

High-performance real-world optical computing trained by in situ gradient-based model-free optimization

Guangyuan Zhao, *Student Member, IEEE*, Xin Shu, *Member, IEEE*, and Renjie Zhou, *Member, IEEE*

Abstract—Optical computing systems provide high-speed and low-energy data processing but face deficiencies in computationally demanding training and simulation-to-reality gaps. We propose a gradient-based model-free optimization (G-MFO) method based on a Monte Carlo gradient estimation algorithm for computationally efficient in situ training of optical computing systems. This approach treats an optical computing system as a black box and back-propagates the loss directly to the optical computing weights' probability distributions, circumventing the need for a computationally heavy and biased system simulation. Our experiments on diffractive optical computing systems show that G-MFO outperforms hybrid training on the MNIST and FMNIST datasets. Furthermore, we demonstrate image-free and high-speed classification of cells from their marker-free phase maps. Our method's model-free and high-performance nature, combined with its low demand for computational resources, paves the way for accelerating the transition of optical computing from laboratory demonstrations to practical, real-world applications.

Index Terms—Computational Optics; Optical Neural Network; Model-free Optimization;

1 INTRODUCTION

OPTICAL computing leverages the properties of light waves to facilitate high-speed data processing while reducing the energy cost [1], [2], [3], [4], [5], [6], [7]. Recent advances in automatic differentiation have enabled in silico training of large-scale optical computing weights, giving rise to the realizations of diffractive neural networks [8], [9], optical reservoir computing [10], [11], and coherent nanophotonic circuits [12].

Problem Statement. Training optical computing systems presents two challenges: an intensive computational process and a performance disparity between simulation and reality when implementing pre-trained weights onto real-world systems [9], [13], [14]. Typically, the optical computing systems are trained in silico using differentiable simulators rooted in the first principle of optics, an approach known as simulator-based training (SBT). While SBT has proven effective within the confines of the simulator, the performance in real systems is largely contingent upon the simulator's fidelity. Factors such as misalignment and aberration, often omitted in simulations, cause significant performance degradation when optical computing weights trained exclusively within the simulator are applied to real-world systems.

To bridge the reality gap between simulation and experiments, physics-aware training (PAT) and hybrid training

(HBT) have been introduced [15], [16] recently. Both training strategies include conducting the forward pass in the real-world system and back-propagating the loss from the system to its weights through the simulator. These in situ approaches allow the training process to access the optical computing system during the forward pass, which leads to more accurate weight updates than in silico training [13].

Despite these recent advances, there is a continued reliance on a physics-based simulator during the back-propagation process in current in situ training methods. Such a setting brings three drawbacks: (1) the bias between the simulator and real system prohibits the above training process from achieving optimal results; (2) the in silico simulation requires large memory and computation, limiting the aforementioned methods from in situ training in edge devices with limited computing resources [17]; and (3) the model-based training strategies need high-fidelity images of the input object, which are costly to acquire in terms of instruments, time and storage memory. Hence, developing a memory- and computation-efficient algorithm to train the optical computing system efficiently is an open problem we address in this paper.

Proposed Solution. Here, we propose an in situ model-free optimization method that does not require back-propagating errors through the simulator. Instead, we use a score gradient estimation algorithm [18], [19] to solely use the forward outputs from the real system to get gradients for updating the weights of the optical computing system. As shown in Fig. 1, our method treats the optical system as a black box and back-propagates task-specific negative-loss as rewards to the source weight distributions (Fig. 1a). This process only requires knowledge of the weights and forward outputs of the optical computing system, unlike the SBT and HBT methods that require a simulator and the images of the input objects (Fig. 1b).

- This work is supported in part by Hong Kong General Research Fund 14209521, in part by Hong Kong Innovation and Technology Fund ITS/178/20FP & ITS/148/20 and in part by Croucher Foundation CM/CT/CF/CIA/0688/19ay. (G. Zhao and X. Shu contributed equally to this work.) (Corresponding author: Guangyuan Zhao.)
- G. Zhao, X. Shu and R. Zhou are with the Department of Biomedical Engineering, the Chinese University of Hong Kong, Hong Kong, China (email: zhaoguangyuan@link.cuhk.edu.hk, shuxin@link.cuhk.edu.hk, rjzhou@cuhk.edu.hk).
- This article has supplementary material available at xxx, provided by the authors.

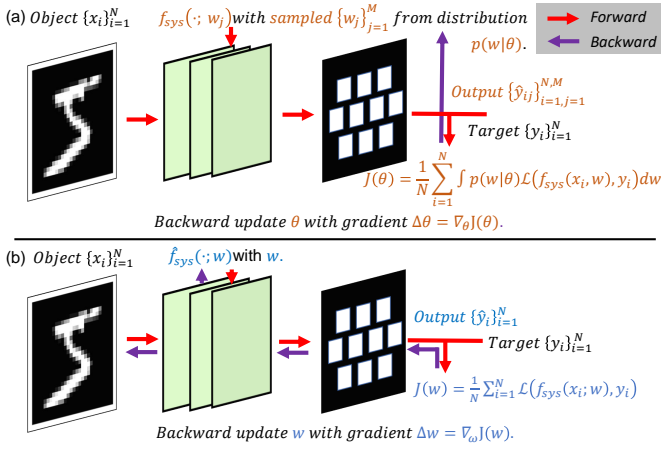


Fig. 1: Gradient-based model-free optimization based training of the optical computing system. (a) The brown highlights show that our gradient-based model-free training strategy back-propagates the error of training to the distribution parameter θ and bypasses the reliance on correct differentiable modeling of the optical system f_{sys} and knowledge on input $\{x_i\}_{i=1}^N$. (b) The blue highlights show that the conventional training of the optical computing system relies on a physics-based simulator \hat{f}_{sys} , which substitutes the inaccessible f_{sys} corresponding to the real system. The training process back-propagates the loss through simulator \hat{f}_{sys} to update the weight w . This is the basis of SBT and HBT methods.

Contributions. We make the following contributions:

- We *introduce* a score-gradient estimation method to train the optical computing system in a model-free manner, providing a computation- and memory-efficient way to mitigate the reality gap to training the real optical computing system (Sec. 3).
- We *validate* our method on a diffractive optical computing system. Experimental results show that G-MFO outperforms hybrid training on the commonly used MNIST and FMNIST datasets [9], [15], [16] (Sec. 4.1).
- We *show* the G-MFO training process only consumes $\sim 0.1\%$ of the GPU time and memory compared with the HBT in situ method by avoiding the computation-intensive modeling of the wave propagation process (Sec. 4.2).
- Application scenario: We *demonstrate*, as a proof-of-concept, that our G-MFO-trained optical computing system's effectiveness on classifying four types of white blood cells from their phase maps in a *marker-free*¹ manner with a testing accuracy of 73.8%, making it a promising approach for image-free, marker-free, and high-speed cell analysis (Sec. 4.4).

Open source. Our code will be available at repository Model-free-Computational-Optics to facilitate further research and applications on computational optics and train-

1. Herein, we use "marker-free" but not "label-free" to avoid confusion. This is because "label-free" has different meanings in biomedical research and high-level vision.

ing of optical computing systems. We hope to promote the reproducibility and follow-ups for this hardware-centric research direction.

2 RELATED WORK

We discuss the following work related to our contributions.

2.1 In situ training strategies to optimize the optical computing system

Most research into optical computing system training has historically relied on in silico training. This process involves forward and backward propagation calculations on an external computer that simulates the physical system through a digital twin [9], [12]. However, this technique can lead to discrepancies between the simulation and actual reality due to inaccuracies in the physical system's representation.

In recent years, progress has been made in developing in situ training strategies that use data gathered from real-world optical computing systems to mitigate the reality gap and improve experimental performance. As a first attempt, we categorize previous and our in-situ training methods into two genres: model-based optimization (MBO) and model-free optimization (MFO) methods. We differentiate these two genres by whether they use the optical computing system emulator models during the parameter updating process. A tabular comparison of the previous and our methods is in Tab. 1. Details of the illustrations and comparisons are below.

MBO methods. *Physics-aware training (PAT)* [15], *hybrid training (HBT)* [16] or *adaptive training (AT)* [20] (*H-MBO*) are one type of the MBO methods that conduct forward pass in a real system and back-propagate the gradients with the aid of a physics-based simulator. *Backpropagation through learned model (L-MBO)* is another method that gathers the real data from the optical computing system to train a neural proxy of the optical computing system [21], [22]. With such a learned proxy rather than the physics-based model utilized in H-MBO, the backward parameter update process can reduce the bias from the model mismatch between the simulator and the real system in L-MBO. During training, one has to optimize both the task's and proxy's parameters to optimize the task performance and improve the fidelity of the simulator. The task performance for the MBO, especially the L-MBO, is high while back-propagating with the digital twin also burdens the training process. The L-MBO methods implemented in optical computing are referred as *hardware (camera)-in-the-loop methods*. These methods have recently been used in computational display [31], [32] and camera ISP [33]. They are also known as *real2sim* in photolithography for solving computational lithography tasks [34], [35]. Despite the computational burden introduced by the emulation process from the digital twin, another disadvantage of the MBO methods is that all the MBO approaches require images of input objects during training, adding further burden to the overall process.

MFO methods. Unlike the aforementioned MBO methods, MFO methods do not require constructing a digital twin of the optical computing system; they directly use the real system's output or intermediate measurements to

	Conventional	MBO		MFO				
	SBT [9], [12]	H-MBO [15], [16], [20]	L-MBO [21], [22]	IMOB [23], [24], [25]	FFT [26], [27]	Trial and Error [28], [29]	GA [30]	G-MFO (ours)
In situ	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Computation overhead	High	High	High	Low	Low	Low	Low	Low
Input image free	No	No	No	Yes	No	Yes	Yes	Yes
In silico training time	High	High	High	Low	Low	Low	Low	Low
Dimensionality	High	High	High	High	High	Low	Low	Medium
Intermediate measurement	No	No	No	Yes	Yes	No	No	No
Gradient	Yes	Yes	Yes	Yes	No	No	No	Yes

TABLE 1: **Comparison of strategies on training optical computing systems** along the axes of in situ training capability (in situ), in silico computation overhead (computation overhead), the requirement on the recording of the input object to the task (input image free), required in silico training time, the dimensionality of trainable parameters (dimensionality), requirements on intermediate measurement and whether or not use gradient-based update that is more efficient (gradient). Simulator-based training (SBT), Hybrid model-based training (H-MBO), Learned model-based training (L-MBO), Intermediate measurement optical backpropagation (IMOB), Forward-forward training (FFT), Trial and error, Genetic algorithm (GA), Gradient-based model-free optimization (G-MFO).

update its parameters. *Population-based algorithms* such as genetic algorithm (GA) methods [30] use the output from the optical computing system to optimize parameters by simulating the process of natural selection. Still, the genetic algorithm's performance scales poorly to the dimensionality of parameters [36]. *Forward-forward training* (FFT) is a greedy multi-layer learning procedure for mortal computation [37] and it has been applied to train optical computing systems recently [26], [27]. However, labels must be added to each input sample during the training and testing. As a result, such training strategies require masking the input, which might be unrealistic in many scenarios, and the testing process of forward-forward training is prolonged. In *Trial and error*, perturbed parameters that enhance performance are retained, while those that do not are discarded, reverting to the previous state [28], [29]. This method, however, does not scale well with the dimensionality of parameters. *Intermediate measurement optical backpropagation* (IMOB) proposed using the adjoint variable method or direct feedback alignment, etc., to calculate gradients [23], [24], [25]. However, such methods require intermediate system measurements for every layer's output, which might be infeasible or require additional measurement setups.

We propose the gradient-based MFO (G-MFO) training method using Monte Carlo gradient estimation to update the optical computing system parameters automatically. Our work shares similarities with Zhang *et al.*, which uses the genetic algorithm for weight update [30], as our and their methods both generate a batch of sampled parameters during the training process. The difference is that our parameter updating process is more efficient owing to a score gradient estimator rather than their heuristic hill-climbing type algorithm. Moreover, our method does not need specific conditions, such as changing the system or manipulating the input object, which are more or less required in other MBO and MFO methods.

2.2 Zeroth-Order Optimization and Its Applications to Computational Optics

The zeroth-order (ZO) optimization uses finite differences from forward evaluations instead of the first-order path-wise gradient to calculate the gradient for updating the

parameters of interest [38], [39]. ZO has been utilized in many areas, such as differentiating through discontinuities in computer graphics [40], optimizing contact dynamics in robotics [41], [42], and policy learning in reinforcement learning [19]. ZO optimization has been implemented with both full-point estimation [43], and stochastic two-point estimation [44]. A significant advancement in ZO is to estimate the descent direction using multiple stochastic finite differences [45] where Monte Carlo approximation is used in the process, also called evolutionary strategies [46]. Moreover, techniques such as antithetic sampling, control variates [47], or more advanced methods to reduce variance [48], [49] in the Monte Carlo approximation process are used in multiple stochastic finite differences. Our work employs Monte Carlo gradient estimation [38], which resembles multiple stochastic finite differences.

Applications of ZO Methods to Computational Optics have been relatively sparse compared to first-order methods. In the past, there have been demonstrations in computational optics tasks such as lens design [50], auto-tuning structured light for depth estimation [51], and computer-generated holography [52]. Using only forward evaluations for gradient updates makes it beneficial for optical computing-related tasks, such as what was mentioned as optical stochastic gradient descent (SGD) in the auto-tuning structured light work [51]. The reason is that forward evaluation in optical computing is inexpensive while finding the first-order gradient is computationally costly and prone to errors. Our work highlights the advantages of using zeroth-order optimization for optimizing more complex systems, such as optical computing, and the efficacy of using Monte Carlo gradient estimation for efficient optimization.

2.3 Application side of optical computing

Optical computing, outperforming electrical computing in parallelism and processing speed, is pivotal for AI progress via optical neural networks (ONN) development [5]. ONNs encode inputs using light's spatial, temporal, and spectral characteristics to enable applications such as image classification, motion detection, and medical diagnosis [29], [53], [54]. Furthermore, optical computing functions as an advanced neural network platform and directly processes

optical signals, facilitating applications like phase reconstruction [55], denoising [56], edge detection [57], and uni-directional imaging [58].

As a secondary contribution, our work showcases the application of optical computing for marker-free cell classification. Particularly relevant to our research is the study by Wang *et al.* [59], which demonstrates the classification of fluorescent-labeled cells using an optical computing system. In contrast, our approach shows cell-classification based on their phase maps, instead of relying on fluorescent markers. This illustrates that optical computing can facilitate low-storage, high-speed, and marker-free cell analysis.

3 METHODOLOGY

In what follows, we mathematically detail the problem setup on training the optical computing system and our solution. We introduce problem formation in Subsec. 3.1 and the conventional solution of using simulator-based training (SBT) in Subsec. 3.2. We then illustrate our solution to the problem, the gradient-based model-free optimization (G-MFO) for training the optical computing system in Subsec. 3.3. Finally, we illustrate the optical computing system and the related simulators in Subsec. 3.4, which we will use to demonstrate the performance of our method.

3.1 Problem setup

We are interested in learning the optimal weight $w \in \mathbb{R}^H$ for the optical computing system on a desired task with a training dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where N is the size of the dataset, H is the number of trainable parameters in w , and x and y denote the input and target of interest, respectively. A function $f_{sys}(\cdot, w)$ maps $x \rightarrow y$ through this optical computing system with w . Specifically, in the image classification task based on the diffractive optical computing system we work on, f_{sys} denotes the optical mapping from the input image x to the output label y , and w is the optical computing weight in the form of phase value modulation.

During training, one can minimize the cost function $J(w)$ as the mean task-specific loss across the entire training data set \mathcal{D} :

$$\arg \min_w J(w) := \mathbb{E}[\mathcal{L}(\mathcal{D}, w)], \quad (1a)$$

$$= \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{sys}(x_i, w), y_i), \quad (1b)$$

where \mathcal{L} is the task-specific loss function, we use cross-entropy loss [60] since we deal with image classification tasks throughout this paper.

We use gradient descent-based search to find the optimal w to minimize the objective function $J(w)$:

$$w = w - \alpha \nabla_w J(w), \quad (2)$$

where ∇_w represents the gradient operator that collects all the partial derivatives of a function concerning parameters in w , and α is the learning rate.

It is straightforward to use the backpropagation method [14] to take the gradient through f_{sys} and finds the gradient $\nabla_w J(w)$ as:

$$\nabla_w J(w) = \frac{1}{N} \sum_{i=1}^N \nabla_w \mathcal{L}(f_{sys}(x_i, w), y_i), \quad (3)$$

when we have an accurate and differentiable f_{sys} modeling. However, this is the case for training digital neural networks [14], but not when we train a real-world optical computing system. **Thus, this paper's critical aim is finding an accurate gradient estimation of $\nabla_w J(w)$ to update optical computing weight w in a real-world optical system.**

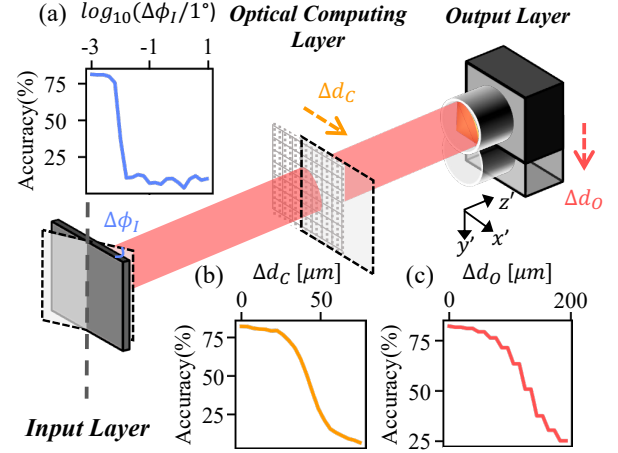


Fig. 2: **System misalignment in a real optical computing system degenerates the performance of the optical computing system trained solely with a physics-based simulator.** (a) Testing accuracy drops to 36.4% from 82.2% when having a misaligned rotation angle $\Delta\phi_I = 0.01^\circ$. (b) Testing accuracy drops to 51.0% from 82.2% when the x' -axis misalignment of the optical computing layer Δd_c is $41.1 \mu\text{m}$. (c) Testing accuracy decreases to 50.9% from 82.2% when the y' -axis misalignment of the output layer Δd_o is $62.4 \mu\text{m}$.

3.2 In silico simulator-based training (SBT)

Back-propagation through the simulator \hat{f}_{sys} . In a real-world optical computing system, since we do not have an exact functional expression of f_{sys} , the **simulator-based training** (SBT) builds a simulator \hat{f}_{sys} as the differentiable approximation of f_{sys} (Fig. 1b). A naive training strategy is substituting f_{sys} in Eq. 3 with the simulator \hat{f}_{sys} and applying in silico training on the simulator:

$$\nabla_w J(w) = \frac{1}{N} \sum_{i=1}^N \nabla_w \mathcal{L}(\hat{f}_{sys}(x_i, w), y_i). \quad (4)$$

The resulting $\nabla_w J(w)$ is used in Eq. 2 to update w . After training, the optimized w is uploaded to the real optical computing system f_{sys} to test the performance.

Simulation-to-reality gap. The aforementioned simulator-based training is based on backpropagation through the simulator \hat{f}_{sys} . The “sim2real” gap is low (i.e., the gradient $\nabla_w J(w)$ is an accurate estimation)

when the simulator \hat{f}_{sys} is similar to f_{sys} . However, this assumption does not hold in many prototypes of optical computing systems where inadequate modeling and misalignment between the optical elements decay the performance of the SBT during the "sim2real" transfer. We use a simulator described in Subsec. 3.4 to assess the adverse effect of misalignment on the image classification results. We impose various misalignments to a well-trained ideal optical computing system and measure drops in classification accuracy. For instance, we show in Fig. 2 that slightly laterally misaligning the optical computing layer by $41.1 \mu m$ reduces the classification accuracy by 31.2%.

3.3 In situ G-MFO

Our solution for solving the aforementioned "sim2real" gap issue in Subsec. 3.2 is in situ learning the optical computing weight w with the gradient-based model-free optimization (G-MFO). In situ learning enables us to access the output of f_{sys} and is feasible on the hardware side because we can use programmable devices such as spatial light modulators to update w . The challenging part is designing a training strategy that efficiently uses the output of actual system f_{sys} to construct an unbiased gradient estimator. Here, we use the score gradient estimator to calculate the gradient [38], [61] for the backward update of parameters in w while circumventing the construction of \hat{f}_{sys} , a biased and resource-intensive numerical modeling of f_{sys} .

Back-propagation through the weights distribution p . In our score gradient estimator for model-free optimization, we model optical computing weight w as a random variable that follows a probability distribution: $w \sim p(w|\theta)$ and rewrite the cost function $J(w)$ in Eq. 1b as a probability cost function $J(\theta)$:

$$\begin{aligned} \arg \min_{\theta} J(\theta) &:= \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{p(w|\theta)} [\mathcal{L}(f_{sys}(x_i, w), y_i)], \quad (5a) \\ &= \frac{1}{N} \sum_{i=1}^N \int p(w|\theta) \mathcal{L}(f_{sys}(x_i, w), y_i) dw, \quad (5b) \end{aligned}$$

where the probability distribution $p(w|\theta)$ is continuous in its domain and differentiable concerning its distribution parameter θ . Accordingly, the original goal of optimizing w is reformulated as finding a most likely distribution $p(w|\theta)$ that minimizes the cost function $J(\theta)$ in Eq. 5. Specifically, we model the distribution p as a multivariate normal distribution with $\theta = \{\mu, \sigma^2\}$ and $p(w|\theta) = \mathcal{N}(w; \mu, \sigma^2)$ for optimizing the continuous phase-valued weight to be uploaded onto the SLM in our work.

To update distribution parameter θ with the gradient descent Eq. 2, we take the gradient of the cost function $J(\theta)$ in Eq. 5:

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} \frac{1}{N} \sum_{i=1}^N \int p(w|\theta) \mathcal{L}(f_{sys}(x_i, w), y_i) dw, \quad (6a)$$

$$= \frac{1}{N} \sum_{i=1}^N \int \mathcal{L}(f_{sys}(x_i, w), y_i) \nabla_{\theta} p(w|\theta) dw. \quad (6b)$$

Applying the log derivative trick, we have:

$$\nabla_{\theta} J(\theta) = \frac{1}{N} \sum_{i=1}^N \int p(w|\theta) \mathcal{L}(f_{sys}(x_i, w), y_i) \nabla_{\theta} \log p(w|\theta) dw. \quad (7)$$

Then we apply Monte Carlo integration to approximate the integral value in Eq. 7 by first drawing M independent samples $\{w_j\}_{j=1}^M$ from the distribution $p(w|\theta)$ and then computing the average function value evaluated in these samples:

$$\nabla_{\theta} J(\theta) = \frac{1}{N} \sum_{i=1}^N \frac{1}{M} \sum_{j=1}^M \mathcal{L}(f_{sys}(x_i, w_j), y_i) \nabla_{\theta} \log p(w_j|\theta), \quad (8a)$$

$$= \frac{1}{M} \sum_{j=1}^M \left[\frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{sys}(x_i, w_j), y_i) \right] \nabla_{\theta} \log p(w_j|\theta), \quad (8b)$$

$$= \frac{1}{M} \sum_{j=1}^M r(w_j) \nabla_{\theta} \log p(w_j|\theta). \quad (8c)$$

Here we define $r(w_j) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{sys}(x_i, w_j), y_i)$ as the negative reward corresponding to each weight w_j . We use Equation 8c as the score gradient estimator for G-MFO, where the score function is $\nabla_{\theta} \log p(w_j|\theta)$, which has been widely used in other areas, such as policy gradient algorithms in reinforcement learning [19] and diffusion models [62].

Variance reduction. The main risk of using the score gradient estimator is the high variance that comes from the Monte Carlo integration step that transits Eq. 7 to Eq. 8a. Such a sampling-based integration step has high variance because different sets of random samples may lead to significantly different integral estimates. We reduce the variance by subtracting the $r(w_j)$ with baseline value $\bar{r} = \frac{1}{M} \sum_{j=1}^M r(w_j)$ while keeping the bias of gradient unchanged [63]:

$$\nabla_{\theta} J(\theta) = \frac{1}{M} \sum_{j=1}^M (r(w_j) - \bar{r}) \nabla_{\theta} \log p(w_j|\theta). \quad (9)$$

Training recipe of G-MFO. We perform the following steps in each training epoch (see Fig. 3): 1. Sample a group of phase-valued optical computing weights $\{w_j\}_{j=1}^M$ from the distribution $w_j \sim p(w|\theta)$ and upload them to the optical computing layer. Upload classification data $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ to the input layer; 2. Collect the optical computing system's output $\{f_{sys}(x_i, w_j)\}_{i=1, j=1}^{N, M}$ for each pair of input and weight; 3. Update the distribution parameter θ in silico using Eqs. 2 and 9; 4. Sample a new group of $\{w_j\}_{j=1}^M$ from the updated distribution $w_j \sim p(w|\theta)$, which is ready to be uploaded to the optical computing layer in the next epoch. The algorithm iterates these steps until convergence.

After minimizing the objective function Eq. 5, we export w_j with the smallest $r(w_j)$ as the output weight w^* . This performs better than setting w^* as the sampled mean μ . An algorithmic overview of the above training recipe is shown in Algorithm 1.

Algorithm 1 Algorithmic overview of G-MFO.

```

1: Input: Classification dataset  $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ , learning
   rate  $\alpha$ , number of sampled weights  $M$ , optical computing
   system  $f_{sys}$ , distribution parameter  $\theta = \{\mu, \sigma^2\}$ , loss
   function  $\mathcal{L}$ , epochs  $K$ .
2: Output: Optimized optical computing weight  $w^*$ .
3: for  $k$  in range  $K$  do
4:   Sample  $\{w_j\}_{j=1}^M$  from distribution  $p(w|\theta)$ .
5:    $\triangleright$  in situ evaluate  $\{w_j\}_{j=1}^M$ .
6:   for  $j$  in range  $M$  do
7:
8:      $r(w_j) \leftarrow \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_{sys}(x_i, w_j), y_i)$ .
9:   end for
10:   $\triangleright$  in silico update  $\theta$ .
11:  Calculate  $\nabla_{\theta} J(\theta)$  via Eq. 9.
12:   $\theta \leftarrow \theta - \alpha \nabla_{\theta} J(\theta)$ .
13: end for
14:  $w^* \leftarrow w_j$  with the smallest  $r(w_j)$ .

```

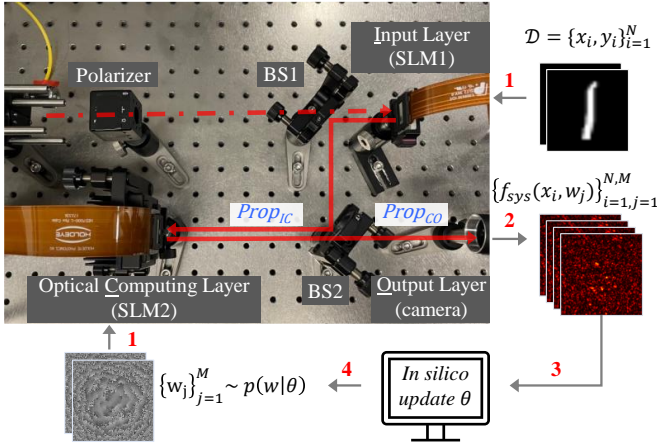


Fig. 3: A visual illustration of the training process for the G-MFO experiment on the real system.

3.4 Experiment and Simulation Details of Our Diffractive Optical Computing System

We validate the effectiveness of the proposed G-MFO-based training strategy on a single-layer optical computing system and a two-layer diffractive optical computing system. Our work focuses on the training strategy rather than the design or advantage of a specific optical computing system. Therefore, we do not consider nonlinearity in constructing our optical computing system, which has recently been implemented using saturation effects [20], [59], [64]. The validity of using a multi-layer versus a single-layer system without nonlinearity has been analyzed empirically in [65], where the authors attribute the improvement to increased energy resulting from the multi-layer settings. Details on the experimental setups ($f_{sys-1-layer}$, $f_{sys-2-layer}$) and the corresponding simulators ($\hat{f}_{sys-1-layer}$, $\hat{f}_{sys-2-layer}$) are in the Supplement Sec. S.1.

4 RESULTS

We aim to answer the following research questions in this section:

- How does G-MFO perform in the numerical simulations and experiments?
- What are the advantages and limitations of G-MFO?
- How is G-MFO's applicability on the marker-free cell classification task?

Thus, we numerically and experimentally evaluate the general performance of our G-MFO method on the open-source MNIST and FMNIST datasets (Subsec. 4.1). We then illustrate G-MFO's advantage of memory- and computation-efficient training in Subsec. 4.2. Moreover, we analyze the limitation of our method in Subsec. 4.3. Lastly, we demonstrate the G-MFO method on a novel application of marker-free classifying white blood cells in Subsec. 4.4.

We compare our method with the following baselines:

- **Ideal:** The in silico simulation results without introducing any artificial misalignment during the simulation. This corresponds to the best achievable result for an optical computing system.
- **SBT:** Simulator-based training with digitally aligned simulator.
- **HBT:** The hybrid training strategies with a physics-based simulator.
- **L-MBO:** We pre-train a proxy using the task dataset to emulate the real system and then optimize the optical computing task based on the pre-trained proxy.
- **Zeroth-order optimization baselines:** Including two-point estimation on a random direction [44] and full-point estimation [43], [51].

The training details for simulations and experiments are in Supplement Sec. S.3. Supplement Sec. S.6 provides a detailed description of our implementation of the HBT baseline. Supplement Sec. S.7 discusses and compares L-MBO.

4.1 General performance evaluation on the MNIST and FMNIST dataset

4.1.1 Simulation comparison between G-MFO and other zeroth-order optimization methods in simulation on a small dataset

We use a single-layer optical computing simulator to compare our method with two zeroth-order (ZO) baselines: the full-point ZO method [43], [51] and the two-point ZO method [44]. This comparison mainly evaluates the training performance, including accuracy, execution, and convergence speed. Therefore, we use a small dataset of 100 training, 100 validation, and 100 testing samples.

We present the simulation results and training time per epoch in Table 2, and the curves of training accuracy in Fig. 5. The results show that sufficient samples are necessary for successful optimization. Specifically, the two-point method, with two samples per epoch, only achieves a training accuracy of 22%, whereas both the full-point method, using 128×128 samples per epoch, and the G-MFO method, using 128 samples per epoch, achieve a training accuracy of 99%. Additionally, G-MFO has improved sample efficiency than the full-point optimization method, requiring only 4%

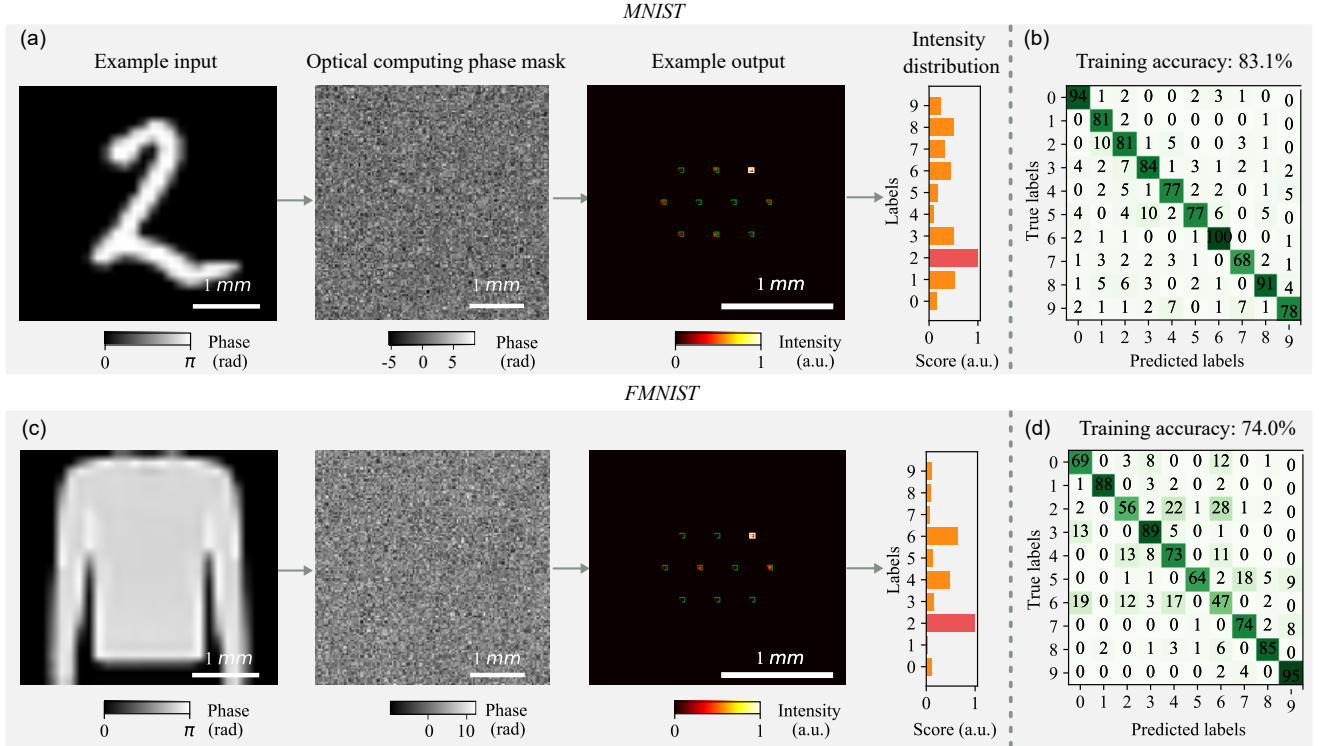


Fig. 4: **Visualization of the optical computing system's experimental outputs and confusion matrices trained with G-MFO.** (a) An input phase object digit '2' from the MNIST dataset is modulated by the optical computing layer with weight w trained using G-MFO. The system correctly predicts the input as digit '2', as the output image has the largest intensity at the region corresponding to digit '2'. (b) Confusion matrix on the MNIST dataset with a training accuracy of 83.1%. (c) An example of a 'pullover' from the FMNIST dataset is correctly predicted. (d) Confusion matrix on the FMNIST dataset with a training accuracy of 74.0%.

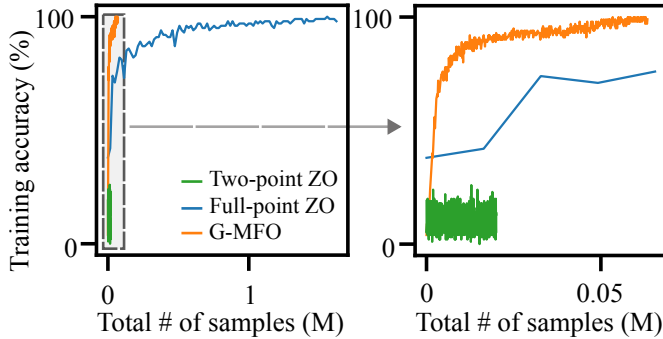


Fig. 5: **G-MFO balances sample efficiency and task performance for the optimization process.** We compare the training curves of the two-point zeroth-order (ZO) method (green), the full-point ZO method (blue), and our G-MFO (orange) method. The full-point ZO method and G-MFO achieve 99% training accuracy on the small dataset. Notably, the total number of samples of G-MFO is only approximately 4% of that required by the full-point ZO method to achieve an accuracy of 99%. Additionally, the two-point ZO method fails to optimize effectively. The right panel presents an enlarged view of the selected region from the left panel.

of the samples to achieve an accuracy of 99% compared to

	Accuracy			# of samples per epoch	Training time per epoch
	Train	Val	Test		
Full-point [43], [51]	99.0%	75.0%	59.0%	128×128	113 min
Two-point [44]	22.0%	27.0%	26.0%	2	2 s
G-MFO (Ours)	99.0%	73.0%	60.0%	128	2 min

TABLE 2: **Numerical comparison of G-MFO and other zeroth-order (ZO) optimization methods on FMNIST dataset on a simulated single-layer diffractive optical computing system.** We show results from the full-point ZO method, the two-point ZO method, and G-MFO.

the full-point optimization method.

4.1.2 Simulation results on the two-layer optical computing system $\hat{f}_{sys-2-layer}$.

We evaluate our method's accuracy by conducting performance tests on a two-layer optical computing simulator $\hat{f}_{sys-2-layer}$ utilizing two classical image classification datasets: MNIST [66] and FMNIST [67]. Each optical computing layer contains $128 \times 128 = 16,384$ trainable parameter, and the system contains $128 \times 128 \times 2 = 32,768$

	MNIST			FMNIST		
	Train	Val	Test	Train	Val	Test
Ideal	90.0%	89.3%	89.5%	82.2%	81.2%	80.5%
HBT	78.0%	78.7%	78.4%	61.6%	61.4%	59.8%
G-MFO (Ours)	86.2%	86.2%	87.0%	75.4%	75.4%	74.1%

TABLE 3: Numerical performance comparison on MNIST and FMNIST datasets on two-layer diffractive optical computing system.

parameters. Furthermore, to assess the robustness of the G-MFO method against system misalignment, we intentionally introduce a slight misalignment on the positive x' -axis direction, amounting to $40\mu\text{m}$ (equivalent to 5 pixels on SLM1) on the optical computing layer1, and $18.7\mu\text{m}$ (corresponding to 5 pixels on SLM2), on the optical computing layer2. We use 10,000 data to train, 10,000 data to validate, and 10,000 data to test.

The simulation outcomes for the MNIST and FMNIST datasets are in Tab. 3. The ideal testing accuracy reaches 89.5% on MNIST and 80.5% on FMNIST. However, the presence of misalignment reduces the accuracy to 78.4% and 59.8% using HBT, a decrease of approximately 10% and 20%. In contrast, the G-MFO method hits an accuracy of 87.0% and 74.1%, effectively mitigating the detrimental impact of system misalignment.

	MNIST			FMNIST		
	Train	Val	Test	Train	Val	Test
Ideal	92.7%	84.3%	82.2%	85.6%	79.9%	76.4%
SBT	81.9%	74.4%	69.3%	68.3%	64.1%	60.9%
HBT	81.9%	75.5%	72.8%	68.3%	68.7%	65.8%
G-MFO (Ours)	83.1%	77.8%	73.6%	74.0%	71.1%	70.4%
HBT+G-MFO (Ours)	87.0%	80.2%	77.7%	75.0%	71.7%	70.2%

TABLE 4: Experimental performance comparison on MNIST and FMNIST datasets on the single-layer optical computing system. Results of the ideal mode are from the simulator, whose parameters are determined from experiments while we impose no misalignment on the simulator. The lower four rows are experimental results from SBT, HBT, G-MFO, and HBT+G-MFO (G-MFO fine-tuning upon the result from HBT). Our method outperforms the SBT and HBT methods on the MNIST and FMNIST datasets.

4.1.3 G-MFO outperforms hybrid training (HBT) experimentally on a single-layer optical computing system $f_{sys-1-layer}$.

We conduct experiments on a real single-layer optical computing system $f_{sys-1-layer}$ on the MNIST and FMNIST datasets. We include in silico SBT and in situ HBT methods utilizing digitally aligned simulators as the comparison baselines. The training, validation, and testing phases each utilize a dataset of 1,000 samples.

Tab. 4 quantitatively shows that our method achieves higher classification accuracy than the HBT and SBT methods on both datasets in the experiments. The SBT method performs poorly due to the reality gap between the simulator and the real system. The HBT method suffers from the bias between \hat{f}_{sys} and f_{sys} in the backward process, while the G-MFO bypasses the bias-sensitive modeling and updates gradients solely with f_{sys} . We further show G-MFO's capability to fine-tune the HBT result. The last row of Tab. 4 shows that the unbiased G-MFO method further improves the results of HBT. Moreover, we also empirically find that fine-tuning outperforms G-MFO only. Fig. 4 visualizes some experimental outputs and confusion matrices using the G-MFO method.

4.1.4 Experimental results on the two-layer optical computing system $f_{sys-2-layer}$.

We also perform experiments on training a real two-layer optical computing system $f_{sys-2-layer}$ on the MNIST dataset. For this experiment, we utilize a dataset comprising 10,000 instances for training, another 10,000 for validation, and a separate set of 10,000 for testing. The training, validation, and testing accuracy are 80.43%, 81.1%, and 80.3%, respectively.

Compared with the simulated G-MFO outcomes presented in Tab. 3, the experimental accuracy is lower. We hypothesize that this discrepancy is attributed, in part, to mechanical disturbances encountered during the multi-day span of the experiment. Moreover, the training accuracy from a two-layer system is lower than that of a single-layer system, while the validation and testing accuracy is higher. This is because we use more samples when training a two-layer system, alleviating overfitting.

4.2 Advantage: memory- and computation-efficient training enabled by G-MFO

In addition to the predicting accuracy discussed in the previous subsection, our G-MFO method has an advantage over other training algorithms regarding GPU time and memory efficiency. The SBT and HBT methods compute $\hat{f}_{sys}(x, w)$ and $\nabla_w \hat{f}_{sys}$ (Fig. 1b) for each input x in silico, which requires a lot of in silico computation resources. In contrast, our G-MFO method executes the calculation of $f_{sys}(x, w)$ in the light-speed real optical computing system, but not in the computation- and memory-heavy simulator $\hat{f}_{sys}(x, w)$. The only step of our G-MFO method that consumes in silico computational resources is the one described in Eq. 9, where we calculate the score gradient. The in silico computational resources consumption in this step is low because it only scales with the dimension of w and has no relation to the complexity of the system's light transport \hat{f}_{sys} .

We compare our G-MFO method with HBT in GPU memory and time usage during experiments on a single-layer optical computing system in Fig. 7. Our G-MFO method requires far less GPU time and memory than the HBT method during the training.

4.3 Limitation: G-MFO exhibits the curse of dimensionality

Our method is not without its limits. Our G-MFO training relies on Monte Carlo integration. It thus inherits the *curse of*

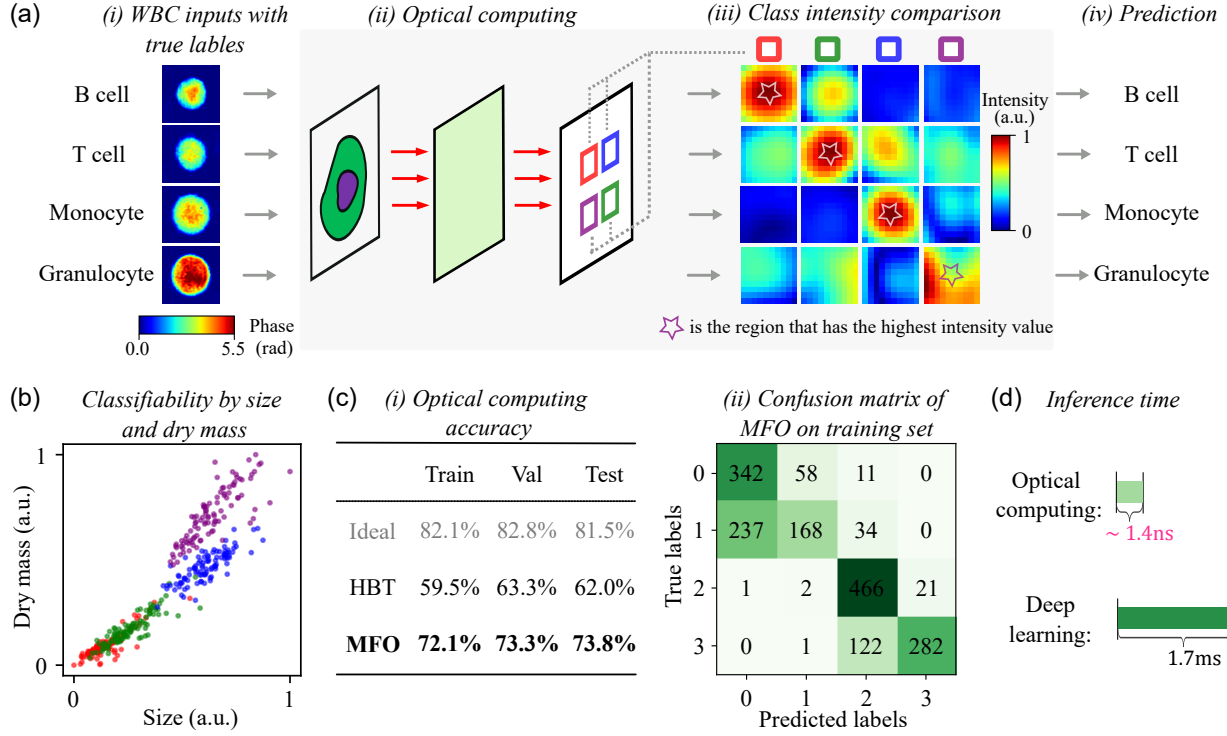


Fig. 6: **All optical marker-free cell classification enabled by G-MFO training.** (a) Our trained optical computing system classifies four WBC subtypes, including B cell (red), T cell (green), monocyte (blue), and granulocyte (purple), in a marker-free manner. The system uses the (i) marker-free phase information of the WBCs to (ii) perform optical classification. The output plane of the system has different regions corresponding to different classes. We (iii) measure the intensity values in these regions and (iv) choose the class with the highest intensity as the prediction. (b) shows that non-morphological features, including size and dry mass, cannot divide the four subtypes well. In contrast, (c)(i) shows the optical computing system trained with our G-MFO method can classify the WBC subtypes with a training/validation/testing accuracy of 72.1%/73.3%/73.8%, which is higher than that of the HBT method. (c)(ii) shows the confusion matrix of G-MFO results on the training set. (d) The optical computing system produces its output at the speed of light (1.4 ns). This is faster than the electronic computing with ResNet10 [68], [69], which has an inference time of 1.7 ms.

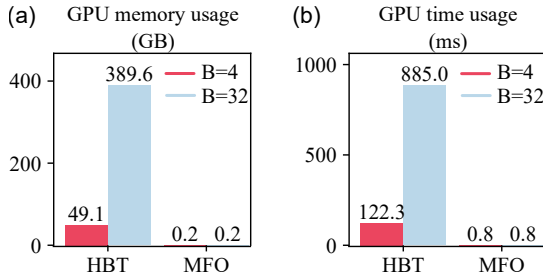


Fig. 7: **Advantage: G-MFO saves in silico computing resources during training.** We compare the GPU (a) memory and (b) time consumption of HBT and G-MFO methods for training one batch of inputs with batch sizes of $B = 4$ and $B = 32$. G-MFO is more efficient than HBT during the in situ training process in memory and time. Moreover, G-MFO's resource consumption does not increase as the batch size increases, unlike HBT.

dimensionality from the Monte Carlo integration [70]. That is, the number of samples M needed to estimate the integration in Eq. 7 with a given level of accuracy grows exponentially concerning the H , the number of trainable parameters (i.e.,

dimensionality) of the function. This is also discussed and alleviated with *variance reduction* in the previous Sec. 3.3. However, the G-MFO strategy presented in this paper is still sample-inefficient, though unbiased and memory-efficient. We need to either limit the number of trainable parameters H , or sample a large number of varied optical computing weights $\{w\}_{j=1}^M$ from distribution $p(w|\theta)$ in every iteration to make G-MFO's gradient less noisy. The former limits our method's design space, while the latter requires more executions on the real system, which prolongs the training time.

We quantitatively investigate the influence of this limitation in Fig. 8 through simulations on a one-layer optical computing system $\hat{f}_{sys-1-layer}$, employing 1,000 training samples from the MNIST dataset. The figure demonstrates how M and H impact the training performance of G-MFO. Shown in Fig. 8(a), G-MFO requires a $M \geq 128$ to achieve a training accuracy of $> 90\%$ given $H = 128^2$. Moreover, in Fig. 8(b), our G-MFO method fails when increasing H beyond 128^2 while keeping the sampling size M fixed to 128.

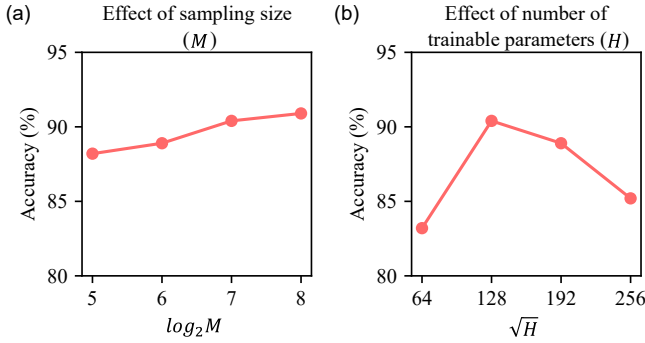


Fig. 8: **Limitation: G-MFO exhibits the curse of dimensionality.** (a) When the sampling size of optical computing weights M rises from $2^5 = 32$ to $2^7 = 128$, the training accuracy increases by 2.2%. However, increasing M from $2^7 = 128$ to $2^9 = 256$ only improves 0.5%. Here, we fix the H as 128^2 . (b) G-MFO achieves the highest accuracy when $H = 128^2$ under a sampling size $M = 128$ while further enlarging H fails the result.

4.4 Application: all-optical classification on marker-free cellular dataset

For the first time, we demonstrate the capability of an optical computing system for marker-free cell analysis, trained by our G-MFO algorithm (Fig. 6). We work on the white blood cells (WBC), whose abnormal subtype percentages indicate the immune system's malfunction or infectious disease [71], [72]. We include details of the WBC phase map dataset in Supplement Sec. S.2. Previously, researchers used machine learning methods to classify WBC subtypes, including monocyte, granulocyte, B cell, and T cell, by their morphology in a marker-free way [68], [73]. However, the analysis process is computationally heavy and time-consuming.

Here, we accelerate the marker-free cell analysis process via computing with light. Our G-MFO method strikes a training/validation/testing classification accuracy of 72.1%/73.3%/73.8% when classifying 4 types of WBC using a one-layer optical computing system, exceeding that of the HBT method (Fig. 6c). Furthermore, Fig. 6d shows that the inference enabled by optical computing is almost instantaneous ($\frac{d_{IO} + d_{OC}}{c} = 1.4 \text{ ns}$, where c is the speed of light), compared to the 1.7 ms of ResNet10, the electronic machine learning model used in [68]. We need 1 more milliseconds of in silico computing of region intensities corresponding to different classes to obtain the prediction. Such a step can be skipped if we use single-photon avalanche diode (SPAD) [74] point detectors to count the corresponding regions' cumulative signals. Though for now, the performance of our single-layer linear optical computing system is not on par with the electronic neural network, which hits a testing classification accuracy of 90.5% [68], the potential of having ultra-high inference speed and $> 70\%$ classification accuracy here point out an exciting direction on further increasing the complexity of our optical computing system to improve the absolute classification accuracy on classifying cells.

5 DISCUSSION

5.1 Future directions

5.1.1 Further apply G-MFO to more complex optical computing systems.

Exploring the scalability of the optical computing system to more complex optical structures, along with integrating more layers and non-linear activation functions, could potentially enhance absolute performance.

5.1.2 Further improve the performance of G-MFO.

Future research could also consider employing more advanced techniques related to Monte Carlo integration to reduce the training variance discussed in the previous Subsec. 4.3, which we anticipate could substantially broaden the viable search space, thus further empowering the G-MFO approach. These include using more advanced sampling strategies [75] or integrating G-MFO with the SBT methods [48] or adding critic function [76]. The latter two methods trade off the introduced bias and sampling variance.

5.1.3 Expanding the application of G-MFO to additional computational optics tasks.

The concept of G-MFO presents promising avenues for application in other areas of computational optics, such as computer-generated holography [52] and lens design [77]. The inherent model-free and resource-efficient characteristics of G-MFO position it as a viable alternative to prevalent model-based methods [77], [78]. Future research could focus on leveraging G-MFO to these domains, potentially enhancing computational efficiency and performance.

5.2 Conclusion

To conclude, our study underscores the effectiveness of a model-free strategy in training optical computing systems in situ, manifesting considerable potential in computational efficiency and reducing the simulation-to-reality performance gap. Although the study does not focus entirely on absolute image classification accuracy as it is based on a simple single or double diffractive optical computing system without non-linearity, it shows relative improvements compared to the existing training strategies, indicating that our strategy is a potentially valuable approach. The model-agnostic nature of our technique may become even more beneficial when implemented in intricate optical systems, representing a robust and versatile alternative to current strategies. It promises a strong foundation for exploring and practically implementing optical computing in real-world applications such as high-speed marker-free cell analysis.

ACKNOWLEDGMENTS

We thank Dr. Cheng Zheng from MIT for the discussions in the early stages of the work.

Author Contributions. G.Z. conceived the project, derived the formulation, and built the backbone code and system. X.S. helped with the concept development, code writing and system setup and collected the simulation and experiment results. R.Z. supervised the project, acquired the funding

support for the project and proposed the application for label-free real-time cell analysis. G.Z. and X.S. wrote the manuscript with comments and edits from R.Z..

REFERENCES

- [1] A. Maréchal and P. Croce, "Un filtre de fréquences spatiales pour lamélioration du contraste des images optiques," *Comptes rendus de l'Académie des Sciences*, vol. 237, no. 12, pp. 607–609, 1953.
- [2] L. Cutrona, E. Leith, C. Palermo, and L. Porcello, "Optical data processing and filtering systems," *IRE Transactions on Information Theory*, vol. 6, no. 3, pp. 386–400, 1960.
- [3] E. O'Neill, "Spatial filtering in optics," *IRE Transactions on Information Theory*, vol. 2, no. 2, pp. 56–65, 1956.
- [4] P. Ambbs, "Optical computing: A 60-year adventure," *Advances in Optical Technologies*, 2010.
- [5] G. Wetzstein, A. Ozcan, S. Gigan, S. Fan, D. Englund, M. Soljačić, C. Denz, D. A. Miller, and D. Psaltis, "Inference in artificial intelligence with deep optics and photonics," *Nature*, vol. 588, no. 7836, pp. 39–47, 2020.
- [6] A. V. Lugt, "Signal detection by complex spatial filtering," *IEEE Transactions on Information Theory*, vol. 10, no. 2, pp. 139–145, 1964.
- [7] J. Zhou, H. Qian, C.-F. Chen, J. Zhao, G. Li, Q. Wu, H. Luo, S. Wen, and Z. Liu, "Optical edge detection based on high-efficiency dielectric metasurface," *Proceedings of the National Academy of Sciences*, vol. 116, no. 23, pp. 11 137–11 140, 2019.
- [8] J. Chang, V. Sitzmann, X. Dun, W. Heidrich, and G. Wetzstein, "Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification," *Scientific Reports*, vol. 8, no. 1, p. 12324, 2018.
- [9] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," *Science*, vol. 361, no. 6406, pp. 1004–1008, 2018.
- [10] M. Rafayelyan, J. Dong, Y. Tan, F. Krzakala, and S. Gigan, "Large-scale optical reservoir computing for spatiotemporal chaotic systems prediction," *Physical Review X*, vol. 10, no. 4, p. 041037, 2020.
- [11] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Network*, vol. 20, no. 3, pp. 391–403, 2007.
- [12] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, vol. 11, no. 7, pp. 441–446, 2017.
- [13] S. M. Buckley, A. N. Tait, A. N. McCaughan, and B. J. Shastri, "Photonic online learning: a perspective," *Nanophotonics*, vol. 12, no. 5, pp. 833–845, 2023.
- [14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [15] L. G. Wright, T. Onodera, M. M. Stein, T. Wang, D. T. Schachter, Z. Hu, and P. L. McMahon, "Deep physical neural networks trained with backpropagation," *Nature*, vol. 601, no. 7894, pp. 549–555, 2022.
- [16] J. Spall, X. Guo, and A. I. Lvovsky, "Hybrid training of optical neural networks," *Optica*, vol. 9, no. 7, pp. 803–811, 2022.
- [17] A. Sludds, S. Bandyopadhyay, Z. Chen, Z. Zhong, J. Cochrane, L. Bernstein, D. Bunandar, P. B. Dixon, S. A. Hamilton, M. Streshinsky *et al.*, "Delocalized photonic deep learning on the internet's edge," *Science*, vol. 378, no. 6617, pp. 270–276, 2022.
- [18] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 949–980, 2014.
- [19] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Reinforcement Learning*, vol. 8, pp. 229–256, 1992.
- [20] T. Zhou, X. Lin, J. Wu, Y. Chen, H. Xie, Y. Li, J. Fan, H. Wu, L. Fang, and Q. Dai, "Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit," *Nature Photonics*, vol. 15, no. 5, pp. 367–373, 2021.
- [21] Z. Zheng, Z. Duan, H. Chen, R. Yang, S. Gao, H. Zhang, H. Xiong, and X. Lin, "Dual adaptive training of photonic neural networks," *Nature Machine Intelligence*, vol. 5, no. 10, pp. 1119–1129, 2023.
- [22] Y. Huo, H. Bao, Y. Peng, C. Gao, W. Hua, Q. Yang, H. Li, R. Wang, and S.-E. Yoon, "Optical neural network via loose neuron array and functional learning," *Nature Communications*, vol. 14, no. 1, p. 2535, 2023.
- [23] T. Zhou, L. Fang, T. Yan, J. Wu, Y. Li, J. Fan, H. Wu, X. Lin, and Q. Dai, "In situ optical backpropagation training of diffractive optical neural networks," *Photonics Research*, vol. 8, no. 6, pp. 940–953, Jun 2020.
- [24] M. Nakajima, K. Inoue, K. Tanaka, Y. Kuniyoshi, T. Hashimoto, and K. Nakajima, "Physical deep learning with biologically inspired training method: gradient-free approach for physical hardware," *Nature Communications*, vol. 13, no. 1, p. 7847, 2022.
- [25] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, "Training of photonic neural networks through in situ backpropagation and gradient measurement," *Optica*, vol. 5, no. 7, pp. 864–871, 2018.
- [26] A. Momeni, B. Rahmani, M. Malléjac, P. Del Hougne, and R. Fleury, "Backpropagation-free training of deep physical neural networks," *Science*, vol. 382, no. 6676, pp. 1297–1303, 2023.
- [27] I. Oguz, J. Ke, Q. Weng, F. Yang, M. Yildirim, N. U. Dinc, J.-L. Hsieh, C. Moser, and D. Psaltis, "Forward-forward training of an optical neural network," *Optics Letters*, vol. 48, no. 20, pp. 5249–5252, 2023.
- [28] J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner, "Reinforcement learning in a large-scale photonic recurrent neural network," *Optica*, vol. 5, no. 6, pp. 756–760, 2018.
- [29] C. Liu, Q. Ma, Z. J. Luo, Q. R. Hong, Q. Xiao, H. C. Zhang, L. Miao, W. M. Yu, Q. Cheng, L. Li *et al.*, "A programmable diffractive deep neural network based on a digital-coding metasurface array," *Nature Electronics*, vol. 5, no. 2, pp. 113–122, 2022.
- [30] H. Zhang, J. Thompson, M. Gu, X. D. Jiang, H. Cai, P. Y. Liu, Y. Shi, Y. Zhang, M. F. Karim, G. Q. Lo *et al.*, "Efficient on-chip training of optical neural networks using genetic algorithm," *ACS Photonics*, vol. 8, no. 6, pp. 1662–1672, 2021.
- [31] P. Chakravarthula, E. Tseng, T. Srivastava, H. Fuchs, and F. Heide, "Learned hardware-in-the-loop phase retrieval for holographic near-eye displays," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–18, 2020.
- [32] Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, "Neural holography with camera-in-the-loop training," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–14, 2020.
- [33] E. Tseng, Y. Zhang, L. Jebe, X. Zhang, Z. Xia, Y. Fan, F. Heide, and J. Chen, "Neural photo-finishing," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–15, 2022.
- [34] C. Zheng, G. Zhao, and P. So, "Close the design-to-manufacturing gap in computational optics with a 'real2sim' learned two-photon neural lithography simulator," in *SIGGRAPH Asia 2023 Conference Papers*, 2023, pp. 1–9.
- [35] W.-P. Liao, H.-L. Liu, Y.-F. Lin, S.-S. Su, Y.-T. Chen, G.-B. Lin, T.-C. Tseng, T.-K. Lin, C.-C. Chen, W.-H. Huang *et al.*, "I-line photolithographic metalenses enabled by distributed optical proximity correction with a deep-learning model," *Optics Express*, vol. 30, no. 12, pp. 21 184–21 194, 2022.
- [36] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.
- [37] G. Hinton, "The forward-forward algorithm: Some preliminary investigations," *arXiv preprint arXiv:2212.13345*, 2022.
- [38] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte carlo gradient estimation in machine learning," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5183–5244, 2020.
- [39] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 43–54, 2020.
- [40] M. Fischer and T. Ritschel, "Zero grads ever given: Learning local surrogate losses for non-differentiable graphics," *arXiv preprint arXiv:2308.05739*, 2023.
- [41] T. Pang, H. T. Suh, L. Yang, and R. Tedrake, "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models," *IEEE Transactions on Robotics*, 2023.
- [42] H. J. T. Suh, T. Pang, and R. Tedrake, "Bundled gradients through contact via randomized smoothing," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4000–4007, 2022.
- [43] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, pp. 462–466, 1952.
- [44] J. C. Spall, "A stochastic approximation technique for generating maximum likelihood parameter estimates," in *1987 American Control Conference*. IEEE, 1987, pp. 1161–1167.

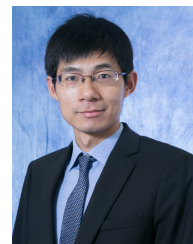
- [45] J. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.
- [46] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—a comprehensive introduction," *Natural Computing*, vol. 1, pp. 3–52, 2002.
- [47] D. Sanz-Alonso and O. Al-Ghathas, "A first course in monte carlo methods," 2024.
- [48] V. Kurenkov and B. Maksudov, "Guiding evolutionary strategies by differentiable robot simulators," *arXiv preprint arXiv:2110.00438*, 2021.
- [49] J. Lehman, J. Chen, J. Clune, and K. O. Stanley, "Es is more than just a traditional finite-difference approximator," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 450–457.
- [50] Y. Nagata, "The lens design using the cma-es algorithm," in *Genetic and Evolutionary Computation Conference*. Springer, 2004, pp. 1189–1200.
- [51] W. Chen, P. Mirdehghan, S. Fidler, and K. N. Kutulakos, "Auto-tuning structured light by optical stochastic gradient descent," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5970–5980.
- [52] G. Zhao and R. Zhou, "Model-free computer-generated holography," in *TENCON 2022-2022 IEEE Region 10 Conference (TENCON)*. IEEE, 2022, pp. 1–3.
- [53] U. Teğin, M. Yıldırım, İ. Oğuz, C. Moser, and D. Psaltis, "Scalable optical learning operator," *Nature Computational Science*, vol. 1, no. 8, pp. 542–549, 2021.
- [54] T. Zhou, X. Lin, J. Wu, Y. Chen, H. Xie, Y. Li, J. Fan, H. Wu, L. Fang, and Q. Dai, "Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit," *Nature Photonics*, vol. 15, no. 5, pp. 367–373, 2021.
- [55] D. Mengü, M. Veli, Y. Rivenson, and A. Ozcan, "Classification and reconstruction of spatially overlapping phase images using diffractive optical networks," *Scientific Reports*, vol. 12, no. 1, p. 8446, 2022.
- [56] Ç. Işıl, T. Gan, F. O. Ardic, K. Montesoglu, J. Digani, H. Karaca, H. Chen, J. Li, D. Mengü, M. Jarrahi *et al.*, "All-optical image denoising using a diffractive visual processor," *Light: Science & Applications*, vol. 13, no. 1, p. 43, 2024.
- [57] M. Park and Y. Park, "Metasurface-based imagers enabled arbitrary optical convolution processing," *Light: Science & Applications*, vol. 11, no. 1, p. 98, 2022.
- [58] J. Li, T. Gan, Y. Zhao, B. Bai, C.-Y. Shen, S. Sun, M. Jarrahi, and A. Ozcan, "Unidirectional imaging using deep learning–designed materials," *Science Advances*, vol. 9, no. 17, p. eadg1505, 2023.
- [59] T. Wang, M. M. Sohoni, L. G. Wright, M. M. Stein, S.-Y. Ma, T. Onodera, M. G. Anderson, and P. L. McMahon, "Image sensing with multilayer nonlinear optical neural networks," *Nature Photonics*, vol. 17, no. 5, pp. 408–415, 2023.
- [60] G. W. Brier *et al.*, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.
- [61] J. Schulman, N. Heess, T. Weber, and P. Abbeel, "Gradient estimation using stochastic computation graphs," *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [62] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [63] J. Mei, W. Chung, V. Thomas, B. Dai, C. Szepesvari, and D. Schuurmans, "The role of baselines in policy gradient optimization," *arXiv preprint arXiv:2301.06276*, 2023.
- [64] D. Zhang, D. Xu, Y. Li, Y. Luo, J. Hu, J. Zhou, Y. Zhang, B. Zhou, P. Wang, X. Li *et al.*, "Broadband nonlinear modulation of incoherent light using a transparent optoelectronic neuron array," *Nature Communications*, vol. 15, no. 1, p. 2433, 2024.
- [65] O. Kulce, D. Mengü, Y. Rivenson, and A. Ozcan, "All-optical synthesis of an arbitrary linear transformation using diffractive surfaces," *Light: Science & Applications*, vol. 10, no. 1, p. 196, 2021.
- [66] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [67] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
- [68] X. Shu, S. Sansare, D. Jin, X. Zeng, K.-Y. Tong, R. Pandey, and R. Zhou, "Artificial-intelligence-enabled reagent-free imaging hematology analyzer," *Advanced Intelligent Systems*, vol. 3, no. 8, p. 2000277, 2021.
- [69] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 630–645.
- [70] R. Bellman and R. Kalaba, "On adaptive control processes," *IRE Transactions on Automatic Control*, vol. 4, no. 2, pp. 1–9, 1959.
- [71] B. J. Bain, *Blood cell morphology in health and disease*. Elsevier, 2017, pp. 61–92.
- [72] G. DOnofrio and G. Zini, *Blood and Bone Marrow Cells*, 2015, pp. 1–87.
- [73] M. Nassar, M. Doan, A. Filby, O. Wolkenhauer, D. K. Fogg, J. Piasecka, C. A. Thornton, A. E. Carpenter, H. D. Summers, P. Rees, and H. Hennig, "Label-free identification of white blood cells using machine learning," *Cytometry Part A*, vol. 95, no. 8, pp. 836–842, 2019.
- [74] C. Bruschini, H. Homulle, I. M. Antolovic, S. Burri, and E. Charbon, "Single-photon avalanche diode imagers in biophotonics: review and outlook," *Light: Science & Applications*, vol. 8, no. 1, p. 87, 2019.
- [75] R. E. Caflisch, "Monte carlo and quasi-monte carlo methods," *Acta Numerica*, vol. 7, pp. 1–49, 1998.
- [76] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in Neural Information Processing Systems*, vol. 12, 1999.
- [77] V. Sitzmann, S. Diamond, Y. Peng, X. Dun, S. Boyd, W. Heidrich, F. Heide, and G. Wetzstein, "End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–13, 2018.
- [78] D. Blinder, T. Birnbaum, T. Ito, and T. Shimobaba, "The state-of-the-art in computer generated holography for 3d display," *Light: Advanced Manufacturing*, vol. 3, no. 3, pp. 572–600, 2022.



Guangyuan Zhao received a BSc degree from Sichuan University and a Master's degree in Optical Engineering from Zhejiang University. He is working towards a PhD degree at the Chinese University of Hong Kong. His research interest lies in the intersection of perception, robotics, and learning. His research aim is to create agents to interactively and informatively sense the world and learn from the sensed data to digitalize the real-world system better and make reliable decisions.



Xin Shu is currently a postdoctoral fellow at the Chinese University of Hong Kong, where she received a PhD degree from the Chinese University of Hong Kong in 2023. Her research interests include computational optics, machine intelligence, and their applications. She received the Hong Kong PhD Fellowship in 2018.



Renjie Zhou is an Associate Professor in the Department of Biomedical Engineering at the Chinese University of Hong Kong. He received a PhD degree in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2014 and undertook postdoctoral training at MIT between 2014–2017. His research interest is in developing optical precision instruments for various research and industrial applications. He has published over 100 journal and conference papers and filed over 10 US/China patents. He is currently serving on the editorial boards of JOSA A, IEEE Photonics Technology Letters, and International Journal of Extreme Manufacturing. He received the Croucher Innovation Awards in 2019 from the Croucher Foundation.