scientific reports

Check for updates

OPEN

Real-time multi-task diffractive deep neural networks via hardware-software co-design

Yingjie Li, Ruiyang Chen, Berardi Sensale-Rodriguez, Weilu Gao✉ & Cunxi Yu✉

Deep neural networks (DNNs) have substantial computational requirements, which greatly limit their performance in resource-constrained environments. Recently, there are increasing efforts on optical neural networks and optical computing based DNNs hardware, which bring significant advantages for deep learning systems in terms of their power efficiency, parallelism and computational speed. Among them, free-space diffractive deep neural networks (D^2NNs) based on the light diffraction, feature millions of neurons in each layer interconnected with neurons in neighboring layers. However, due to the challenge of implementing reconfigurability, deploying different DNNs algorithms requires re-building and duplicating the physical diffractive systems, which significantly degrades the hardware efficiency in practical application scenarios. Thus, this work proposes a novel hardware-software co-design method that enables first-of-its-like *real-time* multi-task learning in $D^2$2NNs that automatically recognizes which task is being deployed in real-time. Our experimental results demonstrate significant improvements in versatility, hardware efficiency, and also demonstrate and quantify the robustness of proposed multi-task D^2NN architecture under wide noise ranges of all system components. In addition, we propose a domain-specific regularization algorithm for training the proposed multi-task architecture, which can be used to flexibly adjust the desired performance for each task.

The past half-decade has seen unprecedented growth in machine learning with deep neural networks (DNNs). Use of DNNs represents the state-of-the-art in many applications, including large-scale computer vision, natural language processing, and data mining tasks[1–3]. DNNs have also impacted practical technologies such as web search, autonomous vehicles, and financial analysis[1]. However, DNNs have substantial computational and memory requirements, which greatly limit their training and deployment in resource-constrained (e.g., computation, I/O, and memory bounded) environments. To address these challenges, there has been a significant trend in building high-performance DNNs hardware platforms. While there has been significant progress in advancing customized silicon DNN hardware (ASICs and FPGAs)[2,4] to improve computational throughput, scalability, and efficiency, their performance (speed and energy efficiency) are fundamentally limited by the underlying electronic components. Even with the recent progress of integrated analog signal processors in accelerating DNNs systems which focus on accelerating matrix multiplication, such as Vector Matrix Multiplying module (VMM)[5], mixed-mode Multiplying-Accumulating unit (MAC)[6–8], resistive random access memory (RRAM) based MAC[9–13], etc., the parallelization are still highly limited. Moreover, they are plagued by the same limitations of electronic components, with additional challenges in the manufacturing and implementation due to issues with device variability[10,12].

Recently, there are increasing efforts on optical neural networks and optical computing based DNNs hardware, which bring significant advantages for machine learning systems in terms of their power efficiency, parallelism and computational speed[14–23]. Among them, free-space *diffractive deep neural networks* (D^2NNs) , which is based on the light diffraction, feature millions of neurons in each layer interconnected with neurons in neighboring layers. This ultrahigh density and parallelism make this system possess fast and high throughput computing capability. Note that the diffractive propagations controlled by such physical parameters are differentiable, which means that such parameters can be optimized via conventional backpropagation algorithms[16,18,19] using `autograd` mechanism[24].

In terms of hardware performance/complexity, one of the significant advantages of D^2NNs is that such a platform can be scaled up to millions of artificial neurons. In contrast, the design and DNNs deployment complexity on other optical architectures, e.g., integrated nantophotnics[14,25] and silicon photnics[23]), can dramatically

Electrical and Computer Engineering Department, University of Utah, 50 S Central Campus Road, Salt Lake City, UT 84112, USA. ✉email: weilu.gao@utah.edu; cunxi.yu@utah.edu

increase. For example, Lin et al.[16] experimentally demonstrated various complex functions with an all-optical D^{2}NNs. In conventional DNNs, forward prorogation are computed by generating the feature representation with floating-point weights associated with each neural layer. In D^{2}NNs, such floating-point weights are encoded in the phase of each neuron of diffractive phase masks, which is acquired by and multiplied onto the light wavefunction as it propagates through the neuron. Similar to conventional DNNs, the final output class is predicted based on generating labels according to a given one-hot representation, e.g., the max operation over the output signals of the last diffractive layer observed by detectors. Recently, D^{2}NNs have been further optimized with advanced training algorithms, architectures, and energy efficiency aware training[18,19,26], e.g, class-specific differential detector mechanism improves the testing accuracy by 1–3%[19,26] improves the robustness of D^{2}NNs inference with data augmentation in training.

However, due to the challenge of implementing reconfigurability in D^{2}NNs (e.g., 3D printed terahertz system[16]), deploying a different DNNs algorithm requires re-building the entire D^{2}NNs system. In this manner, the hardware efficiency can be significantly degraded for multiple DNNs tasks, especially when those tasks are different but related. This has also been an important trend in conventional DNNs, which minimizes the total number of neurons and computations used for multiple related tasks to improve hardware efficiency, namely *multi-task learning*[27]. Note that, realizing different tasks directly from the input data features without separate inputs or user indications is challenging even in conventional DNNs system. In this work, we present the first-of-its-kind real-time multi-task D^{2}NNs architecture optimized in hardware-software co-design fashion, which enables sharing partial feature representations (physical layers) for multiple related prediction tasks. More importantly, our system can automatically recognize which task is being deployed and generate corresponding predictions in real-time fashion, without any external inputs in addition to the input images. Moreover, we demonstrate that the proposed hardware-software co-design approach is able to significantly reduce the complexity of the hardware by further reusing the detectors and maintain the robustness under multiple system noises. Finally, we propose an efficient domain-specific regularization algorithm for training multi-task D^{2}NNs, which offers flexible control to balance the prediction accuracy of each task (task accuracy trade-off) and prevent overfitting. The experimental results demonstrate that our multi-task D^{2}NNs system can achieve the same accuracy for both tasks compared to the original D^{2}NNs, with more than 75% improvements in hardware efficiency; and the proposed architecture is practically noise resilient under detector Gaussian noise and fabrication variations, where prediction performance degrades ≤ 1% within the practical noise ranges.

Results and discussion

Figure 1 shows the proposed real-time multi-task diffractive deep neural network (D^{2}NN) architecture. Specifically, in this work, our multi-task D^{2}NN deploy image classification DNN algorithms with two tasks, i.e., classifying MNIST10 dataset and classifying Fashion-MNIST10 dataset. In a single-task D^{2}NN architecture for classification[16], the number of opto-electronic detectors positioned at the output of the system has to be equal to the number of classes in the target dataset. The predicted classes are generated similarly as conventional DNNs by selecting the index of the highest probability of the outputs (`argmax`), i.e., the highest energy value observed by detectors. Moreover, due to the lack of flexibility and reconfigurability of the D^{2}NN layers, deploying DNNs algorithms for N tasks requires physically designing N D^{2}NN systems, which means N times of the D^{2}NN layer fabrications and the use of detectors. Our main goal is to improve the cost efficiency of hardware systems while deploying multiple related ML tasks. Conceptually, the methodologies behind multi-task D^{2}NN architecture and conventional multi-task DNNs are the same, i.e., maximizing the shared knowledge or feature representations in the network between the related tasks[27].

Let the D^{2}NN multi-task learning problem over an input space $\mathcal{X}$, a collection of task spaces $\mathcal{Y}^n_{n\in[0,N]}$, and a large dataset including data points $\{x_i, y_i^1, \ldots, y_i^N\}_I \in [D]$, where N is the number of tasks and D is the size of the dataset for each task. The hypothesis for D^{2}NN multi-task learning remains the same as conventional DNNs, which generally yields the following empirical minimization formulation:

$$\min_{\theta^{share},\theta^1,\theta^2,\ldots,\theta^N} \quad \sum_{n=1}^{N} c^t \mathcal{L}(\theta^{share}, \theta^t) \tag{1}$$

where $\mathcal{L}$ is a loss function that evaluates the overall performance of all tasks. The finalized multi-task D^{2}NN will deploy the mapping, $f(x, \theta^{share}, \theta^n) : \mathcal{X} \rightarrow \mathcal{Y}^n$, where θ^{share} are shared parameters in the *shared diffractive layers* between tasks and task-specific parameters θ^n included in *multi-task diffractive layers*. Specifically, in this work, we design and demonstrate the multi-task D^{2}NN with a two-task D^{2}NN architecture shown in Figure 1. Note that the system includes four shared diffractive layers (θ^{share}) and one multi-task diffractive layer for each of the two tasks. The multi-task mapping function becomes $f(x, \theta^{share}, \theta^{1,2}) : \mathcal{X} \rightarrow \mathcal{Y}^2$, and can be then decomposed into:

$$f(x, \theta^{share}, \theta^{1,2}) = det\left(f^1\left(\frac{1}{2} \cdot f^{share}(x, \theta^{share}), \theta^1\right) + f^2\left(\frac{1}{2} \cdot f^{share}(x, \theta^{share}), \theta^2\right)\right) \tag{2}$$

$$f^{share} : \mathcal{X} \rightarrow (\Re + \Im)^{\in 200\times 200}, \quad f^1, f^2 : (\Re + \Im)^{\in 200\times 200} \rightarrow (\Re + \Im)^{\in 200\times 200} \tag{3}$$

where f^{share}, f^1, and f^2 produce mappings in complex number domain that represent light propagation in phase modulated photonics. Specifically, the forward functionality of each diffractive layer and its dimensionality $\mathbb{R}^{200\times 200}$ remains the same as[16]. The output $det \in \mathbb{R}^{C\times 1}$ are the readings from C detectors, where C is the largest number of classes among all tasks; for example, $C = 10$ for MNIST and Fashion-MNIST. The proposed multi-task D^{2}NN system is constructed by designing six phase modulators based on the optimized phase parameters

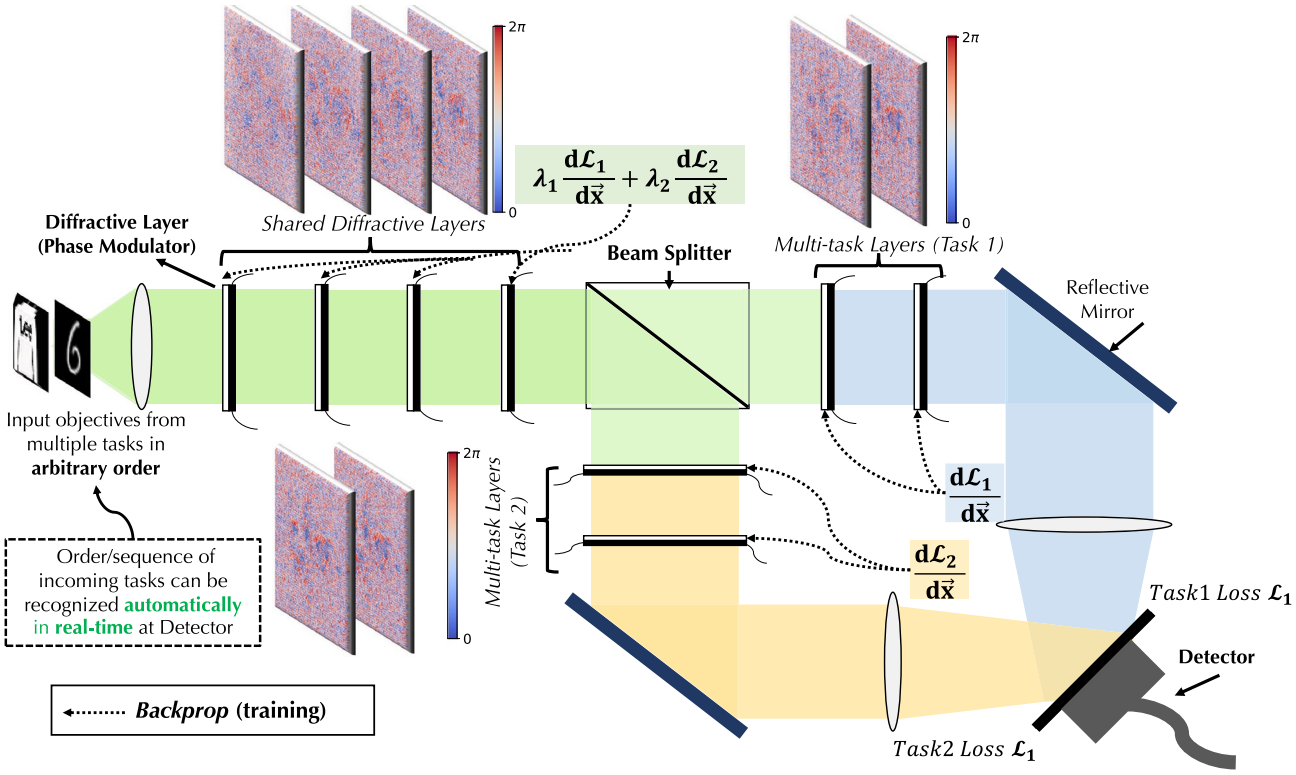

Figure 1. Illustration of multi-task deep learning and multi-task D²NN architecture with two image classification tasks deployed. The proposed multi-task D²NN architecture is formed by four shared diffractive layers and two multi-task layers, where the feed-forward computations have been re-used into multi-task layers using a beam splitter. With a novel training algorithm, the proposed architecture further reduces the hardware complexity that utilizes only ten detectors for both classification tasks, i.e., twenty different classes.

	Single-task system		Multi-task system w 10 det-regions		Multi-task system w 20 det-regions	
	MINST	F-MINST	MINST	F-MINST	MINST	F-MINST
Diffractive layer cost	6 × 200 × 200	6 × 200 × 200	(4 + 2 + 2)×200 × 200		(4 + 2 + 2)×200 × 200	
Detector cost	10	10	10		10 + 10	
Accuracy	0.981	0.889	0.977	0.886	0.979	0.883
Acc-HW Product	1	1	1.99 ×	1.99 ×	~ 1 ×	~ 1 ×

Table 1. Hardware efficiency comparison between single-task and multi-task $\mathbf{D}^2$ NN architectures. For the multi-task D²NN comparison, we compare the hardware efficiency and prediction accuracy between a dual-detection (20 detector regions) architecture and single-detection (10 detector regions). The efficiencies of different D²NN architectures for MNIST and Fashion-MNIST tasks are evaluated using *Accuracy-Hardware product* (a.k.a. Acc-HW), where hardware cost is estimated using the number of detectors.

in the four shared and two multi-task layers (Fig. 1), i.e., $\theta^{share}, \theta^{1,2}$. The phase parameters are optimized with *backpropogation* with gradient chain-rule applied on each phase modulation and adaptive momentum stochastic gradient descent algorithm (`Adam`). The design of phase modulators can be done with 3D printing or lithography to form a passive optical network that performs inference as the input light diffracts from the input plane to the output. Alternatively, such diffractive layer models can also be implemented with spatial light modulators (SLMs), which offers the flexibility of reconfiguring the layers with the cost of limiting the throughput and increase of power consumption.

Table 1 presents the performance evaluation and comparisons of the proposed architecture with other options of classifying both MNIST and Fashion-MNIST tasks. We compare our architecture with—(1) singe-task D²NN architecture, which requires two stand-alone D²NN systems; (2) multi-task D²NN architecture with the same diffractive architecture as Fig. 1 but with two separate detectors for reading and generating the classification results. Specifically, we utilize *Accuracy-Hardware product* (a.k.a. Acc-HW) metric. Regarding the hardware cost, we estimate the cost of the baseline and the proposed systems using the number of detectors. This is because the major cost of the system comes from detectors in practice and the cost of 3D-printed masks is negligible compared to detector cost. To evaluate the hardware efficiency improvements, we set single-task Acc-HW as the baseline, and the improvements of the multi-task D²NN architectures using Eq. (4). We can see that our multi-task D²NN

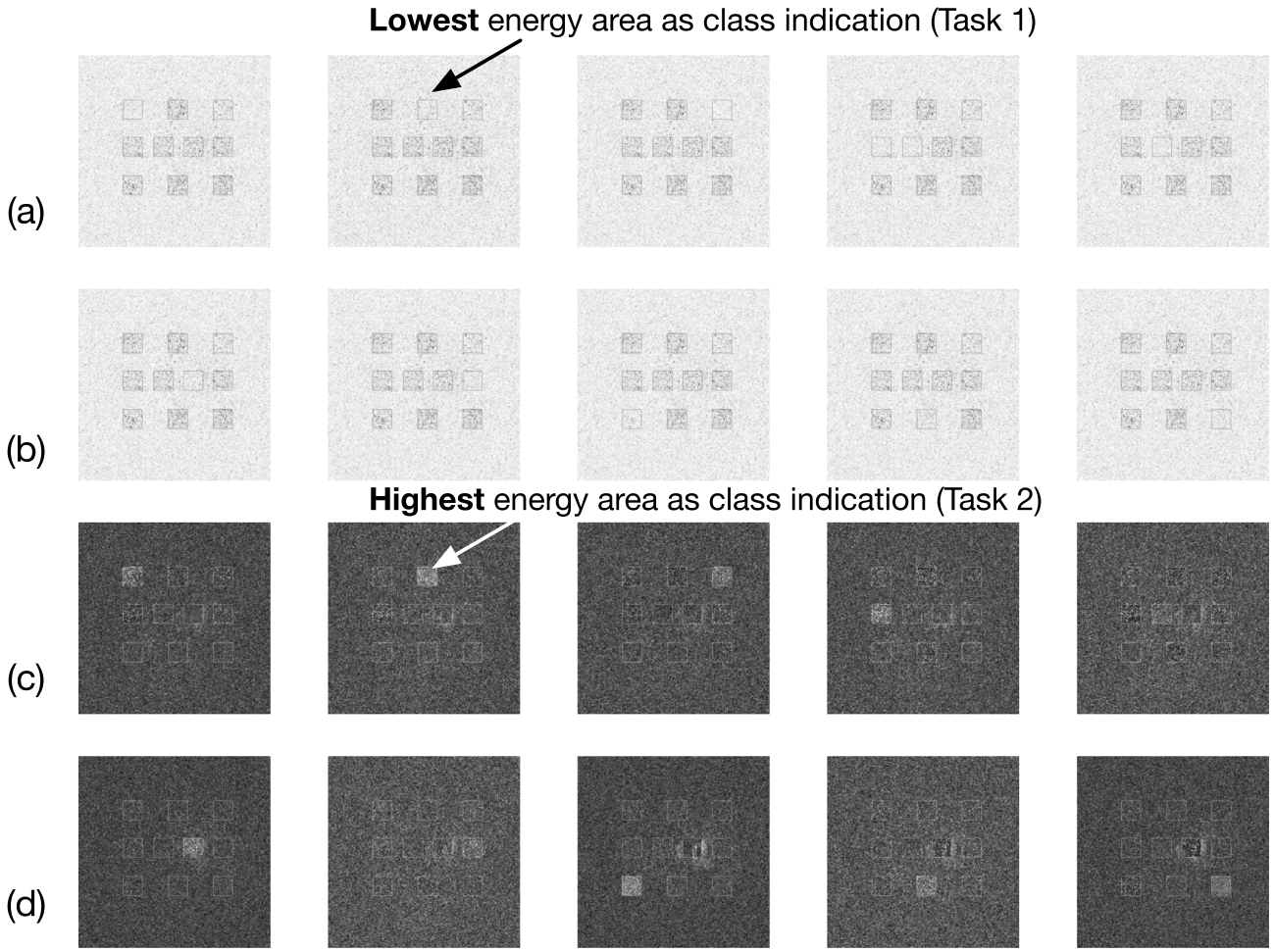

Figure 2. Modeling of ten classes for two different datasets with ten detectors. (**a**, **b**) One-hot encoding for classes 0–9 of the first task (MNIST) represented using the energy value observed at the detectors. Final classes are produced using the index of the lowest energy area, i.e., `argmin`(*det*). (**c**, **d**) One-hot encoding for classes 0–9 represented of the second task (e.g., Fashion-MNIST) using the energy value observed at the detectors. Final classes are produced using the index of the highest energy area, i.e., `argmax`(*det*).

architecture gains 75% efficiency for MNIST task and 72% for Fashion-MNIST task, by introducing a novel multi-task algorithm and modeling that detects 20 different classes (two sets) using only 10 detectors; and gains over 55% and 50% compared to using an architecture that requires two separate sets of detectors.

$$\text{Acc-HW Product} = 1 \cdot \frac{Acc_{multi}}{Acc_{single}} \cdot \frac{HWCost_{multi}}{HWCost_{single}}; HWCost = \#Detectors. \tag{4}$$

Figure 2 illustrates the proposed approach for producing the classes, which re-use the detectors for two different tasks. Specifically, for the multi-task D^{2}NN evaluated in this work, both MNIST and Fashion-MNIST have ten classes. Thus, all the detectors used for one class can be fully re-utilized for the other. To enable an efficient training process, we use one-hot encodings for representing the classes similarly as the conventional multi-class classification ML models. The novel modeling introduced in this work that enables re-using the detectors is—*defining "1" differently in the one-hot representations*. As shown in Fig. 2a,b, for the first task MNIST, the one-hot encoding for classes 0–9 are presented, where each bounding box includes energy values observed at the detectors. In which case, "1" in the one-hot encoding is defined as the lowest energy area, such that the label can be generated as `argmin`(*det*)—the index of the lowest energy area. Similarly, Fig. 2c,d are the one-hot encodings for classes 0–9 of the second task Fashion-MNIST, where label is the index of the highest energy area, i.e., `argmax`(*det*). Therefore, ten detectors can be used to generate the final outputs for two different tasks that share the same number of classes, to gain extra 55% and 50% hardware efficiency of the proposed multi-task D^{2}NN (see Table 1).

Figure 3 includes visualizations of light propagations through multi-task D^{2}NN and the results on the detectors, where the input, internal results after each layer, and output are ordered from left to right. Figure 3a shows one example for classifying MNIST sample, where the output class is correctly predicted (class 7) by returning

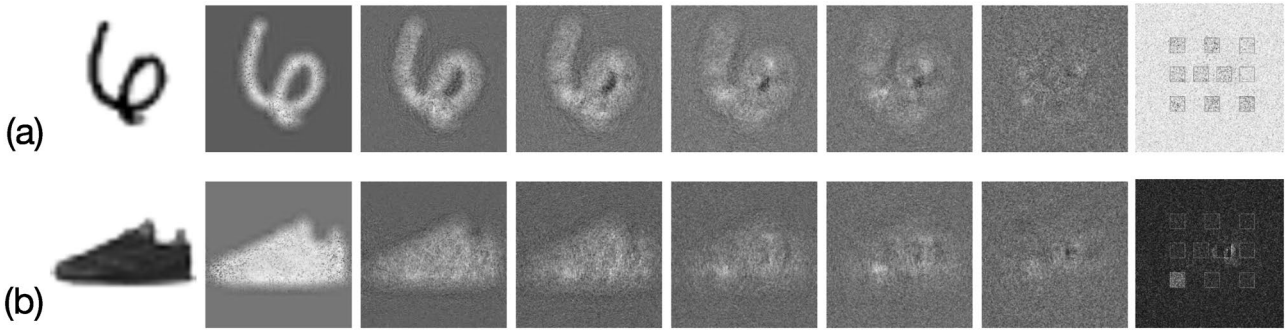

Figure 3. Visualization of propagations through multi-task D^2 NN and the results on the detectors. (**a**) Forward visualization of classifying MNIST10 sample with class = 6, where the 7th detector has the lowest energy value. (**b**) Forward visualization of classifying Fashion-MNIST sample with class = 7, where the 8th detector has the lowest energy value.

the index of the lowest energy detector. Figure 3a presents an example for classifying Fashion-MNIST sample, where the output class is correctly predicted (class 8) by returning the index of the highest energy detector.

While building conventional multi-task DNN, it is well known that the robustness of the multi-task DNNs degrades compared to single-task DNNs, for each individual task. Such concerns become more critical in the proposed multi-task D^2NN system due to the potential system noise introduced by the fabrication variations, device variations, detector noise, etc. Thus, we comprehensively evaluate the noise impacts for our proposed multi-task D^2NN, by considering a wide range of Gaussian noise in detectors and device variations in phase modulators. Details of noise modeling in the proposed systems are discussed in Section Methods (Eqs. 8–10). Figure 4 includes four sets of experimental results for evaluating the robustness of our system under system noise. Specifically, Fig. 4a evaluates the prediction performance of both tasks under detector noise, where the x-axis shows the σ of a Gaussian noise vector S/N (Signal to Noise), and the y-axis shows the accuracy. Figure 4b evaluates the accuracy impacts from device variations of phase modulators, where the x-axis shows the phase variations of each optical neuron in the diffractive layer (note that phase value is $\in [0, 2\pi]$), and the y-axis shows the accuracy. In practice, detector noise is mostly within 5%, and device variations are mostly up to 0.2 (80% yield). We can see that the prediction performance of the proposed system is resilient to a realistic noise range while considering only one type of noise. Moreover, in Fig. 4c,d, we evaluate the noise impacts for MNIST and Fashion-MNIST, respectively, under both detector noise and device variations. While the accuracy degradations are much more noticeable when both noises become significantly, we observe that the overall performance degradations remain $\leq 1\%$ within the practical noise ranges. In summary, the proposed architecture is practically noise resilient.

In multi-task learning, it is often needed to adjust the weight or importance of different prediction tasks according to the application scenarios. For example, one task could be required to have the highest possible prediction performance while the performance of other tasks are secondary. To enable such biased multi-task learning, the shared representations θ^{share} need to carefully adjusted. Figure 5 demonstrates the ability to enable such biased multi-task learning using loss regularization techniques. Specifically, we propose to adjust the performance of different tasks using a novel domain-specific regularization function shown in Eq. (5), where λ_1 and λ_2 are used to adjust the task importance, with a modified *L2 normalization* applied on multi-task layers only. The results with 100 trials of training (with different random seeds for initialization and slightly adjusted learning rate) are included in Fig. 5a. We can see that loss regularization is sufficient to enable biased multi-task learning in the proposed multi-task D^2NN architecture, regardless of the initialization and training setups. Moreover, Fig. 5b empirically demonstrates that with even with very large or small regularization factors, the proposed loss regularization will unlikely overfit either of the tasks because of the adjusted L2 norm used in the loss function (Eq. 6). Note that the adjusted L2 normalization only affects the gradients for θ^1 and θ^2, where λ_{L2} is the weight of this L2 normalization.

$$\mathcal{L}(\theta^{share}, \theta^{1,2}) = \underbrace{\lambda_1}_{\text{t1 factor}} \mathcal{L}_1(\theta^{share}, \theta^1) + \underbrace{\lambda_2}_{\text{t2 factor}} \cdot \mathcal{L}_2(\theta^{share}, \theta^2) + \underbrace{\lambda_{L2} \frac{\lambda_2}{\lambda_1} \cdot ((\theta^1)^2 + (\theta^2)^2)}_{\text{adjusted L2 norm}} \tag{5}$$

Methods

Multi-task D^2 NN architecture. Figure 1 shows the design of the multi-task D^2NN architecture. Based on the phase parameters θ^{share}, θ^1, and θ^2, there several options to implement the diffractive layers to build the multi-task D^2D^2NN system. For example, the passive diffractive layers can be manufactured using 3D printing for long-wavelength light (e.g. terahertz) or lithography for short-wavelength light (e.g. near-infrared), and active reconfigurable ones can be implemented using spatial light modulators. A 50–50 *beam splitter* is used to split the output beam from the last shared diffractive layer into two ideally identical channels for multi-task layers. Coherent light source, such as laser diodes, is use in this system. At the output of two multi-task layers, the electromagnetic vector fields are added together on the detector plane. The generated photocurrent corresponding to the optical intensity of summed vector fields is measured and observed as output labels. Regarding the

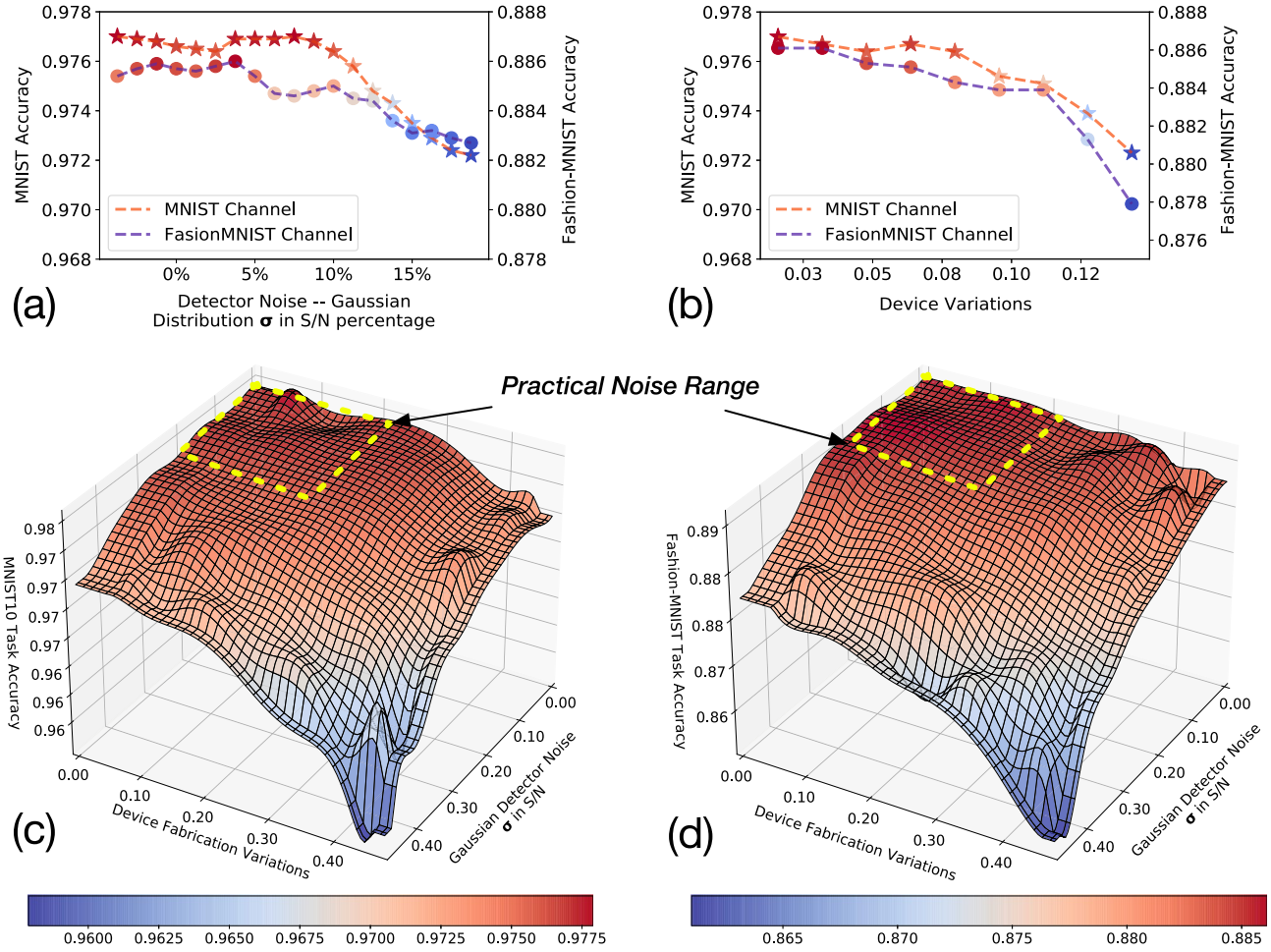

Figure 4. Evaluations of robustness against system noise of the proposed multi-task D^2NN, by considering a wide range of Gaussian noise in detectors and device variations in phase modulators. Details of noise modeling in the proposed systems are discussed in Section Methods (Eqs. 8–10). (**a**) Prediction performance evaluation under Gaussian detector noise with σ shown in S/N (Signal to Noise) $\in [0, 0.2]$. (**b**) Prediction performance evaluation under Gaussian device variations. (**c**) Evaluations of MNIST task accuracy under combined detector noise and device variations. (**d**) Evaluations of Fashion-MNIST task accuracy under combined detector noise and device variations.

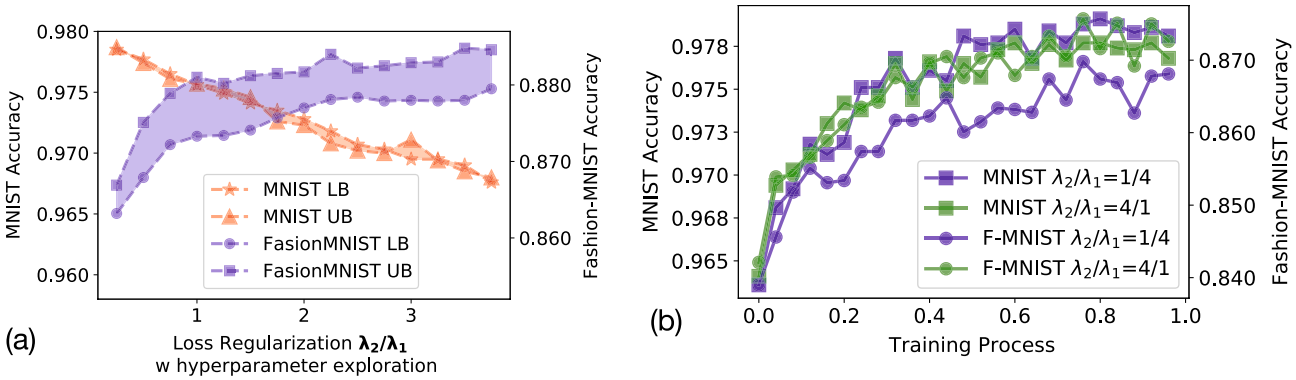

Figure 5. Evaluation of loss regularization for adjusting the performance of each task. (**a**) Testing accuracy with different regularization factors. As $\frac{\lambda_2}{\lambda_1}$ increases (decreases), the final performance of the multi-task D^{2}NN will be bias to Fashion-MNIST (MNIST). We include results of 100 different hyperparameters for training. (**b**) Testing accuracy of both tasks during the training phase, where we can see that even the largest and smallest regularization factors do not cause overfitting.

real-time capability of the proposed system, the proposed architecture performs the same the system proposed in[16], where computation is executed at the speed of light and the information is processed on each neuron/pixel of the phase mask is highly parallel. Thus, the time of light flight is negligible and the determination factor for system hardware performance is dependent on the performance of THz detectors. For a detector with operation bandwidth f, the corresponding latency is $1/f$ and the largest throughput is f frames/s/task. The minimum power requirement for this system is determined by the number of detector, *NEP* (noise-equivalent-power), and , if we assume the loss and energy consumption associated with phase masks is negligible. In practice, considering a room-temperature VDI detector (https://www.vadiodes.com/en/products/detectors?id=214) operating at $\sim 0.3\ THz$, $f =\sim 40\ GHz$, and $NEP = 2.1pW/\sqrt[2]{Hz}$, the latency of the system will be 25 ps, throughput is $4 \times 10^{10}\ fps/task$ (frame/second/task), with power consumption 0.42 uW. In addition to mitigate the large cost of detectors, alternative materials can be used, such as graphene. For example, the specific detector performance shown in[28] is $NEP =\sim 80pW/\sqrt[2]{Hz}$, and $f =\sim 300\ MHz$. In which case, the system atency is $\sim 30ns$, such that the throughput is $3 \times 10^8\ fps/task$ with the estimated minimum power 1.4 uW.

Training and inference of multi-task D² NN. The proposed system has been implemented and evaluated using Python (v3.7.6) and Pytorch (v1.6.0). The basic components in the multi-task D²NN PyTorch implementation includes (1) diffractive layer initialization and forward function, (2) beam splitter forward function, (3) detector reading, and (4) final predicted class calculation. First, each layer is composed of one diffractive layer that performs the same phase modulation as[16]. To enable high-performance training and inference on GPU core, we utilize for complex-to-complex Discrete Fourier Transform in PyTorch (`torch.fft`) and its inversion (`torch.ifft`) to mathematically model the same modulation process as[16]. Beam splitter that evenly splits the light into *transmitted light* and *reflected light* is modeled as dividing the complex tensor produced by the shared layers in half. The trainable parameters are the phase parameters in the diffractive layers that modulate the incoming light. While all the forward function components are differentiable, the phase parameters can be simply optimized using automatic differentiation gradient mechanism (autograd). The detector has ten regions and each detector returns the sum of all the pixels observed (Fig. 2). To enable training with two different one-hot representations that allow the system to reuse ten detectors for twenty classes, the loss function is constructed as follows:

$$\begin{aligned}\mathcal{L} = \lambda_1 \cdot \underbrace{\texttt{MSELoss}(LogSoftmax(f(\theta^{share}, \theta^1, \mathcal{X}^1), (label^1 + 1)\%2)}_{\text{one-hot encoding with one“0” and nine “1s”}} \\ + \lambda_2 \cdot \underbrace{\texttt{MSELoss}(LogSoftmax(f(\theta^{share}, \theta^2, \mathcal{X}^2), label^2)}_{\text{one-hot encoding with one “1” and nine “0s”}} \\ + \underbrace{\lambda_{L2} \cdot \frac{\lambda_2}{\lambda_2}\texttt{L2}(\theta^1, \theta^2)}_{\text{L2 norm on multi-task diffractive layers}}\end{aligned} \tag{6}$$

The original labels $label^1$ and $label^2$ are represented in conventional one-hot encoding, i.e., one “1” with nine of “0s”, and $label^1$ has been converted into an one-hot encoding with one “0” and nine “1s”. Note that LogSoftmax function is only used for training the network, and the final predicted classes of the system are produced based on the values obtained at the detectors. With loss function shown in Eq. (6) and the modified one-hot labeling for task 1, the training process optimizes the model to (1) given an input image in class c for task 1 (MNIST), minimize the value observed at (c+1)th detector, as well as maximize the values observed at other detectors; (2) given an input image in class c for task 1 (Fashion-MNIST), maximize the value observed at (c+1)th detector, as well as minimize the values observed at other detectors. Thus, the resulting multi-task model is able to automatically differentiate which task the input image belongs to based on the sum of values observed in the ten detectors, and then generate the predicted class using `argmin` (`argmax`) function for MNIST (Fashion-MNIST) task. The gradient updates have been summerized in Eq. (7).

$$\begin{aligned}\theta^{share} &= \theta^{share} - \frac{1}{2} \cdot \eta \frac{\lambda_2}{\lambda_1}(\nabla\theta^1 + \nabla\theta^2) \\ \theta^{1'} &= \theta^1 - \eta\nabla\theta^1 - 2\eta\lambda_{L2}||\theta^1 + \theta^2|| \\ \theta^{2'} &= \theta^2 - \eta\nabla\theta^2 - 2\eta\lambda_{L2}||\theta^1 + \theta^2||\end{aligned} \tag{7}$$

System noise modeling. We demonstrate that the proposed system is robust under the noise impacts from the device variations of diffractive layers and the detector noise in our system. Specifically, to include the noise attached to the detector, we generate a Gaussian noise mask $\mathcal{N}(\sigma, \mu) \in \mathbb{R}^{200\times200}$ with on the top of the detector readings, i.e., each pixel observed at the detector will include a random Gaussian noise. As shown in Fig. 4a, we evaluate our system under multiple Gaussian noises defined with different σ with $\mu = 0$. We also evaluated the impacts of μ, while we do not observe any noticeable effects on the accuracy for both tasks. This is because increasing μ of a Gaussian noise tensor does not change the ranking of the values observed by the ten detectors, such that it has no effect on the finalized classes generated with `argmax` or `argmin`. The forward function for ith task with detector noise is shown in Eq. (8).

$$c^i = \texttt{argmax/argmin}(det(f(\theta^{share}, \theta^i, \mathcal{X}^i)) + \mathcal{N}(\sigma, 0)), \quad i = \{1, 2\} \tag{8}$$

We also considered the imperfection of the devices used in the system. With 3D printing or lithography based techniques, the imperfection devices might not implement exactly the phase parameters optimized by the training process. Specifically, we consider the imperfection of the devices that affect the phases randomly under a Gaussian noise. As shown in Fig. 4b, the x-axis shows that the σ of Gaussian noise that are added to the phase parameters for inference testing. The forward function is described in Eq. (9). Beam splitter noise has also been quantified, where we do not see direct impacts on both tasks (see Fig. 2 in supplementary file SI.pdf).

$$\begin{aligned} \theta^{share}_{\mathcal{N}} &= (\theta^{share} + \mathcal{N}(\sigma, 0)) \quad \% \quad 2\pi \\ \theta^{i}_{\mathcal{N}} &= (\theta^{i} + \mathcal{N}(\sigma, 0)) \quad \% \quad 2\pi, \quad i = \{1, 2\} \\ c^i &= \texttt{argmax/argmin}(det(f(\theta^{share}_{\mathcal{N}}, \theta^i_{\mathcal{N}}, \mathcal{X}^i_{\mathcal{N}}))), \quad i = \{1, 2\} \end{aligned} \tag{9}$$

Finally, for results shown in Fig. 4c,d, we include both detector noise and device variations in our forward function (Eq. 10):

$$\begin{aligned} \theta^{share}_{\mathcal{N}} &= (\theta^{share} + \mathcal{N}^1(\sigma^1, 0)) \quad \% \quad 2\pi \\ \theta^{i}_{\mathcal{N}} &= (\theta^{i} + \mathcal{N}^1(\sigma^1, 0)) \quad \% \quad 2\pi, \quad i = \{1, 2\} \\ c^i &= \texttt{argmax/argmin}(det(f(\theta^{share}_{\mathcal{N}}, \theta^i_{\mathcal{N}}, \mathcal{X}^i_{\mathcal{N}})) + \mathcal{N}^2(\sigma^2, 0))), \quad i = \{1, 2\} \end{aligned} \tag{10}$$

Received: 4 February 2021; Accepted: 4 May 2021
Published online: 26 May 2021

References

1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
2. Senior, A. W. *et al.* Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).
3. Silver, D. *et al.* Mastering the game of go without human knowledge. *Nature* **550**, 354–359 (2017).
4. Jouppi, N.P. *et al.* In-datacenter performance analysis of a tensor processing unit. *Int'l Symp. on Computer Architecture (ISCA)*, 1–12 (2017).
5. Schlottmann, C. R. & Hasler, P. E. A highly dense, low power, programmable analog vector-matrix multiplier: The fpaa implementation. *IEEE J. Emerg. Select. Top. Circ. Syst.* **1**, 403–411 (2011).
6. Bankman, D., Yang, L., Moons, B., Verhelst, M. & Murmann, B. An always-on 3.8μ j /86% cifar-10 mixed-signal binary cnn processor with all memory on chip in 28-nm cmos. *IEEE J. Solid State Circ.* **54**, 158–172 (2018).
7. LiKamWa, R., Hou, Y., Gao, J., Polansky, M. & Zhong, L. Redeye: Analog convnet image sensor architecture for continuous mobile vision. *ACM SIGARCH Comput. Arch. News* **44**, 255–266 (2016).
8. Wang, Z. *et al.* Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electron.* **1**, 137–145 (2018).
9. Boybat, I. *et al.* Neuromorphic computing with multi-memristive synapses. *Nat. Commun.* **9**, 1–12 (2018).
10. Hu, M. *et al.* Memristor-based analog computation and neural network classification with a dot product engine. *Adv. Mater.* **30**, 1705914 (2018).
11. Jiang, Y. *et al.* Design and hardware implementation of neuromorphic systems with rram synapses and threshold-controlled neurons for pattern recognition. *IEEE Trans. Circ. Syst. I Regul. Pap.* **65**, 2726–2738 (2018).
12. Wang, Z. *et al.* Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. *Nat. Mater.* **16**, 101–108 (2017).
13. Zand, R., & DeMara, R.F. Snra: A spintronic neuromorphic reconfigurable array for in-circuit training and evaluation of deep belief networks. in *2018 IEEE International Conference on Rebooting Computing (ICRC)* (IEEE, 1–9, 2018).
14. Feldmann, J., Youngblood, N., Wright, C. D., Bhaskaran, H. & Pernice, W. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature* **569**, 208–214 (2019).
15. Hamerly, R., Bernstein, L., Sludds, A., Soljačić, M. & Englund, D. Large-scale optical neural networks based on photoelectric multiplication. *Phys. Rev. X* **9**, 021032 (2019).
16. Lin, X. *et al.* All-optical machine learning using diffractive deep neural networks. *Science* **361**, 1004–1008 (2018).
17. Luo, Y. *et al.* Design of task-specific optical systems using broadband diffractive neural networks. *Light Sci. Appl.* **8**, 1–14 (2019).
18. Mengu, D., Luo, Y., Rivenson, Y. & Ozcan, A. Analysis of diffractive optical neural networks and their integration with electronic neural networks. *IEEE J. Select. Top. Quant. Electron.* **26**, 1–14 (2019).
19. Mengu, D., Rivenson, Y., & Ozcan, A. Scale-, shift-and rotation-invariant diffractive optical networks. http://arxiv.org/abs/2010.12747 (2020).
20. Rahman, M.S.S., Li, J., Mengu, D., Rivenson, Y., & Ozcan, A. Ensemble learning of diffractive optical networks. http://arxiv.org/abs/2009.06869 (2020).
21. Shen, Y. *et al.* Deep learning with coherent nanophotonic circuits. *Nat. Photon.* **11**, 441 (2017).
22. Silva, A. *et al.* Performing mathematical operations with metamaterials. *Science* **343**, 160–163 (2014).
23. Tait, A. N. *et al.* Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.* **7**, 1–10 (2017)
24. Paszke, A. *et al.* Automatic differentiation in pytorch. in *Advances in neural information processing systems* (NeurIPS'17, (2017).
25. Feldmann, J. *et al.* Parallel convolution processing using an integrated photonic tensor core. *Nature* (2020).
26. Li, J., Mengu, D., Luo, Y., Rivenson, Y., & Ozcan, A. Class-specific differential detection improves the inference accuracy of diffractive optical neural networks. In *Emerging Topics in Artificial Intelligence 2020*, vol. 11469, 114691A (International Society for Optics and Photonics, 2020).
27. Ruder, S. An overview of multi-task learning in deep neural networks. http://arxiv.org/abs/1706.05098 (2017).
28. Castilla, S. *et al.* Fast and sensitive terahertz detection using an antenna-integrated graphene pn junction. *Nano Lett.* **19**, 2765–2773 (2019).

Acknowledgements
C.Y. thanks the support from grants NSF-2019336 and NSF-2008144. C.Y. and W.G. thank the support from the University of Utah start-up fund. B.S.R thanks the support from grants NSF-1936729.

Author contributions
C.Y., B.S.R., W.G. contributed to the overall idea of this paper. C.Y. and W.G. wrote the main manuscript text. Y.L. and C.Y. prepared Figs. 1, 2, 3, 4 and 5 and R.C. and W.G. prepared Supplementary Fig. 1. All authors reviewed the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-90221-7.

Correspondence and requests for materials should be addressed to W.G. or C.Y.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.