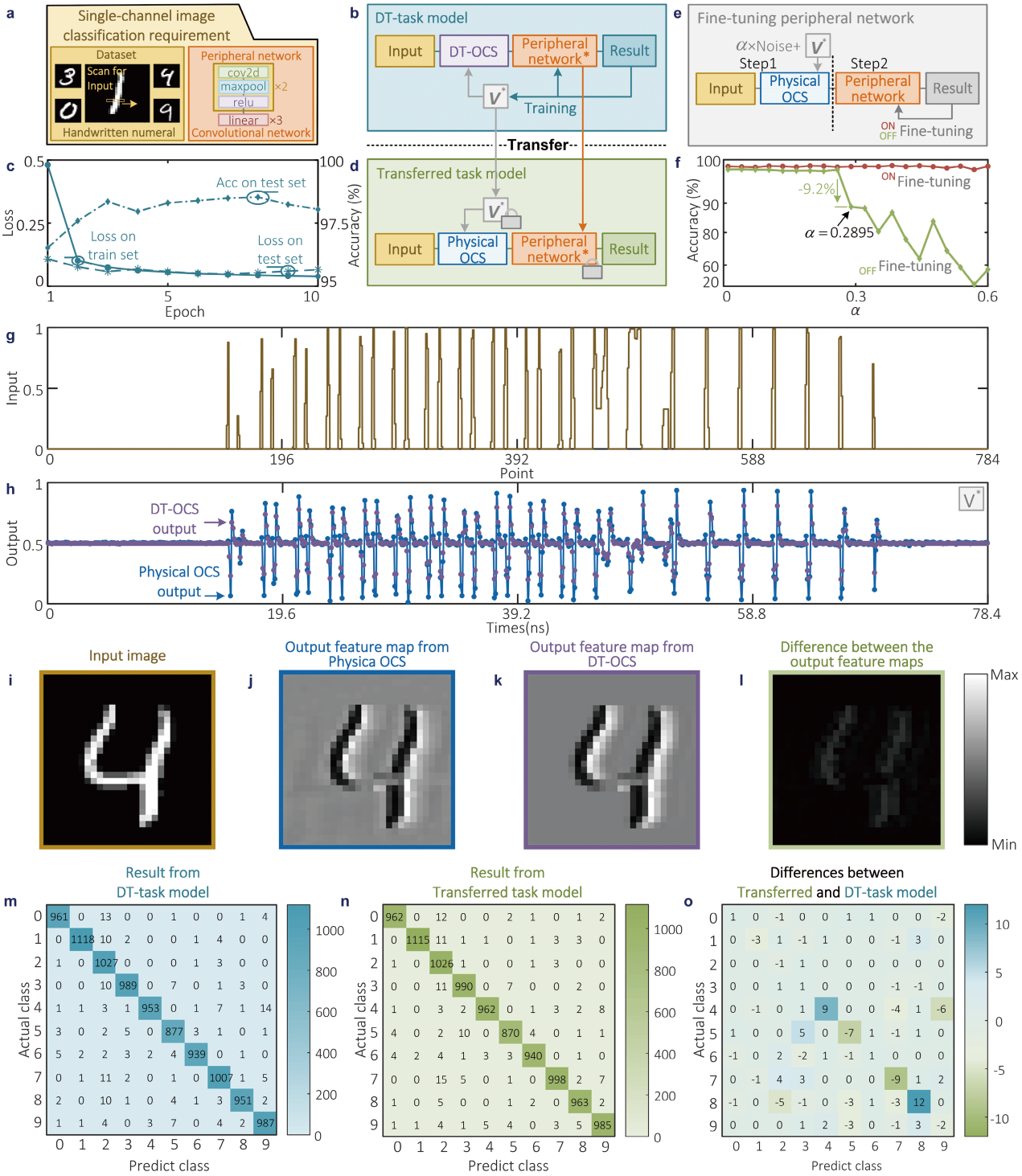


Fig. 4 | DT-OCS participates in the MNIST task. (a) Dataset and peripheral network used for the single-channel image classification task. (b) DT-task model composed of DT-OCS and the peripheral network. (c) Training process of the DT-task model, including the loss function on both the train and test sets, and classification accuracy on the test set across epochs. (d) Transferred task model. (e) Model that allows fine-tuning of the peripheral network based on the output of the physical OCS. (f) Classification accuracy fluctuation under different noise amplitudes, plotted on an exponential scale. The red line indicates the models allow further fine-tuning of peripheral networks, while the green line indicates those without fine-tuning. (g) A segment of the input sequence, derived from the original image “4” shown in the (i). (h) Outputs from DT-OCS and the physical OCS. (i) Original input image “4”. (j) Recovered feature map from the DT-OCS output. (k) Recovered feature map from the physical OCS output. (l) Difference between the DT-OCS and physical OCS output feature maps. (m) Confusion matrix of classification with the DT-task model. (n) Confusion matrix of classification with the transferred task model. (o) Difference between the confusion matrices of classification with the DT-task model and the transferred task model.

is offline and supports the application of the parameter transfer strategy.

Based on the optimized DT-task model obtained through training, we examine the role of joint optimization between the OCS configuration and the peripheral network. Specifically, a perturbation experiment is conducted on the optimized DT-task model, as shown in Fig. 4(e). Under the condition that the peripheral network parameters remain fixed (i.e., not fine-tuned), uniformly distributed random noise is added to the optimized configuration parameters $\boldsymbol{V}^*$ of DT-OCS. The classification accuracy of the DT-task network on the test set is then evaluated as a function of the average noise amplitude α. Each experiment is repeated 10 times, and the average classification accuracy is shown by the green line in Fig. 4(f). When the average noise amplitude α exceeds 0.072 V, the classification accuracy drops by 0.051%, which is greater than the performance gap (0.02%) between the optimized DT-task model and the transferred task model. When the noise amplitude increases further to 0.2895 V, the accuracy drops significantly by 9.2%, reaching 88.9%. These results indicate that the optimal configuration has converged to a task-relevant optimum, and deviations from this optimum cannot be compensated when the peripheral network is kept fixed, highlighting the essential contribution of the OCS configuration to the achieved performance. For comparison, we then consider a fine-tuning setting in which the training data are first processed using a DT-OCS with noise-corrupted configuration parameters, and then allow the peripheral network to be fine-tuned based on the resulting outputs. The corresponding accuracy is shown by the red curve in Fig. 4(f). Regardless of the noise amplitude added to $\boldsymbol{V}^*$, the fine-tuned peripheral network can compensate adaptively, and the final classification accuracy on the test set remains stable between 98.04% and 98.85%. This comparison suggests that, if the OCS does not participate in joint training with other parts of the model, its design value may be overridden by downstream adaptation through peripheral fine-tuning. In contrast, the approach proposed in this work optimizes the OCS configuration jointly with the peripheral network in the DT-task model and validates the transferred task model without any further fine-tuning, which is inherently different from the above fine-tuning scheme.

Furthermore, the adaptability of DT-OCS is evaluated in a more complex task scenario using a multi-channel color clothing and footwear classification task, as shown in Fig. 5(a). Compared to the single-channel grayscale MNIST task, this task involves increased input dimensionality and visual complexity. The dataset consists of five categories: pants, shoes, shorts, T-shirts, and undershirts, with 2306 RGB images for training and 254 images for testing. In addition to the presence of multiple color channels, variations in object size, position, and orientation further increase the complexity of the input structure, placing higher demands on feature extraction. To address these challenges, as shown in Fig. 5(b), we construct a task model consisting of three OCSs and a deep peripheral convolutional network. Specifically, each RGB image is decomposed into its monochrome components, which are line-scanned to generate three input sequences corresponding to the R, G, and B channels. These sequences are processed independently by three OCSs, each operating under its own configuration parameters. The outputs of the three OCSs are then reshaped to feature maps and interpreted by the peripheral network to perform final classification. Implementation details are provided in Supplementary Section 5. To accommodate the small input image size (30×30), the first two pooling layers of the original VGG11 architecture are removed, yielding the modified peripheral network used in this task.

We perform the training process using the DT-task model, during which all parameters of the OCSs and the peripheral network are jointly optimized. As illustrated in Fig. 5(c), the DT-task model converges after 50 training epochs and achieves a classification accuracy of 83.07% on the test set. Subsequently, as shown in Fig. 5(d), the optimal configuration parameters from the three DT-OCSs $\boldsymbol{V}^*=\begin{bmatrix}\boldsymbol{V}_1^*\\ \boldsymbol{V}_2^*\\ \boldsymbol{V}_3^*\end{bmatrix}=\begin{bmatrix}1.3153\text{ V}, 1.8178\text{ V}, 0.7617\text{ V}\\ 0.7987\text{ V}, 0.8428\text{ V}, 1.1512\text{ V}\\ 1.3595\text{ V}, 1.3239\text{ V}, 0.6055\text{ V}\end{bmatrix}$, together with the peripheral network parameters, are transferred to form the transferred task model. During the testing, representative input segments derived from an image of a pair of shorts are shown in Fig. 5(g), and the corresponding outputs from the DT-OCSs and the physical OCSs are presented in Fig. 5(h). The feature maps reconstructed from these outputs, along with their difference, are shown in Fig. 5(i). By combining the outputs of three DT-OCSs or three physical OCSs, color feature maps are reconstructed, as shown in Fig. 5(j), yielding an average PSNR of 35.18 dB and an SSIM of 0.94 over the test set. The transferred task model achieves the same test accuracy of 83.07% as the DT-task model. As shown in Fig. 5(k), the confusion matrices obtained using the DT-task model and the transferred task model are identical. These results demonstrate that multiple DT-OCSs can support full task training while the physical OCSs remain offline, and the optimized parameters can be directly transferred to the physical system without performance degradation. Additional comparisons using other selected peripheral networks are provided in Supplementary Section 6.

Similar to the MNIST task, we also examine the impact of DT-OCS configuration parameters on performance in the color image task as shown in Fig. 5(e). The experiment shows that, as the amplitude of uniformly distributed random noise added to the optimal configuration parameters $\boldsymbol{V}^*$ increases, the classification accuracy of the model without fine-tuning the peripheral network, shown by the green line in Fig. 5(f), begins to decline. When the average noise amplitude reaches 0.6316 V, the average classification accuracy on the test set over 10 independent tests drops by

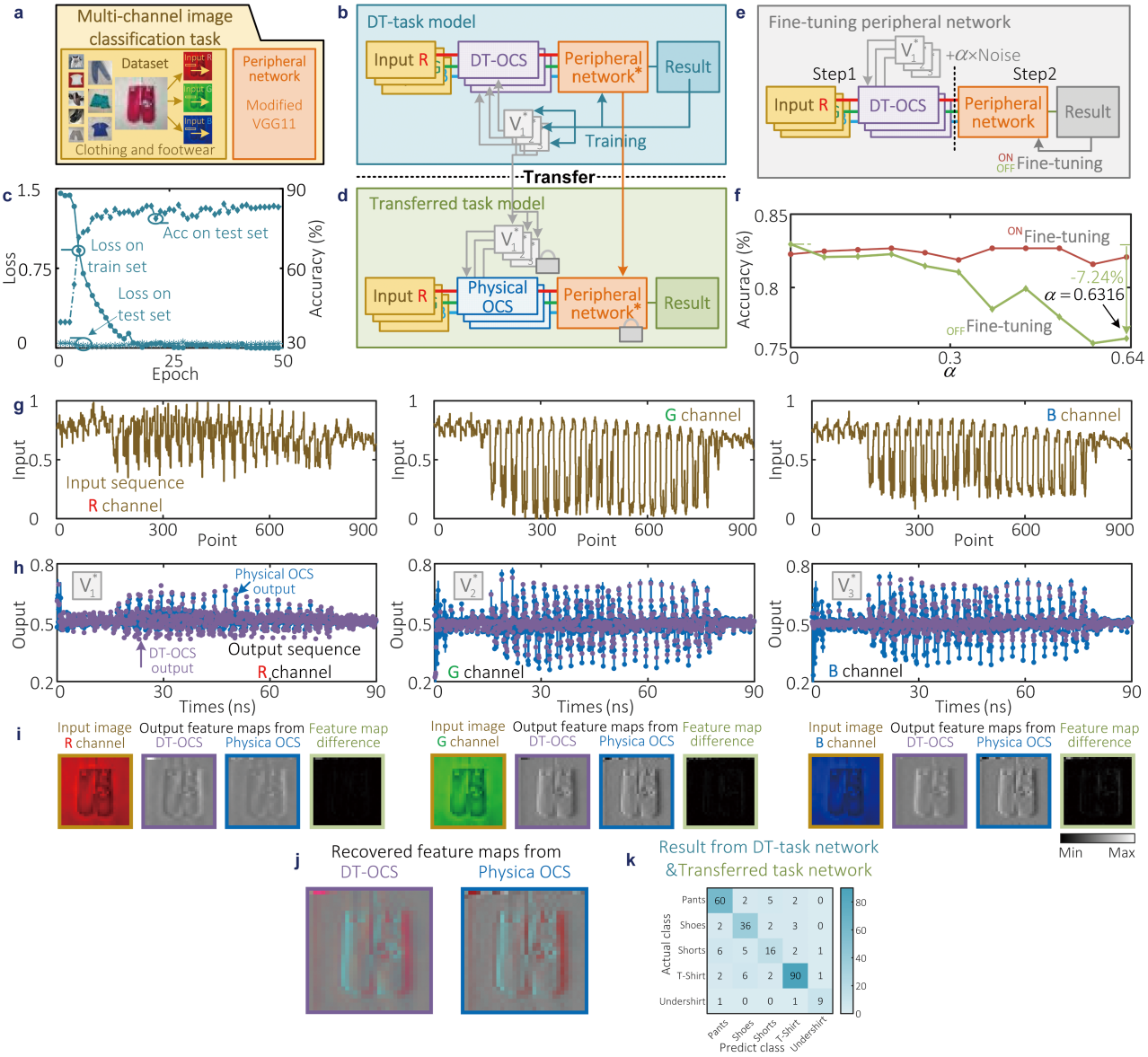


Fig. 5 | DT-OCS participates in the classification task for multi-channel color clothing and footwear images. (a) Dataset and peripheral network used for the multi-channel color image classification task. (b) DT-task model composed of multiple DT-OCSs and the peripheral network. (c) Training process of the DT-task model, including the loss on the train and test sets, and the classification accuracy on the test set across epochs. (d) Transferred task model. (e) Model that allows fine-tuning of the peripheral network based on the output of the physical OCS. (f) Classification accuracy fluctuation under different noise amplitudes, plotted on an exponential scale. The red line indicates the models allow further fine-tuning of peripheral networks, while the green line indicates those without fine-tuning. (g) Input sequences for the R, G, and B channels, from left to right, derived from the three monochrome components of the original image shown in the (i). (h) Outputs from DT-OCSs and the physical OCSs. (i) Monochrome channels of the original input image, the feature map reconstructed from the DT-OCS output, the feature map reconstructed from the physical OCS output, and the difference between the feature maps. From left to right, the three groups correspond to the R, G, and B color channels, respectively. (j) Color feature maps reconstructed by combining the outputs of three DT-OCSs or the physical OCSs. (k) Confusion matrix for the classification with the DT-task model or the transfer-task model.

more than 7.24%. Because the modified VGG11 used as the peripheral network has stronger tolerance to turbulence, this task model exhibits higher tolerance to OCS configuration parameter deviations than that of the MNIST task. In contrast, as shown by the red line in Fig. 5(f), the classification accuracy of the model that allows peripheral network fine-tuning consistently remains in the range of 81.7% to 82.8%.

To further validate the generality and transferability of DT-OCS in temporal decision-making scenarios, we consider a gold trading strategy task as shown in Fig. 6(a). In this task as shown in Fig. 6(b), a normalized gold price

sequence is supplied to the OCS, which processes the input sequence under a given configuration parameter. The output of the OCS is then compared against a predefined threshold to generate trading actions, where outputs exceeding the threshold trigger a buy action, while the remaining cases correspond to a hold decision. In contrast to the image tasks, no neural peripheral network is introduced. Instead, the action is generated directly from the OCS output through a simple thresholding operation. As a result, the decision latency is governed by the physical OCS-readout-thresholding chain, without requiring iterative digital inference. The objective of the task is to optimize the OCS configuration parameter such that the resulting actions yields higher expected returns over time. The quality of the generated action sequence is evaluated using a time-discounted reward function

$$G_n = \sum_{k=n+1}^{n+51} a^{k-n}(P_k - P_n)/50\ , \tag{7}$$

where n represents the action time point, k denotes a future time, and a is the discount factor, fixed at 0.8. P_k and P_n are the gold prices at the future time point k and action time point n, respectively, and their difference forms the sequence of future returns. The training objective is to optimize the configuration parameters of the OCS to maximize the average value of the optimization function $\overline{G} = \sum_n A_n G_n \big/ N$ over the train sequence, where A_n denotes the decision action at time point n, with values of 1 and 0 corresponding to buy and hold, respectively.

In our previous work[25], executing this task relied on the physical OCS combined with the particle swarm optimization (PSO) algorithm[40] to perform parameter search and optimization. This category of online optimization methods suffers from the following common limitations: 1) Training stages cannot run in parallel, as the training process completely occupies the physical OCS. The training data must be repeatedly loaded into the physical OCS. For each configuration, the system adjusts the voltage parameters, samples the output signal, and evaluates the profitability of that configuration. 2) The training efficiency is extremely low. For a PSO algorithm using 100 particles and 150 iterations, a total of 15,000 rounds of parameter loading and output sampling are required. Each round takes approximately 10 seconds, due to the need to wait for the optical computing chip in the physical OCS to return to thermal stability, and for the sampling device (OSC) to return to steady output. The total training time is approximately 1.74 days, and only one final configuration is usable. As a result, this approach is not suitable for rapid validation or broad reuse.

However, the use of DT-OCS for task training naturally avoids occupying the physical OCS and significantly improves training speed. More importantly, since the core structure of DT-OCS is based on a Dynamic LSTM cell that supports error backpropagation, gradient descent methods can be employed for task optimization instead of the time-consuming PSO method. With DT-OCS, the training process illustrated in Fig. 6(c) is executed. First, a grid search is used to initialize the configuration parameters $\boldsymbol{V}$. The entire three-dimensional space of $\boldsymbol{V}$ (as the physical OCS

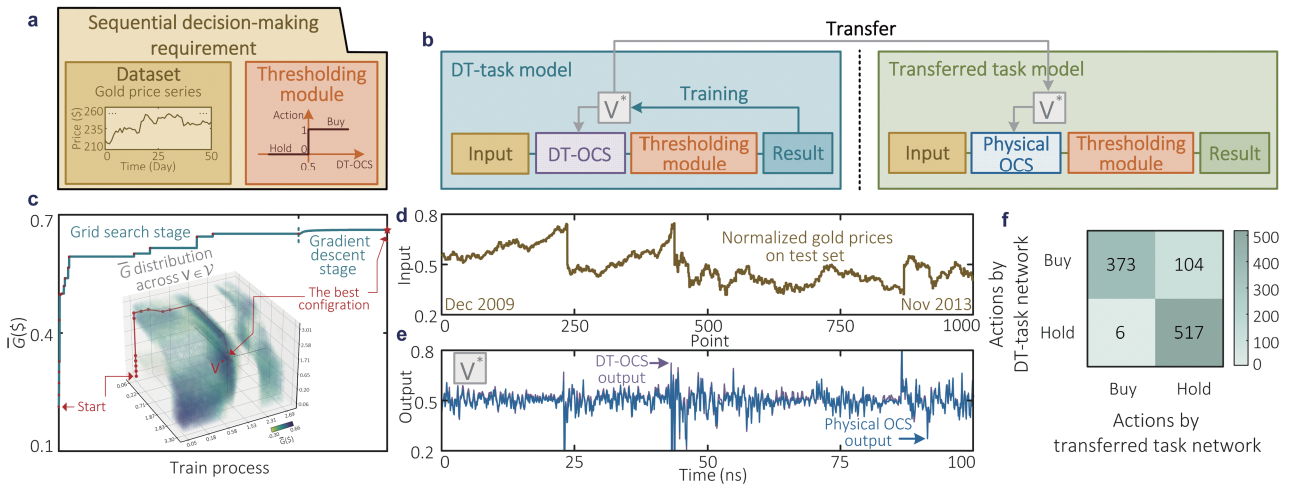


Fig. 6 | Participation of the DT-OCS in the gold trading strategy task. (a) Input price sequence and thresholding module used in the gold trading task. (b) DT-task model and transfer-task model composed of DT-OCS or physical OCS and the thresholding module. (c) Training process of the DT-task model, including grid search and gradient descent. During training, the optimization function $\overline{G}$ increases progressively, exhibiting a staircase-like increases during the grid search stage and reaching its final maximum through gradient descent. The inset shows the distribution of the optimization function $\overline{G}$ over the entire configuration parameter space on the training sequence. The evolution of the configuration point during training is illustrated by the red trajectory, converging toward the optimal configuration marked by a star. (d) Input test sequence. (e) Outputs of the physical OCS and DT-OCS on the test sequence. (f) Confusion matrix between the actions generated by the physical OCS and DT-OCS.

contains three actively controlled phase shifters, $\boldsymbol{V}$ takes value in a three-dimensional space) is sampled with 15 points along each dimension, resulting in an initial pool of 15^3=3375 candidate configurations. It takes only about 9 minutes to identify the best-performing initial solution on a personal computer. Subsequently, 200 rounds of gradient descent with error backpropagation are performed, requiring only 100 seconds. As illustrated in Fig. 6(c), the optimization function $\overline{G}$ increases progressively during training, exhibiting a staircase-like improvement during the grid search stage and converging to its maximum value through gradient descent on train price sequence. This process yields the optimal configuration parameters $\boldsymbol{V}^* = [2.2511\ \mathrm{V}, 1.6382\ \mathrm{V}, 0.9495\ \mathrm{V}]$, and DT-task model achieves the maximum value of the optimization function $\overline{G}$=\$3.059 on the test price sequence. The corresponding evolution of the configuration in the parameter space is visualized in the inset of Fig. 6(c), where the trajectory converges toward the optimal configuration indicated by a red star. Figure 6(d) shows the normalized input test price sequence. In Fig. 6(e), the purple line represents the DT-OCS output under the optimal configuration. The same optimal configuration parameters $\boldsymbol{V}^*$ are then transferred directly to the physical OCS for testing. The same test sequence is used, and the blue line in Fig. 6(e) shows the output from the physical OCS. The outputs of DT-OCS and the physical OCS exhibit notable similarity. The confusion matrix between the actions derived from the DT-OCS and the physical OCS is shown in Fig. 6(f). On the test sequence, 89.0% of the output actions are classified as the same. The optimization function value obtained by the physical OCS on the test sequence is $\overline{G}$= \$2.877, and the relative difference from the value obtained by DT-OCS is within 5.9%. These results indicate that DT-OCS supports training for temporal strategy tasks. For this task, the total training time is reduced from 1.74 days (using the physical OCS with standard PSO) to only 10.7 minutes, which is less than 0.4% of the original time. By introducing DT-OCS, the inherent limitation of the physical OCS in supporting parallel training and testing is resolved, and training time is significantly reduced.

Through these three tasks, we demonstrate that DT-OCS can fully support task training while the physical OCS remains offline. Because DT-OCS is implemented as a code-based digital model, its structure and parameters can be readily replicated, enabling multiple DT-OCS instances to be instantiated simultaneously across independent research threads or users. This allows different tasks or task variants to be developed and trained independently, without contention for the physical OCS, thereby naturally supporting parallel and reproducible task development. To facilitate further validation and reuse, the three DT-task networks and the initial datasets used in this work are open-sourced, with links provided at the end of the paper.

4 Discussion

4.1 Residual between physical OCS and DT-OCS

The residual between the output of the physical OCS and that of its digital twin model, DT-OCS, serves as a key metric for evaluating both the computational reliability of the OCS and the accuracy of the DT-OCS framework. The origins of the residual can be attributed to two primary sources: 1) Model approximation error. DT-OCS approximates the computational function of the physical OCS by mapping voltage configuration $\boldsymbol{V}$ to the internal parameters of the Dynamic LSTM through a function block. In the current implementation, this function block adopts a single-layer linear mapping, as illustrated in the inset of Fig. 7(a). The resulting residual distribution on the test set, together with its Voigt profile fitting shown by the red curve, is shown in Fig. 7(a) (identical to Fig. 3(e)). While this linear design offers high computational efficiency and stable training behavior, it may not fully capture nonlinear coupling mechanisms among configuration parameters in the physical OCS. A representative example is thermal crosstalk among thermo-optic phase shifters, where localized temperature variations induced by different voltage configurations lead to heat diffusion and mutual interference, giving rise to nonlinear parameter coupling that is difficult to represent using a purely linear mapping[41,42]. To examine the impact of enhanced parameter mapping, we introduce a more complex function block by inserting an intermediate nonlinear transformation between two linear layers, as illustrated in Fig. 7(b). The detailed layer architecture and parameter dimensions of both the linear and nonlinear function blocks are explicitly provided in the Supplementary Section 3. Using this enhanced function block, we re-evaluate the residual distribution and again fit the distribution with a Voigt profile. The standard deviation of the Gaussian component is $\sigma_{\mathrm{Gaussian}} = 3.41 \times 10^{-3}$, and that of the Lorentzian component is $\sigma_{\mathrm{Lorentzian}} = 3.25 \times 10^{-3}$. Compared to the single-layer configuration, the standard deviation of Gaussian component of the fitted distribution increases by 5.24%, while the standard deviation of Lorentzian component decreases by 32.57%, indicating a more Gaussian-like distribution with suppressed long-tail behavior. As shown by the yellow curve in Fig. 7(b), the residuals obtained with the nonlinear complex function block are more concentrated than those of the simple linear design (red dashed line), demonstrating improved fitting fidelity. However, this improvement comes at the cost of increased computational complexity. A quantitative comparison of the associated parameter counts and computational complexity is provided in the Supplementary Section 3. To balance modeling accuracy and efficiency, the simplified linear function block is adopted as the primary configuration in this work, while the nonlinear alternative is also provided for readers inter-

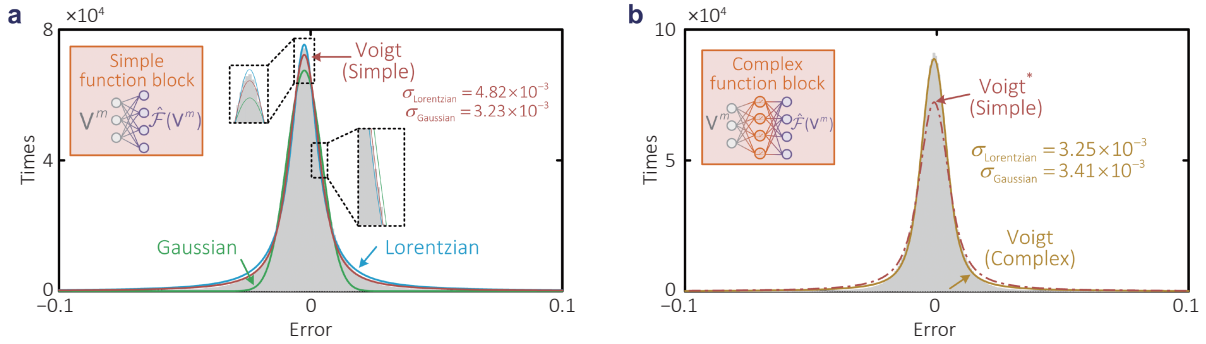


Fig. 7 | Residual distribution and fitting results. (a) Residual distribution and fitting results for the simple function block. The gray histogram shows the residual distribution. The green, blue, and red curves represent the fitted Gaussian, Lorentzian, and Voigt profiles, respectively. The orange inset illustrates the structure of the simple function block. (b) Residual distribution and fitting result for the complex function block. The yellow curve represents the Voigt profile fit. For comparison, the Voigt fit corresponding to the simple function block is plotted as a red dashed line, aligned by shifting its center. The orange inset shows the structure of the complex function block. The first linear transformation in the complex function block is followed by a nonlinear activation before the subsequent linear mapping.

ested in exploring this trade-off further. A quantitative comparison of fitting statistics for both designs on the training and test sets is reported in Supplementary Section 3. Implementations of both designs are provided in the open-sourced codebase referenced in the Code Availability section. 2) Physical noise from hardware. Residuals also arise from stochastic noise sources inherent to the physical OCS hardware. Signal generation, modulation, and readout components, including the AWG, modulation amplifiers, high-speed phase modulators, PD, and associated electronic circuits, introduce timing jitter, thermal noise, and random fluctuations. Instabilities in waveguides and coupling fibers can further lead to phase noise, while optical amplification contributes to gain noise. These physical noise sources often manifest as statistically distributed disturbances, yet they are difficult to capture accurately using a single general model or fixed functional form. In summary, the residuals arise from a combination of model approximation error under practical design choices and hardware-induced physical noise mechanisms. Figure 7(a) shows that the Voigt profile (red line) matches the residual distribution more closely than either a purely Gaussian (green line) or Lorentzian (blue line) profile. This likely results from a combination of the aforementioned factors. Although the architecture and operation differ, MZI mesh-based structures in coherent optical systems have exhibited similar residual behavior[6]. Once the residual distribution is obtained, noise following the same distribution can be injected into the DT-OCS output via rejection sampling. This offers a flexible mechanism to incorporate measured noise into the digital twin framework when robustness evaluation or data augmentation is desired. The concrete implementation details of the noise-generation module are provided in the Code Availability section.

4.2 Nonlinear response in OCS

To investigate potential nonlinear response characteristics that may arise during OCS operation, we design an excitation experiment with inputs of identical shape but varying amplitude, as illustrated in Fig. 8(a). The system response is analyzed using DT-OCS. Specifically, as shown in Fig. 8(b), we fix the shape of the input signal, a threshold signal with a width of 8, and gradually increase its amplitude by adjusting the amplification factor Amp, over the range from 0 to 1. The configuration parameter is selected from the supervised learning dataset ($\boldsymbol{V}^{2000}$).

In an ideal linear system, responses to inputs with the same shape but different amplitudes should completely overlap after bias removal and normalization (i.e., division by the amplification factor Amp). However, as shown in Fig. 8(c), after the same bias removal and normalization, the DT-OCS output exhibits significant variations not only in amplitude but also in shape and peak position as the input amplitude increases. Additionally, the gray lines in Fig. 8(c) indicate the DT-OCS outputs at two specific positions P=10 and P=18, whose raw bias-removed outputs as functions of the input amplitude are shown by the colored curves in Fig. 8(d) and 8(e), respectively. As the input amplitude increases, the two positions show markedly different nonlinear behaviors in their raw output amplitudes. The same experiment is repeated on the physical OCS, where the corresponding raw bias-removed outputs are shown as discrete points in Fig. 8(d) and 8(e). Despite measurement noise and deviations, the physical OCS outputs exhibit nonlinear response trends consistent with those of the DT-OCS, with the two positions again showing markedly different nonlinear behaviors. These observations collectively demonstrate the presence of complex internal nonlinear mechanisms within the system. Possible sources of the observed nonlinearity include phase-to-intensity mapping, non-ideal behavior of high-speed elec-

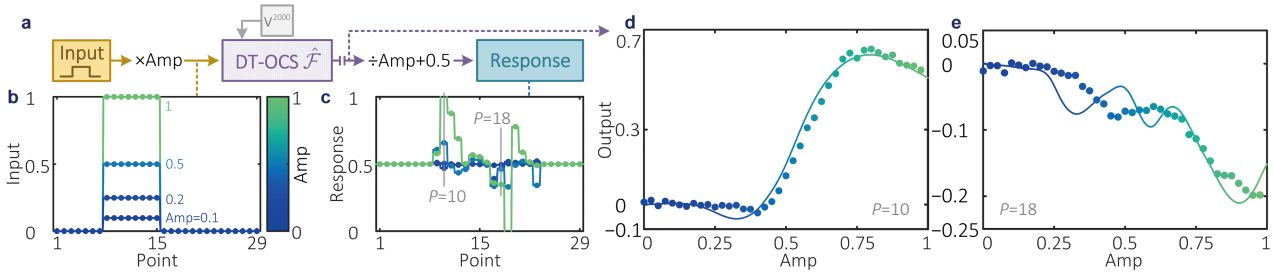


Fig. 8 | Validation of nonlinear response. (a) Experimental setup for nonlinear response validation. (b) Input signals with identical shape but varying amplitudes. A color gradient indicates increasing amplitude. (c) Normalized responses corresponding to different input amplitudes. Two gray lines mark the responses at positions *P*=10 and *P*=18. (d) Relationship between the bias-removed raw output and input amplitude at *P*=10, with DT-OCS results shown as colored curves and physical OCS measurements shown as discrete points. (e) Relationship between the bias-removed raw output and input amplitude at *P*=18.

tro-optic modulators, and saturation effects in the modulation amplifier. DT-OCS captures and reproduces these nonlinear response characteristics, enabling system-level modeling for task training and parameter optimization.

4.3 Temporal synchronization and input-output alignment

Accurate modeling and deployment of high-speed OCSs require a well-defined timing relationship between the input and output signals, which in practice is established through system-level synchronization between signal generation and acquisition. In our experimental setup, the AWG serves as both the signal source and the timing reference. It generates the input signal and provides a trigger signal to the oscilloscope, establishing stable synchronization during signal acquisition. To provide a consistent temporal reference across measurements, a predefined threshold-based header pattern is inserted before each data sequence and used as the trigger signal for the oscilloscope. The rising edge of this header acts as a temporal alignment marker. In the recorded output signal, the corresponding initial rise of the measured output response indicates the onset of the system's causal response, consistent with the beginning of the receptive field marked by the left gray dashed line in Fig. 3(c), thus establishing a well-defined input-output correspondence. By fixing this reference point, the portion of the output response corresponding to the data payload in the header-augmented input signal can be consistently identified, enabling reliable subsequent segmentation of the continuous output signal into valid output sequences. An illustrative example is provided in Supplementary Section 6. This input-output correspondence is determined during the receptive-field measurement stage and is subsequently maintained throughout data acquisition, training, and inference, ensuring causal consistency between the physical OCS and its digital twin. This mechanism effectively avoids causal ambiguity in high-speed OCS operation and forms the basis for reliable DT-OCS modeling.

4.4 System-level scope and hardware adaptation of DT-OCS

In this work, DT-OCS is instantiated on a representative high-speed silicon-based optical computing system to validate the feasibility of the proposed framework. This implementation demonstrates that the DT-OCS methodology can be constructed and deployed on a practical optical computing platform. DT-OCS models the end-to-end behavior of the assembled system under a given hardware configuration, thereby reflecting the system-level characteristics of the underlying physical platform.

In practical operation, hardware configuration may change due to component replacement, system upgrade, or architectural modification. The impact of such changes on DT-OCS depends on the scale and structural level of the modification. When optoelectronic components are replaced by devices with the same specifications and operating characteristics, the system-level input-output behavior is expected to remain consistent, and the trained DT-OCS remains applicable. If replacement introduces changes in device-level characteristic parameters, such as bandwidth, noise level, gain, or nonlinear response, that in turn alter the system-level input-output behavior, the digital twin can be updated by re-measuring system-level input-output data and re-training the DT-OCS while retaining the same modeling framework.

When the optical processor is redesigned into a different computational architecture, the underlying computational function changes, and the previously trained DT-OCS instance no longer corresponds to the updated system. In this case, as long as the updated optical computing system defines a causal input-output mapping governed by configuration parameters, a corresponding DT-OCS can be constructed under the same measurement-driven framework. Although different platforms may require adjustments in modeling details or training procedures due to variations in physical implementation, the overall methodology and workflow remain unchanged. In this sense, DT-OCS should be understood as a reproducible framework for

encapsulating the configuration-dependent behavior of an optical computing system into a digital operator, supporting systematic modeling and integration of optical processors within hybrid optoelectronic computing systems.

5 Conclusions

In this work, we proposed a digital twin-based development framework for optical computing systems, termed DT-OCS, to address the fundamental limitations of existing OCS architectures in supporting offline model development, reproducible benchmarking, and parallel task development. By constructing a system-level digital surrogate of a specific physical OCS through supervised learning on experimentally measured input-output data, DT-OCS accurately reproduces the computational behavior of the physical hardware while remaining fully deployable back to the original system.

Using DT-OCS, across representative tasks, complete task models that comprise the OCS together with the peripheral networks can be optimized entirely in the digital domain, without occupying the physical hardware. Subsequently, the optimized configuration parameters obtained via DT-OCS are directly transferred to the physical OCS without any hardware-in-the-loop training or post-deployment fine-tuning. The transferred models exhibit performance nearly identical to their digital counterparts. On the test sets of the MNIST handwritten digit recognition and color clothing and footwear classification tasks, the classification accuracy differences between the transferred models and their digital counterparts are 0.02% and 0, respectively. For the gold trading temporal strategy task, the deviation in profitability on the test sequence is controlled within 5.9%. These results confirm the fidelity and transferability of DT-OCS-based offline training.

Beyond task-level validation, DT-OCS provides system-level insights into the physical OCS. By analyzing residual distributions between DT-OCS and the physical system, we characterize the combined effects of model approximation error and hardware-induced noise on the system output, and quantify the accuracy-efficiency trade-off in digital twin construction. DT-OCS further captures nonlinear response behaviors and preserves temporal causality, enabling reliable modeling of complex system dynamics that are difficult to address using idealized or component-level models.

By encapsulating a physical OCS as a reproducible and callable digital model, DT-OCS fundamentally transforms optical computing systems from hardware-bound, serialized experimental platforms into software-accessible computing resources. This paradigm enables offline development, parallel exploration across tasks and users, and reproducible validation of optical computing capabilities. The fully open-sourced DT-OCS framework and datasets provided in this work offer a practical foundation for future research and the development of more complex hybrid optoelectronic computing systems.

References

1. Meng XY, Zhang GJ, Shi NN et al. Compact optical convolution processing unit based on multimode interference. *Nat Commun* **14**, 3000 (2023).
2. Wang DL, Nie YK, Hu GL et al. Ultrafast silicon photonic reservoir computing engine delivering over 200 TOPS. *Nat Commun* **15**, 10841 (2024).
3. Xu XY, Tan MX, Corcoran B et al. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature* **589**, 44–51 (2021).
4. Zhang WK, Wu B, Gu WT et al. Large-scale optical programmable logic array for two-dimensional cellular automaton. *Adv Photonics* **6**, 056007 (2024).
5. Bandyopadhyay S, Sludds A, Krastanov S et al. Single-chip photonic deep neural network with forward-only training. *Nat Photonics* **18**, 1335–1343 (2024).
6. Hua SY, Divita E, Yu SS et al. An integrated large-scale photonic accelerator with ultralow latency. *Nature* **640**, 361–367 (2025).
7. Huang YY, Liu WC, Sun R et al. Diffraction-driven parallel convolution processing with integrated photonics. *Laser Photonics Rev* **19**, 2400972 (2025).
8. Liu WC, Huang YY, Sun R et al. Ultra-compact multi-task processor based on in-memory optical computing. *Light Sci Appl* **14**, 134 (2025).
9. Xu ZH, Zhou TK, Ma MZ et al. Large-scale photonic chiplet Taichi empowers 160-TOPS/W artificial general intelligence. *Science* **384**, 202–209 (2024).
10. Zhu HH, Zou J, Zhang H et al. Space-efficient optical computing with an integrated chip diffractive neural network. *Nat Commun* **13**, 1044 (2022).
11. Luo XG. Digital optics and optical intelligent agent. *Appl Phys Lett* **127**, 010501 (2025).
12. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* **521**, 436–444 (2015).
13. Abadi M, Agarwal A, Barham P et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv: 1603.04467, 2016. https://doi.org/10.48550/arXiv.1603.04467
14. Mattson P, Reddi VJ, Cheng C et al. MLPerf: an industry standard benchmark suite for machine learning performance. *IEEE Micro* **40**, 8–16 (2020).
15. Bergstra J, Bardenet R, Bengio Y et al. Algorithms for hyper-parameter optimization. In *Proceedings of the 25th International Conference on Neural Information Processing Systems* 2546–2554 (Curran Associates Inc. , 2011).
16. Li LS, Jamieson K, DeSalvo G et al. Hyperband: a novel bandit-based approach to hyperparameter optimization. *J Mach Learn Res* **18**, 1–52 (2018).
17. Miller DAB. Perfect optics with imperfect components. *Optica* **2**, 747–750 (2015).
18. Shastri BJ, Tait AN, De lima TF et al. Photonics for artificial intelligence and neuromorphic computing. *Nat Photonics* **15**, 102–114 (2021).
19. Wright LG, Onodera T, Stein MM et al. Deep physical neural networks trained with backpropagation. *Nature* **601**, 549–555 (2022).
20. Wan YJ, Liu XD, Wu GZ et al. Efficient stochastic parallel gradient descent training for on-chip optical processor. *Opto-Electron Adv* **7**, 230182 (2024).
21. Han YN, Xiang SY, Song ZW et al. Pattern recognition in multi-synaptic photonic spiking neural networks based on a DFB-SA chip. *Opto-Electron Sci* **2**, 230021 (2023).
22. Feldmann J, Youngblood N, Karpov M et al. Parallel convolutional processing using an integrated photonic tensor core. *Nature* **589**, 52–58 (2021).

23. Zhao BH, Cheng JW, Wu B et al. Integrated photonic convolution acceleration core for wearable devices. *Opto-Electron Sci* **2**, 230017 (2023).
24. Xia F, Kim K, Eliezer Y et al. Nonlinear optical encoding enabled by recurrent linear scattering. *Nat Photonics* **18**, 1067–1075 (2024).
25. Fu TZ, Zang YB, Huang YY et al. Photonic machine learning with on-chip diffractive optics. *Nat Commun* **14**, 70 (2023).
26. Wu TW, Menarini M, Gao ZH et al. Lithography-free reconfigurable integrated photonic processor. *Nat Photonics* **17**, 710–716 (2023).
27. Wu CM, Jiao ZY, Deng HQ et al. Reconfigurable inverse-designed phase-change photonics. *APL Photonics* **10**, 016113 (2025).
28. Cheng JW, Huang CR, Zhang JL et al. Multimodal deep learning using on-chip diffractive optics with in situ training capability. *Nat Commun* **15**, 6189 (2024).
29. Pai S, Sun ZH, Hughes TW et al. Experimentally realized in situ backpropagation for deep learning in photonic neural networks. *Science* **380**, 398–404 (2023).
30. Pérez-López D, López A, DasMahapatra P et al. Multipurpose self-configuration of programmable photonic circuits. *Nat Commun* **11**, 6359 (2020).
31. Bogaerts W, De Heyn P, Van Vaerenbergh T et al. Silicon microring resonators. *Laser Photon Rev* **6**, 47–73 (2012).
32. Zhang WP, Huang CR, Peng HT et al. Silicon microring synapses enable photonic deep learning beyond 9-bit precision. *Optica* **9**, 579–584 (2022).
33. Fu TZ, Huang YY, Sun R et al. Integrated diffractive optical neural network with space-time interleaving. *Chin Opt Lett* **21**, 091301 (2023).
34. Sun R, Li YM, Fu TZ et al. High-speed and low-latency optical feature extraction engine based on diffraction operators. *Adv Photonics Nexus* **4**, 056012 (2025).
35. Shen YC, Harris NC, Skirlo S et al. Deep learning with coherent nanophotonic circuits. *Nat Photonics* **11**, 441–446 (2017).
36. Bai BW, Yang QP, Shu HW et al. Microcomb-based integrated photonic processing unit. *Nat Commun* **14**, 66 (2023).
37. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* **9**, 1735–1780 (1997).
38. Kong J, Kim J, Bae J. HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* 1428 (Curran Associates Inc. , 2020).
39. Yamamoto R, Song E, Kim JM. Parallel WaveGAN: a fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In: *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 6199–6203 (IEEE, 2020). http://doi.org/10.1109/ICASSP40776.2020.9053795.
40. Kennedy J, Eberhart R. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks* 1942–1948 (IEEE, 1995). http://doi.org/10.1109/ICNN.1995.488968.
41. Ning NN, Zhang Q, Huang QK et al. Thermal flux manipulation on the silicon photonic chip to suppress the thermal crosstalk. *APL Photonics* **9**, 046108 (2024).
42. Teofilovic I, Cem A, Sanchez-Jacome D et al. Thermal crosstalk modelling and compensation methods for programmable photonic integrated circuits. *J Lightwave Technol* **42**, 7816–7824 (2024).

Acknowledgements

The authors wish to acknowledge the support of National Key Research and Development Program of China (2024YFE0203600) and National Natural Science Foundation of China (NSFC) (62135009, 62405370)

Author contributions

H.C., T.F. and R.S. conceived the project. H.C. supervised the study. R.S. and Y.L. designed the DT-OCS and built the optical system. R.S. and Y.L. processed and analyzed the captured data in the experiment. R.S., Y.L., T.F. and H.C. participated in the writing of the paper. S.Y., T.F., and W.L. participated in the discussion.

Competing interests

The authors declare no competing financial interests.

Data availability

The data that support the results of this study and other findings of the study are available from the corresponding author upon reasonable request.

Code availability

The code and datasets used to construct DT-OCS and carry out the tasks can be found at https://cloud.tsinghua.edu.cn/d/146f72dd4f2b46f394bd/, where they are freely available for download. We sincerely invite readers to test and deploy them, integrate DT-OCS into more complex network architectures, validate it under conditions where physical OCS is offline, or explore other properties of the system.

Computational resources

The training of DT-OCS is conducted on a server equipped with dual Intel Xeon Platinum 6164 processors and NVIDIA RTX 3090 GPUs. The image classification tasks are also trained on the same server. In contrast, the sequence-based tasks are performed on a personal computer powered by an Intel 12400 CPU without a dedicated GPU.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/
©The Author(s) 2026.
Published by Editorial Office of *Opto-Electronic Advance*, Institute of Optics and Electronics, Chinese Academy of Sciences.

Opto-Electronic Journals Group
www.oejournal.org
Scan for Article PDF