DIFFERENTIABLE DISCRETE DEVICE-TO-SYSTEM CODESIGN FOR OPTICAL NEURAL NETWORKS VIA GUMBLE-SOFTMAX

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks (DNNs) have significantly improved the productions in many areas like large-scale computer vision and natural language processing. While conventional DNNs implemented on digital platforms have intrinsic limitations in computation and memory requirements, optical neural networks (ONNs), such as diffractive optical neural networks (DONNs), have attracted lots of attention as they can bring significant advantages in terms of power efficiency, parallelism, and computational speed. In order to train DONNs, fully differentiable physical optical propagations have been developed, which can be used to train the physical parameters in optical systems using conventional gradient descent algorithms. However, inversely mapping algorithm-trained physical model parameters onto the applied stimulus in real-world optical devices is a non-trivial task, which can involve multiple imperfections (e.g., quantization and non-monotonicity) and is especially challenging in complex-valued domains. This work proposes a novel device-to-system hardware-software codesign framework, which enables efficient training of DONNs w.r.t arbitrary experimental measured optical devices across layers. Specifically, *Gumbel-Softmax* with a novel complex-domain regularization method is employed to enable differentiable one-to-one mapping from discrete device parameters into the forward function of DONNs, where the physical parameters in DONNs can be trained by simply minimizing the loss function of the ML task. The experimental results have demonstrated significant advantages over traditional quantization-based methods with low-precision optical devices (e.g., 8 discrete values), with $\sim$20% accuracy improvements for MNIST and $\sim$28% for FashionMNIST. More importantly, our framework provides high versatility in codesign even for one system implemented with mixed optical devices. In addition, we include comprehensive studies of regularization analysis, temperature scheduling exploration, and runtime complexity evaluation of the proposed framework.

1 INTRODUCTION

During the past half-decade, there has been significant growth in machine learning with deep neural networks (DNNs). DNNs improve productivity in many domains such as large-scale computer vision, natural language processing, and data mining tasks (LeCun et al. (2015); Silver et al. (2017); Senior et al. (2020)). However, conventional DNNs implemented on digital platforms have intrinsic limitations in computation and memory requirements (Jouppi et al. (2017); Sharma et al. (2016); Abadi et al. (2016)). When it deals with computation-intense tasks, its energy cost will be a great concern. To overcome limitations in resources and find an energy-save computation method, people have turned their eyes to optics. Specifically, the free-space diffractive optical neural networks (DONNs), which is based on light diffraction, featuring millions of neurons in each layer interconnected with neurons in neighboring layers, show its great potential in improving efficiency in computing with neural networks Lin et al. (2018). More importantly, Lin et al. (2018); Rahman et al. (2020); Li et al. (2021) demonstrated that diffractive models controlled by physical parameters are differentiable, such that the parameters can be optimized with conventional automatic differentiation engines.

However, when such DONNs system is deployed on physical hardware, it shows significant accuracy degradation compared to the numerical physics emulation. To narrow the algorithm-hardware

miscorreclation gaps between differentiable numerical physics models and physical optical systems, hardware-software codesign training algorithms are needed to deal with the practical response of optical devices, i.e., the feasible and accurate deployment of trained parameters onto devices. For example, the reconfigurability of DONNs is implemented using spatial light modulators (SLMs), which generally have a discrete and non-monotonic complex-valued modulation of propagating optical fields as a function of applied voltages with finite-precision. Therefore, despite the diffractive propagation in the DONNs system is differentiable, directly adding discrete mapping from device to DONNs system will break the gradient chain in backpropagation. More importantly, diffractive layers can behave differently due to different optical configuration or device responses. Thus, while training a multi-layer DONNs system, there is a great need to develop a flexible training framework that can optimize the DONNs parameters w.r.t various optical devices from layer to layer.

This work proposes a novel, efficient, and flexible framework that enables differentiable discrete mappings from devices to DONNs systems via Gumbel-Softmax. Moreover, we introduce a complex-domain regularization technique, which significantly improves the training performance. The proposed framework overcomes the physics-aware training limitations of existing training methodologies. Our experimental results demonstrate the advantages over existing state-of-the-art codesign training algorithms, particularly for devices with few discrete states (≤ 16), in various DONNs architectures settings. With the regularization technique, our framework is to overcome the training limitations on complex-valued backpropagation in DONNs training.

2 BACKGROUND

Diffractive Optical Neural Networks (DONNs) Recently, there have been increasing efforts on optical neural networks and optics-based DNNs hardware accelerators, which bring significant advantages for machine learning systems in terms of their power efficiency, parallelism and computational speed, demonstrated at various optical computing systems by Gao et al. (2021b;a); Mengu et al. (2020); Lin et al. (2018); Feldmann et al. (2019); Shen et al. (2017); Tait et al. (2017); Rahman et al. (2020); Li et al. (2021). Among them, free-space *diffractive optical neural networks* (DONNs) , which is based on the light diffraction, features 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. One of the significant advantages of DONNs is the computational density, where such a platform can be scaled up to millions of artificial neurons. In contrast, the design complexity for deploying deep learning algorithms on other optical architectures, e.g., silicon photonic platform proposed by Feldmann et al. (2019; 2020)Tait et al. (2017)), can dramatically increase. For example, Lin et al. (2018); Li et al. (2021); Mengu et al. (2019; 2020) experimentally demonstrated various complex functions with an all-optical DONNs. In conventional DNNs, forward propagations are computed by generating the feature representation with floating-point weights associated with each neural layer. While in DONNs, such floating-point weights are encoded in the complex-valued transmission coefficient of each neuron in diffractive layers and free-space propagation function, which is multiplied onto the light wavefunction as it propagates through the neuron to next diffractive layer. 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 energy reading over the output signals of the last layer observed by detectors. Specific examples of the system at training and inference can be found in the next section (Figure 1a).

However, there are several critical limitations in the existing DONNs training methodology. First, the prediction accuracy of DONNs with few diffractive layers (≤ 3) (Lin et al. (2018)) is highly limited, even with simple image classification datasets. While Lin et al. (2018) claimed that it is caused by fundamental limitations of optical physics, it is believed that there are potentials on improving the complex-domain training algorithms. Moreover, when such DONNs systems are deployed on physical hardware, existing training approaches do not take real device response into consideration and only assume simple phase-only modulation without any limitations. This ideal assumption creates miscorrelation gaps between numerical models and hardware deployment, leading to significant accuracy degradation such as 30% drop on MNIST dataset demonstrated by Zhou et al. (2021). The framework proposed in this work aims to overcome all these limitations with Gumbel-Softmax enabling discrete mapping training and a novel complex-domain regularization technique.

Gumbel-Softmax Gumbel-Softmax is a continuous distribution on the simplex which can be used to approximate discrete samples (Maddison et al. (2016); Jang et al. (2016); Gumbel (1954)). With Gumbel-Softmax, discrete samples can be differentiable and their parameter gradients can be easily computed with standard backpropagation. Let z be the discrete sample with one-hot representation with k dimensions and its class probabilities are defined as $\pi_1, \pi_2, ..., \pi_k$. Then, according to the Gumbel-Max trick proposed by Gumbel (1954), the discrete sample z can be presented by:

$$z = \texttt{one\_hot}(\arg\max_i [g_i + log\pi_i]) \tag{1}$$

where g_i are i.i.d samples drawn from Gumbel(0, 1). Then, we can use the differentiable approximation `Softmax` to approximate the one-hot representation for z, i.e., $\nabla_\pi z \approx \nabla_\pi y$:

$$y_i = \frac{exp((log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} exp((log(\pi_i) + g_i)/\tau)} \tag{2}$$

where $i = 1, 2, ..., k$. The softmax temperature τ is introduced to modify the distributions over discrete states. Softmax distributions will become more discrete and identical to one-hot encoded discrete distribution as $\tau \to 0$, while at higher temperatures, the distribution becomes more uniform as $\tau \to \infty$ (Jang et al. (2016)). Gumbel-Softmax distributions have a well-defined gradient $\frac{\partial y}{\partial \pi}$ w.r.t the class probability π. When we replace discrete states with Gumbel-Softmax distribution depending on its class probability, we are able to use backpropagation to compute gradients.

3 APPROACH

This section describes the proposed training framework integrated with Gumbel-Softmax and complex-domain regularization, which emulates optical diffraction in the complex domain and conducts physic-aware training with real-world optical devices.

System Overview We illustrate the proposed training framework using five-layer DONNs system implemented for a ten-class image classification task, with ten detector regions placed evenly on the detector plane. The optical devices used for deploying phase and amplitude modulation are controlled by the input discrete voltage values. As shown in Figure 1, each diffractive layer modifies the amplitude and phase of the input light signal. To enable differentiable discrete mapping, our framework defines the input discrete voltage values as trainable parameters, where each pixel is represented using a one-hot vector. The trainable parameters dimensions is then defined by (1) the system size and (2) the number of discrete values in the devices. For example, let the system size be 200×200 with 8 discrete states in the devices, the trainable voltage parameters will be $200 \times 200 \times 8$ in each layer. Note that the phase and amplitude modulation will still be in the shape of 200×200, where the optical properties are mapped by `matmul` the one-hot vectors and the device state vector.

To deal with the problem of gradient chain breakage brought by the discrete trainable parameters, *Gumbel-Softmax* is added in the numerical modeling of DONNs. As it is shown in Figure 1, during backpropagation in the training process, instead of propagating gradients to the discrete one-hot voltage levels directly, it will propagate through the differentiable approximation to the discrete levels generated by Gumbel-Softmax distribution with its class probability θ. The differentiable approximation will be updated according to the training algorithm, and the discrete voltage levels will be updated by its class probability θ from approximation. Let A be the array with discrete calibrated amplitude value, P be the array with discrete calibrated phase value and $w^{i,j}$ be the voltage level applied to the pixel located at $[i, j]$ in the diffractive layer with a size $N \times N$.

$$\begin{aligned} w^{i,j} &= \texttt{one\_hot}(\frac{exp((log(\theta^{i,j}) + g^{i,j})/\tau)}{\sum_{j=1}^{k} exp((log(\theta^{i,j}) + g^{i,j})/\tau)}), g^{i,j} \sim \text{Gumbel}(0,1), k = \text{discrete size} \\ w_{\mathbb{C}}^{i,j} &= \overbrace{\underbrace{(w^{i,j} \cdot A)}_{\text{Matmul}} \times \cos(w^{i,j} \cdot P)}^{\text{Real}} + \underbrace{i(w^{i,j} \cdot A) \times \sin(w^{i,j} \cdot P)}_{\text{Imaginary}}, i, j \in [0, N-1] \end{aligned} \tag{3}$$

Let $w^{i,j} \in W$, where $i, j \in [0, N-1]$, N is the size of diffractive layer. According to Equation 3, the discrete variable W has the distribution depending on θ and cost function $f(W)$. The objective is

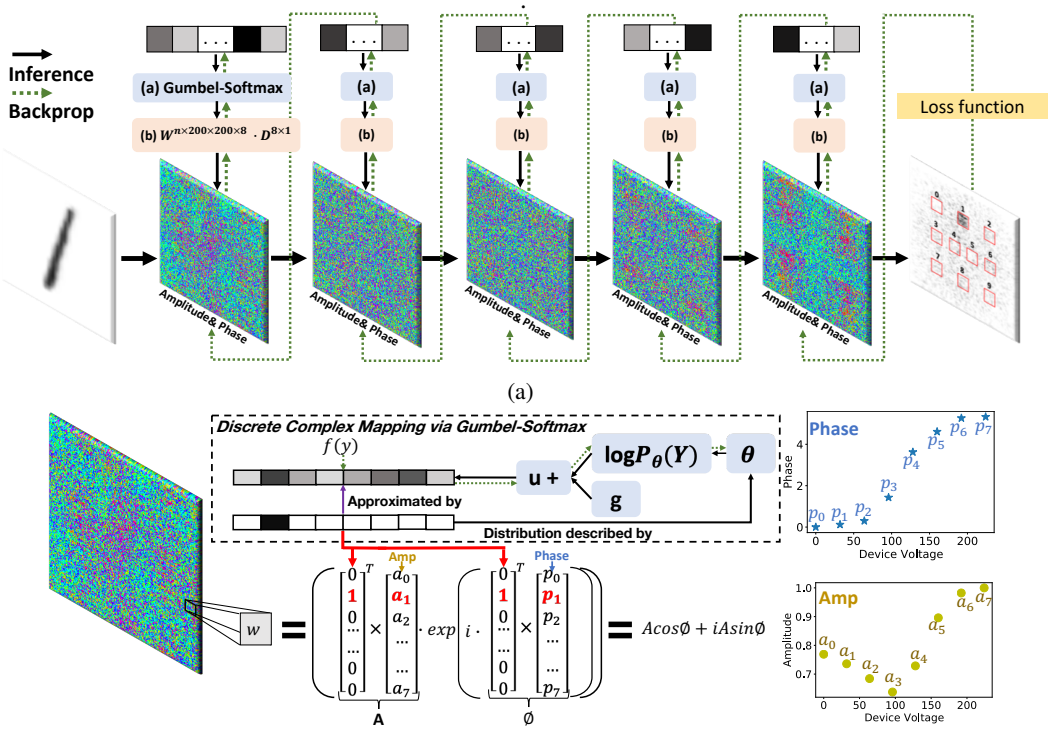


Figure 1: Illustration of the proposed framework in training a five-layer DONNs system with Gumbel-Softmax and complex-domain regularization – (a) In the inference path, input signal will be modulated by each layer in both amplitude and phase encoded by optical devices controlled by discrete voltage values. In backpropagation path in training, starting from loss function, the gradients will flow through Gumbel-Softmax over the approximated differentiable representation of the one-hot encoded discrete voltage levels; (b) Detailed illustration of discrete device-to-system mapping via Gumbel-Softmax in both forward and backpropagation.

to minimize the expected cost $L(\theta) = \mathbb{E}_{W\sim p_\theta(W)}[f(W)]$, which is the ML loss in the system, e.g., in DONNs system for image classification, L is usually set as the MSE Loss (Lin et al. (2018), Zhou et al. (2021)), via gradient descent, which requires us to estimate $\nabla_\theta \mathbb{E}_{W\sim p_\theta(W)}[f(W)]$. The discrete sample W can be approximated by $G(\theta, g)$. The gradients from f to θ will be computed as follows:

$$\frac{\partial}{\partial\theta}\mathbb{E}_{W\sim p_\theta(W)}[f(W)] = \frac{\partial}{\partial\theta}\mathbb{E}_g[f(G(\theta,g))] = \mathbb{E}_{g\sim\text{Gumbel}(0,1)}[\frac{\partial f}{\partial G}\frac{\partial G}{\partial\theta}] \tag{4}$$

To be more specific, as shown in Figure 1b, for each pixel ($w^{i,j}$) in diffractive layers, the modulation provided by the pixel is represented by complex tensors that emulate the diffraction of light. The complex number is transformed from phase and amplitude modulation by *Euler's formula*. Since the input light signal is also described by complex numbers, the modulation can be easily realized by multiply the two complex numbers (see Equation 3). In Gumbel-Softmax, to approximate the discrete levels, a Gumbel distribution $g \sim Gumbel(0, 1)$ and a class probability θ for the discrete states will be introduced. With the help of these two differentiable elements, the approximation for discrete states will be differentiable. As a result, the device-to-system codesign with consideration of discrete device information will be differentiable and can be trained with conventional *autograd* optimization algorithms simply minimizing DONNs' ML training loss function. Note that in the DONNs system, since diffractive layers are propagated in sequence, each layer can be implemented using different devices, i.e., with different calibrated data points. As we can see in Figure 1, mapping multiple devices in one DONNs system can be simply realized using the proposed framework by replacing the Amp and $Phase$ vectors.

Gumbel-Softmax Exploration As it is discussed in Section 2, the variance of the approximated Gumbel-Softmax distribution over discrete states is determined by τ, which is also referred as

temperature in Gumbel-Softmax. While giving a higher temperature value, Gumbel-Softmax will result in less variance that is close to a uniform distribution. In opposite, when τ is close to 0, it will be more variant over discrete states, i.e., be more identical to one-hot distribution. Thus, similar to simulated annealing, the algorithm should be first deployed with high temperature to enable coarse-grain global search. The temperature should then be annealed down to shrink the search space to find the local optimized point. Specifically, at the early training stage, we expect the variance between different states to be small, such that the discrete values are easier to be changed during gradient descent optimization. As the optimization efforts increase, it is expected to decrease the temperature to fine-tune the optimization, where most of the discrete values are far more stable during gradient descent optimization. In Section 4.1, we explore and provide comprehensive discussions on six different temperature schedules.

Complex-domain Regularization As shown in Figure 1, the trainable discrete values control both optical amplitude and phase, i.e., the gradients backpropagated to Gumbel-Softmax are the average of the upstream gradients for amplitude and phase (complex-domain mapping). In addition, according to Lin et al. (2018), DONNs is more phase modulation dominated. Thus, we introduce a novel regularization factor λ in the forward function described in Equation 4 to improve the training efficiency, which can flexibly change the gradient scales between amplitude and phase modulations. Specifically, λ is applied to amplitude vector A. Let $\mathcal{M}$ as the mapping of $(w^{i,j}, A, P) \xrightarrow{\mathcal{M}} w_{\mathbb{C}}^{i,j}$. Let $\nabla w_{\mathbb{C}}^{i,j}$ be the upstream gradient to calculate $\nabla w^{i,j}$. The difference of the gradient updates can be summarized as follows:

$$\nabla w^{i,j} = \nabla w_{\mathbb{C}}^{i,j} \cdot \frac{\frac{\partial \mathcal{M}}{\partial w^{i,j}} \cdot A + \frac{\partial \mathcal{M}}{\partial w^{i,j}} \cdot P}{2} \quad \xrightarrow{\lambda} \quad \nabla w_{reg}^{i,j} = \nabla w_{\mathbb{C}}^{i,j} \cdot \frac{\frac{1}{\lambda} \cdot (\frac{\partial \mathcal{M}}{\partial w^{i,j}} \cdot A) + \frac{\partial \mathcal{M}}{\partial w_{i,j}} \cdot P}{2} \tag{5}$$

Since in DONNs system, the intensity of the input light decreases exponentially as diffractive layers increase, the weight of amplitude modulation in forward and backward propagation are expected to decrease exponentially, as the number of layers increases. Thus, the optimal value for the algorithmic parameter λ needs to be explored for different DONNs architectures. Empirically, λ should be $\geq$1, i.e., optimizing $w^{i,j}$ with more bias on phase gradients, for any number of layers in DONNs. In Section 4.3, we explore the performance of the proposed regularization technique with 1, 3 and 5 diffractive layers in DONNs, which confirms the discussed regularization characteristic.

4 RESULTS

System Setup The default system used in this work is designed with five diffractive layers with the size of 200×200, i.e., the size of layers and the size of total ten detector regions are 200×200. To fit the optical system, the original input images from MNIST (LeCun (1998)) and FashionMNIST (FMNIST) (Xiao et al. (2017)) with size of 28×28 will be interpolated into size of 200×200 and encoded with the laser source whose wavelength is $532\ nm$. The physical distances between layers, first layer to source, and final layer to detector, are set to be $30\ cm$. As shown in Figure 1, ten separate detector regions for ten classes are placed evenly on the detector plane with the size of 20×20, where the sums of the intensity of these ten regions are equivalent to a 1×10 vector in `float32` type. The final prediction results will be generated using `argmax`. The default DONNs system is implemented with an optical device consisting of 8 discrete voltage values for modulation.

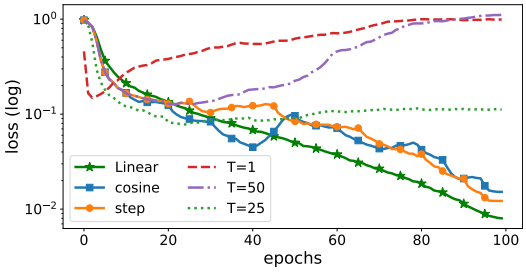
Training Setups The learning rate in the training process is 0.5 trained with 100 epochs for all experiments using `Adam` (Kingma & Ba (2014)) with batch size 500. The implementations are constructed using PyTorch v1.8.1. All experimental results are conducted on an Nvidia 2080 Ti GPU, except the runtime complexity analysis is conducted on Nvidia 2080 Ti and 3090 Ti GPUs.

4.1 TEMPERATURE SCHEDULING FOR GUMBEL-SOFTMAX

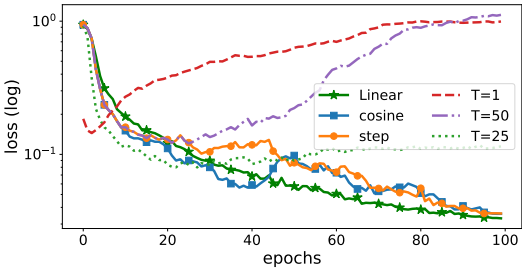
While deploying Gumbel-Softmax to enable differentiable discrete training for ONNs, the temperature value in Gumbel-Softmax is known as an important hyperparameter for the training performance. Thus, we first explore different temperature schedules in the proposed training framework for implementing ONNs with low-precision optical devices. Specifically, the results shown in Figure 2 are trained with experimental measured devices with 8 discrete values enabled (see Figure 1b

in Section 3). As discussed in Section 3, higher temperature leads to less variation between the discrete states, while lower temperature leads to a distribution closer to discrete states. Specifically, with higher temperature implemented in Gumbel-Softmax, a larger range of possible states will be explored as the distribution is more uniform over all possible states.

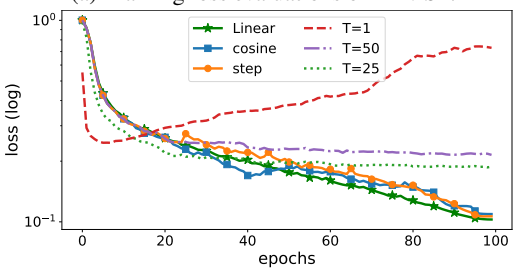
In this work, we mimic the concept of temperature scheduling in *simulated annealing* into our Gumbel-Softmax based training framework. To limit the exploration space of Gumbel-Softmax training, we evaluate six different temperature schedules, which all start with highest temperature $\tau_h = 50$, and lowest temperature $\tau_l = 1$ with 100 training epochs (Figure 2). First, we set three static temperature training as baselines for comparisons, which are trained with static temperature for the whole training process, i.e., (1) $\tau = 1$, (2) $\tau = 25$, and (3) $\tau = 50$. For dynamic temperature scheduling, we evaluate (4) linear temperature decaying scheduling (`Linear`), with temperature decay rate as 0.5 per epoch; and (5) cosine-annealing-decaying (`Cosine`) temperature schedule, where we set $\tau_{cosine} = [50, 40, 30, 20, 40, 30, 20, 30, 15, 5, 10, 1]$. With higher temperature, i.e., larger exploration space for the algorithm, it is expected to train more epochs. Thus, we set the training epochs for each temperature as $[10, 10, 10, 10, 10, 10, 10, 8, 7, 5, 5, 5]$; and (6) step temperature decaying (`Step`) the temperature schedule is set as $\tau_{step} = [50, 40, 30, 20, 10, 5, 1]$ with the training epochs per temperature as $[25, 20, 20, 15, 10, 5, 5]$.



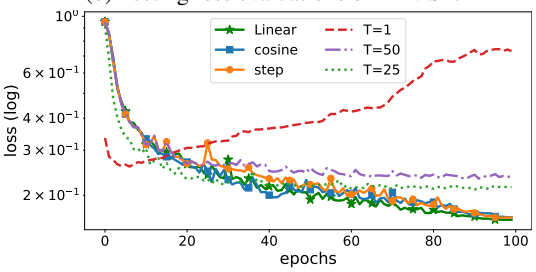
(a) Training loss evaluations on MNIST.



(b) Testing loss evaluations on MNIST.



(c) Training loss evaluations on F-MNIST.



(d) Testing loss evaluations on F-MNIST.

	`Linear`	`Cosine`	`Step`	$\tau = 50$	$\tau = 25$	$\tau = 1$
MNIST	0.997/0.979	0.994/0.978	0.995/0.978	0.138/0.133	0.930/0.926	0.143/0.146
FMNIST	0.941/0.891	0.935/0.888	0.938/0.889	0.857/0.844	0.876/0.854	0.446/0.444

(e) Final training/testing accuracy of all temperature schedules.

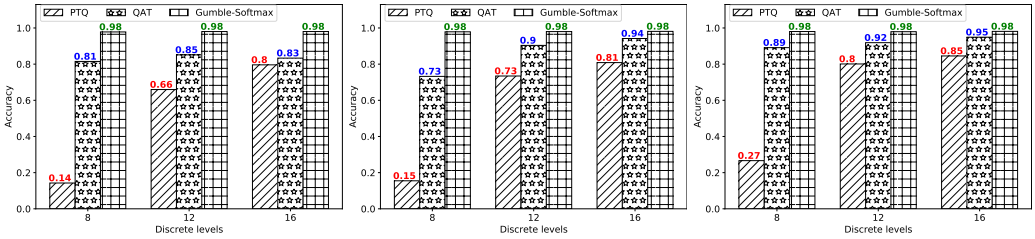
Figure 2: Evaluations of temperature scheduling in Gumbel-Softmax for training the proposed differentiable discrete device-to-system DONNs codesign framework.

In Figures 2(a)–(d), the training and testing processes are fully recorded for MNIST and FMNIST, with the x-axis representing the training iteration and the y-axis representing the loss value. The training and testing accuracy is shown in Figure 2(e). As we can see, for both MNIST and FMNIST, the system performs better with annealing temperature schedules in Gumbel-Softmax compared to the static temperature setups. To be more specific, in Figure 2, we can conclude from the results: **(1)** The system implemented with simulated annealing applied Gumbel-Softmax performs better than that implemented with static temperature applied Gumbel-Softmax. With bad static temperature implemented (e.g., $\tau = 50$ and $\tau = 1$ for MNIST dataset and $\tau = 1$ for FMNIST dataset), the system can get worse as training efforts increases. **(2)** The `Linear` temperature annealing schedule works

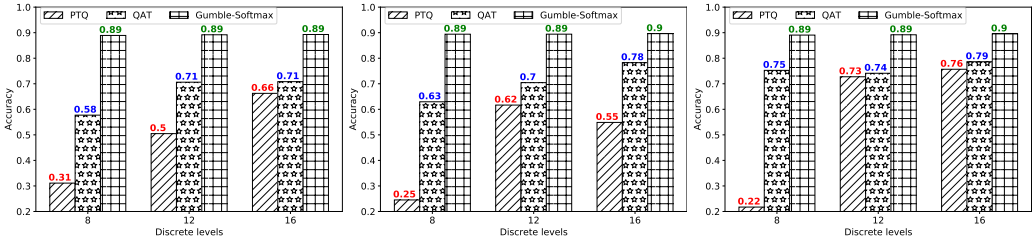
best for our system, as training and testing loss converge simultaneously and most efficiently. Thus, we deploy `Linear` temperature decaying in the rest of the section.

4.2 Comparisons with Conventional Quantization

In this section, we compare the performance of the proposed training approach with existing approaches used in experimental optical studies (Lin et al. (2018); Zhou et al. (2021)). Specifically, we experimentally measured discrete values of amplitude and phase modulation realized in the optical setups discussed in *System Overview*. The models evaluated in Figure 3 are trained with the proposed framework by feeding in these measured device parameters. During deployment, the trained discrete values are used to configure the device, which produces the complex-valued property of optical devices. We compared the proposed framework with the same system setup but implemented with fitted curve and conventional quantization method, i.e., straightforward post-training quantization (PTQ), quantization-aware training (QAT). For the system implemented with conventional quantization methods, we first fit a multi-polynomial using regression methods, which takes the device voltage value as inputs. For PTQ, we train the DONNs while considering the device voltage values being feasible in `float32` precision and round the voltages to the nearest discrete points after training. For QAT, instead of quantization after all training iterations, the values are rounded to the nearest discrete values after each training iteration, where the loss is calculated with the rounded voltages.



(a) 3-Layer D2NNs with MNIST-10 (b) 4-Layer D2NNs with MNIST-10 (c) 5-Layer D2NNs with MNIST-10



(d) 3-Layer D2NNs with FashionMNIST (e) 4-Layer D2NNs with FashionMNIST (f) 5-Layer D2NNs with FashionMNIST

Figure 3: Training performance comparisons between the proposed framework and leveraging conventional quantization methods for discrete device mapping in 3, 4, and 5-layer D2NNs using MNIST and FMNIST datasets.

As shown in Figure 3, three different co-design training algorithms are evaluated with 8, 12, and 16 discrete states in three DONNs systems with 3, 4, 5 diffractive layers, respectively. For the PTQ algorithm, when discrete states are as few as 8, the accuracy is unstable. To be fair, we collect the average accuracy of ten runs. First, within the same depth of DONNs systems, the models trained with the proposed Gumbel-Softmax enabled framework show clear advantages in all cases. More importantly, for the system implemented with different discrete states, we can see that the proposed framework is able to train the DONNs to match the best accuracy regardless of its complexity. Moreover, for systems with different structural complexity, Gumbel-Softmax shows its tremendous advantages in training, especially devices with fewer discrete states. **Our experimental results have demonstrated that the proposed differentiable discrete training framework offers significant training performance improvements over traditional quantization algorithms, especially in co-designing DONNs systems built very limited non-uniform distributed discrete states.**

4.3 COMPLEX-VALUED REGULARIZATION EVALUATION

As discussed in Section 3, in our DONNs system, regularization is required for system optimization. Thus, the regularization factor λ is introduced to regularize the gradients from amplitude and phase. Table 1 shows that **(1)** regularization will improve the optimization of the DONNs implemented with our system structure significantly, i.e., for MNIST dataset, $36\%/18\%/3\%$ accuracy improvement for the systems with 1/3/5 diffractive layers respectively and for FMNIST dataset, $27\%/15\%/8\%$ accuracy improvement for the systems with 1/3/5 diffractive layers respectively. **(2)** Regularization needs to be carefully adjusted for specific DONNs architecture. As it is discussed in section 3, in our system, the regularization factor needs to be decreased as the implemented diffractive layers increase. For the single-layer DONNs, the best regularization factor is $\lambda_1 = 90$, where the training accuracy can be 0.9874 and testing accuracy can be 0.9698. While applying regularization factors explored for deeper DONNs, the accuracy has significantly degraded. In this work, we empirically discover that regularization factor $\lambda_3 = 5$ and $\lambda_5 = 1.8$ work the best for three- and five-layer DONNs systems in image classification on MNIST and FMNIST. Also, we can see that inappropriate regularization factors can significantly degrade the training performance such that it requires careful exploration.

Table 1: Evaluations of the complex-valued regularization technique in training ONNs with 8 discrete values in the optical devices, with training/testing accuracy. *Results only include testing accuracy.

Dataset	DONNs Depth	$\lambda_1 = 90$	$\lambda_3 = 5$	$\lambda_5 = 1.8$	w/o reg* (Lin et al. (2018))
MNIST	Depth=1	**0.9691/0.9591**	0.0491/0.0466	0.0476/0.0445	0.67
	Depth=3	0.2095/0.2118	**0.9984/0.9781**	0.0408/0.0309	0.91
	Depth=5	0.0946/0.0938	0.9641/0.9594	**0.9971/0.9791**	0.95
FMNIST	Depth=1	**0.9083/0.8746**	0.1287/0.1175	0.1001/0.0955	0.54
	Depth=3	0.1937/0.1969	**0.9532/0.8892**	0.0162/0.0144	0.83
	Depth=5	0.1000/0.0941	0.7476/0.7416	**0.9412/0.8906**	0.87

As shown in Table 1, DONNs implemented with different diffractive layers can be trained to achieve similar prediction accuracy using our framework, which is a significant boost to the state-of-the-art results discussed by Lin et al. (2018). Although our framework can still achieve decent accuracy performance by adjusting the regularization factor, there are some fundamental limitations on the optical physics side in small depth DONNs. Thus, we further analyze the performance/robustness of the DONNs trained with different regularization factors. Specifically, we explore the confidence of the predictions acquired by the system. When the sample is classified correctly, we decrease the highest probability generated by softmax function (softmax of intensity values collected in the ten detector regions) by 1%, 3% and 5%, and then evenly distribute to the other nine outputs, i.e., increasing the probabilities of the other outputs by 0.11%, 0.33% and 0.55%, respectively. The results are shown in table 2. We can see that for both datasets, as the depth of DONNs increases, the prediction confidence increases, while the prediction accuracy are all relatively the same. For example, there is no accuracy degradation on five-layer DONNs for MNIST, and less than 1% degradation on FMNIST. However, for single-layer DONNs, the accuracy drops 63% for MNIST and 54% for FMNIST when 1% error applied and drops to 0 when applied error increases to 3% and 5%.

4.4 RUNTIME COMPLEXITY ANALYSIS

Finally, we analyze the runtime complexity of the proposed framework. As shown in Figure 1, the size of the trainable parameter is related to the system size and the number of discrete values in the devices. For a given DONNs system, the size of the training models will be linear to the number of discrete values in the devices. Thus, it is expected that the runtime complexity of the proposed method could be limited to the number of discrete values. In Figure 4, we analyze the runtime complexity of training a five-layer DONNs architecture, where we change the number of discrete values in the optical devices[1]. The left x-axis represents the runtime per training epoch on Nvidia 2080 Ti, and

[1]Here we use synthetic devices values since we only evaluate the training runtime complexity.

Table 2: Confidence evaluation of DONNs trained with complex-domain regularization.

dataset	DONNs Depth	0	1%	3%	5%
MNIST	Depth=1	0.9600	0.3260	0	0
	Depth=3	0.9820	0.9640	0.850	0.5880
	Depth=5	0.9860	0.9860	0.9860	0.9860
FMNIST	Depth=1	0.8780	0.3400	0	0
	Depth=3	0.8940	0.8120	0.502	0.264
	Depth=5	0.8940	0.8940	0.892	0.8900

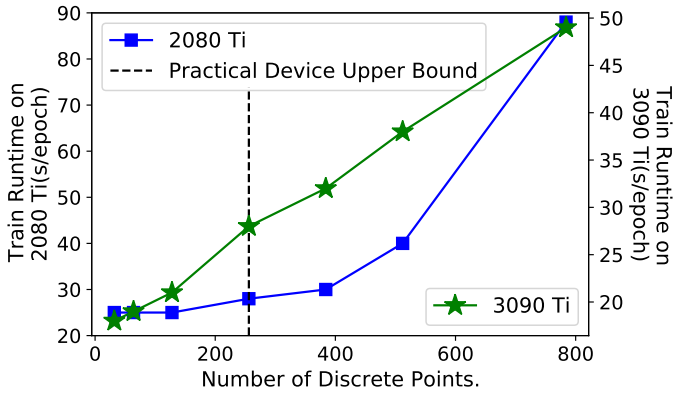


Figure 4: Runtime analysis of Gumbel-Softmax based discrete optical neural network training, measured on 2080 Ti and 3090 Ti GPU. The complexity of training a 5-layer ONNs is almost **linear** to the number of discrete size of the targeted hardware devices on 3090 Ti, and **linear** on 2080 Ti up to device with less than 384 discrete values.

the right x-axis represents the runtime on 3090 Ti. We can see that the training runtime is almost linear on 3090 Ti up to 784 discrete values due to the more efficient memory hierarchy in the newer GPU design. While using 2080 Ti, we start observing quadratic runtime increases for more than 512 discrete values. However, note that the number of discrete values in real-world optical devices are mostly limited to 256 voltage values and are mostly limited to 16 states for high-speed devices (Ríos et al. (2019); Sebastian et al. (2020)).

5 Conclusion

This work proposes a novel flexible device-to-system hardware-software codesign framework which enables efficient training of DONNs systems implemented with arbitrary experimental measured optical devices across the layers. Specifically, this framework realizes backpropagation through discrete parameters via Gumbel-Softmax and improves the training performance for DONNs systems by introducing a novel complex-domain regularization technique. Our experiments demonstrate that the DONNs system optimized with the proposed framework will acquire tremendous accuracy improvements compared to the state-of-the-art experimental studies. Moreover, exploration for temperature schedule for Gumbel-Softmax in DONNs system, confidence evaluation for the same system architecture implemented with different complex-domain regularization factor and runtime complexity analysis are comprehensively discussed.

Acknowledgments

Removed for review.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale Machine Learning. *Symp. on Operating System Design and Implementation (OSDI)*, pp. 265–283, 2016.

J Feldmann, N Youngblood, C David Wright, H Bhaskaran, and WHP Pernice. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature*, 569(7755):208–214, 2019.

Johannes Feldmann, Nathan Youngblood, Maxim Karpov, Helge Gehring, Xuan Li, Manuel Le Gallo, Xin Fu, Anton Lukashchuk, Arslan Raja, Junqiu Liu, et al. Parallel convolution processing using an integrated photonic tensor core. *Nature (to appear)*, 2020.

Weilu Gao, Cunxi Yu, and Ruiyang Chen. Artificial intelligence accelerators based on graphene optoelectronic devices. *Advanced Photonics Research*, 2(6):2100048, 2021a.

Weilu Gao, Cunxi Yu, and Ruiyang Chen. Graphene optoelectronic artificial intelligence accelerators. In *CLEO: QELS_Fundamental Science*, pp. JTu3A–88. Optical Society of America, 2021b.

Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter Performance Analysis of a Tensor Processing Unit. *Int'l Symp. on Computer Architecture (ISCA)*, pp. 1–12, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Yingjie Li, Ruiyang Chen, Berardi Sensale Rodriguez, Weilu Gao, and Cunxi Yu. Multi-task learning in diffractive deep neural networks via hardware-software co-design. *Scientific Reports*, pp. 1–9, 2021.

Xing Lin, Yair Rivenson, Nezih T Yardimci, Muhammed Veli, Yi Luo, Mona Jarrahi, and Aydogan Ozcan. All-optical machine learning using diffractive deep neural networks. *Science*, 361(6406): 1004–1008, 2018.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Deniz Mengu, Yi Luo, Yair Rivenson, and Aydogan Ozcan. Analysis of diffractive optical neural networks and their integration with electronic neural networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 26(1):1–14, 2019.

Deniz Mengu, Yair Rivenson, and Aydogan Ozcan. Scale-, shift-and rotation-invariant diffractive optical networks. *arXiv preprint arXiv:2010.12747*, 2020.

Md Sadman Sakib Rahman, Jingxi Li, Deniz Mengu, Yair Rivenson, and Aydogan Ozcan. Ensemble learning of diffractive optical networks. *arXiv preprint arXiv:2009.06869*, 2020.

Carlos Ríos, Nathan Youngblood, Zengguang Cheng, Manuel Le Gallo, Wolfram HP Pernice, C David Wright, Abu Sebastian, and Harish Bhaskaran. In-memory computing on a photonic platform. *Science advances*, 5(2):eaau5759, 2019.

Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature nanotechnology*, 15(7):529–544, 2020.

Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

Hardik Sharma, Jongse Park, Emmanuel Amaro, Bradley Thwaites, Praneetha Kotha, Anmol Gupta, Joon Kyung Kim, Asit Mishra, and Hadi Esmaeilzadeh. Dnnweaver: From high-level deep network models to fpga acceleration. In *the Workshop on Cognitive Architectures*, 2016.

Yichen Shen, Nicholas C Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, et al. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7):441, 2017.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Alexander N Tait, Thomas Ferreira De Lima, Ellen Zhou, Allie X Wu, Mitchell A Nahmias, Bhavin J Shastri, and Paul R Prucnal. Neuromorphic photonic networks using silicon photonic weight banks. *Scientific reports*, 7(1):1–10, 2017.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Tiankuang Zhou, Xing Lin, Jiamin Wu, Yitong Chen, Hao Xie, Yipeng Li, Jingtao Fan, Huaqiang Wu, Lu Fang, and Qionghai Dai. Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. *Nature Photonics*, 15(5):367–373, 2021.