OPTICA

Experimental evaluation of digitally verifiable photonic computing for blockchain and cryptocurrency

SUNIL PAI,[1,2,*] TAEWON PARK,[1] MARSHALL BALL,[3,4] BOGDAN PENKOVSKY,[4] MICHAEL DUBROVSKY,[4] NATHNAEL ABEBE,[1,5] MAZIYAR MILANIZADEH,[6] FRANCESCO MORICHETTI,[6] ANDREA MELLONI,[6] SHANHUI FAN,[1] OLAV SOLGAARD,[1] AND DAVID A. B. MILLER[1]

[1] *Department of Electrical Engineering, Stanford University, Stanford, California 94305, USA*
[2] *Current address: PsiQuantum, Palo Alto, California 94304, USA*
[3] *Courant Institute, New York University, New York, New York 10012, USA*
[4] *PoWx, Cambridge, Massachusetts 02144, USA*
[5] *Current address: Google, Mountain View, California 94043, USA*
[6] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy*
**sunilpai@stanford.edu*

Received 28 September 2022; revised 11 March 2023; accepted 17 March 2023; published 27 April 2023

As blockchain technology and cryptocurrency become increasingly mainstream, photonic computing has emerged as an efficient hardware platform that reduces ever-increasing energy costs required to verify transactions in decentralized cryptonetworks. To reduce sensitivity of these verifications to photonic hardware error, we propose and experimentally demonstrate a cryptographic scheme, LightHash, that implements robust, low-bit precision matrix multiplication in programmable silicon photonic networks. We demonstrate an error mitigation scheme to reduce error by averaging computation across circuits, and simulate energy-efficiency-error trade-offs for large circuit sizes. We conclude that our error-resistant and efficient hardware solution can potentially generate a new market for decentralized photonic blockchain.

https://doi.org/10.1364/OPTICA.476173

1. INTRODUCTION

Photonic integrated circuits (PICs) consisting of networks or "meshes" of Mach–Zehnder interferometers (MZIs) [1,2] are typically proposed as time- and energy-efficient matrix multiplication accelerators for analog domain applications such as quantum computing [3,4], sensing, telecommunications [5], and machine learning [6]. Since photonic meshes can be designed and mass-produced using well-established silicon foundry processes, there has recently been increased effort to commercialize the technology for analog domains that do not necessarily require high accuracy for high performance (e.g., machine learning). Such a computing approach has been previously explored in mixed-signal deep neural network inference designs that favor low-bit resolution up to thermal noise limits [7]. In this work, we implement a computing approach in photonics extending applications of photonic meshes from the continuous analog domains of sensing and quantum computing to discrete digital domains of cryptography and blockchain technology at low-bit precision [2]. To that end, we design photonic matrix multiplication hardware under more stringent numerical accuracy requirements requiring "near-perfect" digital computation.

As our core application, we explore "photonic blockchain" technology, which implements "optical proof of work" (oPoW) [8], proof that this computational work has been performed in the optical domain. In general, oPoW is designed to favor optical computation over digital alternatives; HeavyHash (the first proposal for oPoW [8]) is currently implemented on two live networks (oBTC and Kaspa [9]) and is under consideration for the popular cryptocurrency Bitcoin [10] as part of Bitcoin Improvement Proposal (BIP) 52 [11].

Transitioning Bitcoin to oPoW would require an update to the Bitcoin codebase that now uses the SHA-256 hash in its proof of work (PoW) algorithm [10]. Equipped by optical computation with sufficient accuracy, such protocols can leverage energy-efficient computation and state-of-the-art photonic hardware (such as photonic meshes) to verify cryptocurrency transactions and ultimately other wide ranging applications of blockchain such as medical data, smart contracts, voting, logistics and tracking, spam filters, and protection from distributed denial-of-service (DDoS) attacks [12,13]. In situations where energy cost is a bottleneck, blockchain technologies that use oPoW inherently incentivize using photonic hardware over other alternatives to gain competitive advantages in compute efficiency and further security against malicious actors such as malware or attack vectors [8]. For example, at the time of writing, cryptocurrency mining accounts for as much energy as many countries (e.g., exceeding all of Sweden, the 27th-most energy-consuming country in a 2021 energy estimate [14]), and this energy consumption will increase, by design, as more value is stored in decentralized PoW blockchains. Energy cost concerns have contributed to recent crashes in the cryptocurrency marketplace, including over the span of two weeks in May 2021 and again in January 2022 that reduced market capitalization of popular cryptocurrencies by the equivalent of more than 1 trillion dollars [15]. Photonic blockchain could thus serve as a timely application for a PoW-based cryptocurrency that incentivizes energy-efficient photonic hardware, which can furthermore prove to be an appealing option for other blockchain-based applications.

While our primary emphasis is implementation and performance of oPoW, we must first clarify the energy-efficiency problem in blockchain and cryptocurrency. Cryptocurrency is a decentralized currency market where transactions (e.g., "Alice gives Bob one bitcoin") are stored in a chain of blocks ("blockchain"). To earn a share of the market, a cryptocurrency miner can "mine" (add a new block of transactions) to the blockchain using computational PoW where the computer can solve a puzzle for a payout reward. This puzzle consists of generating a 256-element bitvector (vector of 1's and 0's) by feeding digital block transaction data through a cryptographic hash function H, such as SHA-256 (which converts any digitally encoded data into 256-bit numbers and which is infeasible to invert), which is the energy-intensive "computational work." [Such hash functions, "one-way" (non-invertible) functions for private-key cryptography, are more generally used to securely encrypt and decrypt data for various secure applications beyond blockchain.] This function is called twice (once on the original block data and again on the result of the first call) through a scheme similar to "Hashcash" [13] while adjusting a *nonce* (32-bit pseudorandom number) in the block until the first B bits in the bitvector are 0, which proves that sufficient computational work has been done and adds the block to the blockchain. The parameter B is a tunable difficulty parameter that is increased as the coin (which in many cases has limited supply) is more scarce, and the expected number of cycles before the puzzle is solved is 2^B. Crucially, cryptocurrency mining comes at an energy cost proportional to the number of hash function evaluations before a block is mined and transactions in the block are verified.

In oPoW, the miner is incentivized to reduce mining costs by choosing a hash function H such that some choice of optical (photonic) hardware improves the energy efficiency and speed of computation compared to digital alternatives for H. The key idea here is not to choose an H that outperforms an existing hash function e.g., Bitcoin [10]. Rather, as argued in BIP 52 [11] and Ref. [8], we intentionally choose some feasible H for which optical hardware is more energy efficient compared to any digital hardware [8] to evaluate a specific H. The overall goal is to shift the overall mining budget of a given cryptocurrency from operating expense (energy) to capital expense (hardware) by incentivizing more complex and costly, but also more energy-efficient, hardware. This energy efficiency arises due to limited hardware resources and high capital expense, reducing the number of H evaluations required to operate the blockchain. Reduced energy consumption is possible even when computational energy per H evaluation exceeds that of Bitcoin [8], likely required to ensure the same security guarantee as Bitcoin.

However, the great challenge of such computing is that, for a well-designed cryptographic hash, any error in the bits output by analog hardware renders an entire hash verification invalid; this necessitates some strict design criteria and possibly some error mitigation, which we explore in this paper. We address this accuracy problem by numerically and experimentally evaluating a new photonic hash function called "LightHash," a modified hash function from Bitcoin's Hashcash that combines the energy efficiency of low-bit precision photonic matrix–vector multiplication (MVM) with the security assurances of the Bitcoin protocol (i.e., digital hash functions SHA-3 or SHA-256). We define feasible design criteria (e.g., number of photonic inputs and outputs) and propose a hardware-agnostic error mitigation scheme that enables our photonic hash function to outperform any digital alternative. Notably, other non-oPoW approaches promise a low-energy blockchain security algorithm (e.g., proof of stake [16]); however, they present new unexplored security risks and alter the basic game theory underlying systems such as Bitcoin [8].

2. PHOTONIC BLOCKCHAIN

Photonic blockchain can be defined as any blockchain technology incorporating a photonic link and/or computational element aimed at improving the energy efficiency required to add blocks to the blockchain. In optical cryptography, a cryptographer encrypts or decrypts a message by sending the message bits through a hash function, where at least one part of the hash function favors the energy efficiency of optics, which can help shift the market away from centralized corporate entities specializing in digital hardware mining [8]. Here, we propose a class of photonic hash functions ("LightHash") that modifies the Bitcoin scheme and benefits from energy-efficient MVM performed optically [8].

Our implementation, though by no means the only possible photonic implementation of efficient MVM for LightHash [17,18], is built from N-port triangular or rectangular MZI networks [1,19,20]. Meshes operate by repeatedly interfering spatially multiplexed mode vectors of coherent light (over N ports), where modes are represented as complex numbers with amplitude and phase. The constructive and destructive interference can be programmed using electrically controlled phase shifts to implement any unitary transmission matrix $U \in \mathrm{U}(N)$ (satisfying the energy-conserving property $U^\dagger U = I$) [1,19]. After adding a column of "singular value" MZIs followed by a second universal network as in Ref. [1], it is possible to compute an arbitrary linear operator based on the singular value decomposition (SVD) of any matrix Q [1]. The resulting photonic processors can be programmed to implement arbitrary linear operations in an energy-efficient manner; though energy must be spent in generating, modulating, and detecting the optical signals, the actual matrix–vector product is performed by passive linear optical transformations without additional power. The rest of the computations in photonic hash functions include logical operations on bits that are best implemented in the digital domain (e.g., SHA-256 is efficiently implemented on digital processors) and ultimately provide the necessary provably secure protection. By co-integrating this digital functionality with photonic meshes in a systematic manner, we leverage the unique benefits of optics (linear computation) and electronics (nonlinear computation and logic) for a fully integrated photonic cryptographic solution. While previous proposals of this scheme exist (e.g., HeavyHash [8]), a protocol that is sufficiently error tolerant and is both time and energy efficient (including analog–digital conversion) is yet to be proposed. Similar challenges are faced in photonic circuits for digital optical telecommunications, and indeed, the mathematics of "bit error rates" (BERs) also can be applied to the problem of optical cryptography. Ultimately, the core challenge is to find a protocol that successfully brings photonic computing, a technology typically used for analog computing, into the digital realm with near perfect accuracy.

To this end, we examine whether meshes can accurately implement matrix multiplication compared to an electronic digital implementation so that they could ultimately be used for PoW cryptography and confer a "photonic advantage." The photonic advantage for our particular scheme follows from the conjecture that, within the LightHash evaluation, photonic hardware performs amortized matrix multiplication by a random block-diagonal Q operator at least an order of magnitude more efficiently than traditional hardware, where an element within the blocks of Q is sampled from uniform distributions over a set of K integers. First, through numerical simulation, we show that programming a block integer matrix Q onto a series of SVD-based photonic architectures [1] (as opposed to purely unitary circuits [19]) and adjusting the numerical precision through different integer K values can minimize the systematic error in analog computation to make it more amenable to optical cryptography. Then, we experimentally evaluate the cryptographic protocol on a physical photonic chip accelerator capable of performing 4×4 unitary matrix–vector products to estimate performance on our new proposed LightHash protocol. Since the LightHash matrix–vector operation is performed in discrete space, we can find conditions such that possible outputs are separated sufficiently far enough to guarantee near-perfect accuracy. The increased energy-efficiency-per-compute of a photonic platform would increase the security of cryptocurrencies and blockchain operations and significantly shift from energy cost (operating expense) to resource cost (capital expense) in cryptocurrency mining [8]. The resulting increased demand for photonic chips could incentivize PIC development and manufacturing by adding new applications.

3. ALGORITHM

As shown in Fig. 1, our photonic cryptocurrency protocol incorporates photonic meshes within Bitcoin's PoW hash computation to implement LightHash.

Our LightHash photonic cryptographic protocol transforms block (transaction) data into a 256-bit "possible solution" to a cryptographic puzzle, and includes a photonic integrated chip computation within the protocol. The protocol begins with the well-known SHA3-256 protocol, which is part of the already-prevalent digital cryptocurrency Bitcoin and converts block data (containing transactions in the marketplace) into a 256-bit vector containing a sequence of 256 0's and 1's. These bit data are directly fed to optical modulators controlling the optical input into the photonic accelerator chip in chunks of N bits with the following protocol.

1. **Input**: the input into the photonic network is a phase-shift keyed bitstream $\boldsymbol{b}_{\text{in}}$ that is represented as inputs of equal magnitude set to either $x_n = \{1, -1\}$ depending on bit value $b_n = \{0, 1\}$, i.e., $\boldsymbol{x} = e^{i\pi \boldsymbol{b}_{\text{in}}}/\sqrt{N}$; this is set by sending digital signals to well-calibrated optical modulators.
2. **Device operator**: as shown in Fig. 1(b) for $N = 4$, the device operator for each block $Q_m = U\Sigma V^\dagger$ with circuit size N consists of two unitary operators U, V of size N, implemented using triangular or rectangular networks [19,20] and a set of N singular values Σ, implemented using MZI node attenuators (with "drop ports") [1]. The elements $Q_{m,ij}$ are randomly sampled to be one of K distinct integers centered symmetrically around 0 and spaced 2 apart. At block creation only, a digital computer is used to find the static phase shifts for meshes implementing U, V, Σ to ultimately program Q onto the chip.
3. **Output**: the output of the device is the complex output vector $\boldsymbol{y} = Q\boldsymbol{x}$ with output power $\boldsymbol{p} = |\boldsymbol{y}|^2$, where $|\cdot|^2$ is an elementwise absolute value-squared operation. A photodetector equipped with a transimpedance element (load resistor or amplifier) converts power to voltage, which is then fed through output comparators corresponding to threshold power $p_{\text{th}} = y_{\text{th}}^2$ to determine output bits $\boldsymbol{b} := H(\boldsymbol{p} - p_{\text{th}})$ (where H is the Heaviside step function). At block creation, selection of p_{th} via simulation guarantees roughly equal probability of a 0 or 1 output bit.

Note that the definition of the threshold amplitude $|y_{\text{th}}|$ should be consistent with the scaling of the blocks in matrix Q. Since the maximum singular value of Q is set to 1 in the physical implementation (no optical gain elements are used in our photonic mesh), the threshold amplitude is also scaled by this factor (Supplement 1, Section 6).

The LightHash function, a relatively simple modification of the Bitcoin protocol, was chosen carefully to allow for a feasible photonic cryptographic protocol. The key insight in LightHash is that spacing possible optical output values in a discrete grid (i.e., using integer math) ultimately enables an error-tolerant threshold and "digital verifiability" of the hash function. This digital optical data encoding is required in the blockchain PoW protocol and may be checked by other digital systems already used throughout the cryptonetwork.

A unique feature of LightHash is the bit resolution K, which can be used to change the range of possible output values. For instance, $K = 2$ means the matrix elements can be either 1 or -1, and $K = 4$ means the options for each matrix element are $(-3, -1, 1, 3)$. Each row vector–vector product in the overall matrix–vector product can actually be thought of as a random walk with K defining possible step sizes (1 for $K = 2$ and (1, 3) for $K = 4$). Since the inputs are either -1 or 1, an increase in K means an increase in the range of possible output values and effectively the number of bits or quantized levels present in the output. Due to a larger required number of output bits, the use of higher K, as with higher N, leads to higher computational efficiency but a more error-prone photonic chip. We center the possible integers in the matrix to zero since LightHash is designed to represent an optical physical random walk in discrete space. We space the integers by 2 instead of 1 to maintain integer step sizes for both odd and even K.

Note that the device is set to implement Q only once per block added to the blockchain, which means that the photonic miner has some time to self-configure itself to implement Q, and block times can generally be several minutes at sufficiently high difficulty [8]. If $N < 256$, we repeat $256/N$ times (assuming N divides 256) to output a total of 256 bits that is "exclusive or'd" (XOR'd) with the original input vector and fed into the second SHA3-256 function as in Fig. 1(b).

Systematic error (e.g., loss, coupling, and phase errors) can be compensated for in various ways using phase shifter calibration

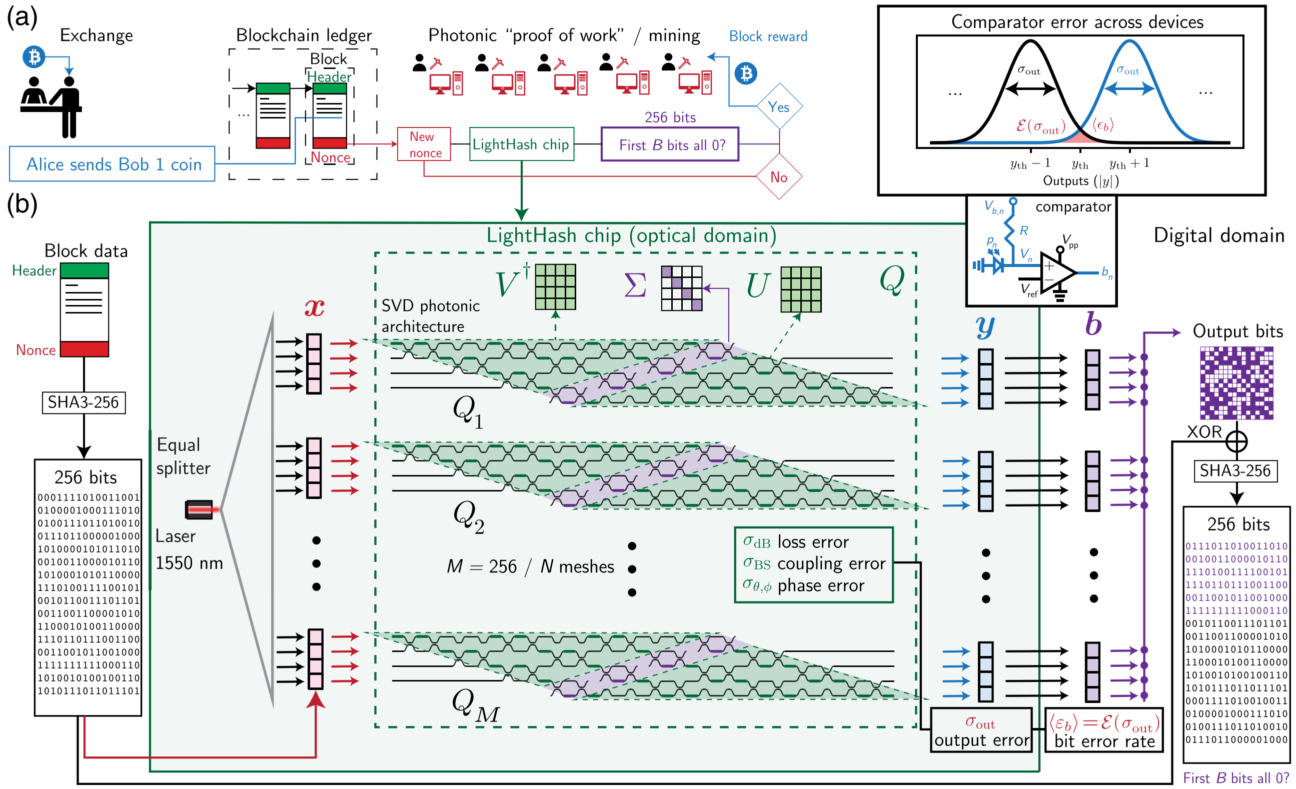

Fig. 1. LightHash chip and protocol. (a) The LightHash optical proof of work protocol, similar to HeavyHash [8], is a slight modification of the Bitcoin protocol, where an arbitrary photonic mesh-based matrix–vector product is inserted in the middle (green). Transactions are verified by photonic miners as in the Bitcoin network, with photonic chips being the ideal technology to achieve a block reward. (b) LightHash conceptual chip footprint consists of a laser, a digital processor to accelerate SHA3-256, modulator and comparator for optoelectronic conversions, and optical processors (green) implementing Q using $256/N$ parallel SVD operations of size N, here depicted for $N=4$. Output bits are ideally measured using comparators (inset) running at GHz speeds [21]. The photonic proof of work error analysis model shows how systematic (loss, coupling, phase) error in the device propagates all the way to an overall hash error rate $\langle\varepsilon\rangle \approx 256\langle\varepsilon_b\rangle$ that arises due to overlap between successive values near the threshold shown in the inset. Reducing the hash error rate is the main aim of this work and is necessary to implement LightHash in practice.

in a photonic mesh; for example, self-configuration [1,22], *in situ* training [23], or off-chip calculation [24] can compensate for phase and coupling errors, but not loss variance errors. Calibration and error mitigation generally occur at the level of individual "unit cell" nodes or MZIs that can be more straightforwardly characterized [24].

To correct this error, we use a simple form of "hardware-agnostic error mitigation" [Supplement 1, Fig. S2(c)] in which the computation is repeated up to R times across R circuit copies. (We note similarities to the port allocation scheme of Ref. [25].) If the expected error is σ_{out}, this can reduce the error to $\sigma_{\text{out}}/\sqrt{R}$, a factor of $\sqrt{R}$ improvement. This repetition may be implemented using R separate devices implemented on the same chip. To save energy, we can use the same number of modulators and split the input signal $\boldsymbol{x}$ across R different meshes implementing the same Q but different error, in a process called "hardware-agnostic error mitigation," as each mesh samples a presumably random systematic error. One potential drawback is that the systematic error may be correlated across the R meshes; one way to address this is to permute the singular values (and basis vectors of U, $V^\dagger$). The number of comparators is the same given the photocurrents from corresponding photodetectors at the R device operator outputs can be grouped into a single current, then passed through a transimpedance amplifier and comparator. Note that if a fixed optical budget is used, the $\sqrt{R}$ improvement assumes photodiode error is sufficiently low compared to systematic error.

The challenge of photonic cryptography that we address is to ensure accuracy across all 256 bits [hash or packet error rate (PER)] while also affording a significant advantage over equivalent digital hardware in speed and energy efficiency (otherwise, there would be less demand for photonic hardware). For simplicity, we may consider all bits to have independent BERs ε_b so the hash error rate is given by $\varepsilon = 1-(1-\varepsilon)^{256} \approx 256\varepsilon_b$. To put this in practical terms, for any given device to have 1% PER, each of the individual bits should have roughly 0.004% BER. This increases the importance of error mitigation in PICs, which is especially challenging in the presence of unbalanced photonic loss. This requires exploring the trade-offs of increasing circuit size N and difficulty K (which increase the difficulty) over the error. The pseudocode for oPoW based on LightHash is in Supplement 1, Section 8, Algs. S1 and 2.

4. SCALING SIMULATIONS

First, we numerically evaluate scaling of energy efficiency, noise error, and systematic error where the triangular meshes are replaced by low-depth rectangular meshes [20] using circuit sizes $N=8$ to 128 for $K=3, 5, 9, 17$ (smallest K requiring digital representation of 2, 3, 4, 5 digital bits, respectively) (Fig. 2). Ideally, the outputs $\boldsymbol{y}$ follow a roughly discretized Gaussian distribution as

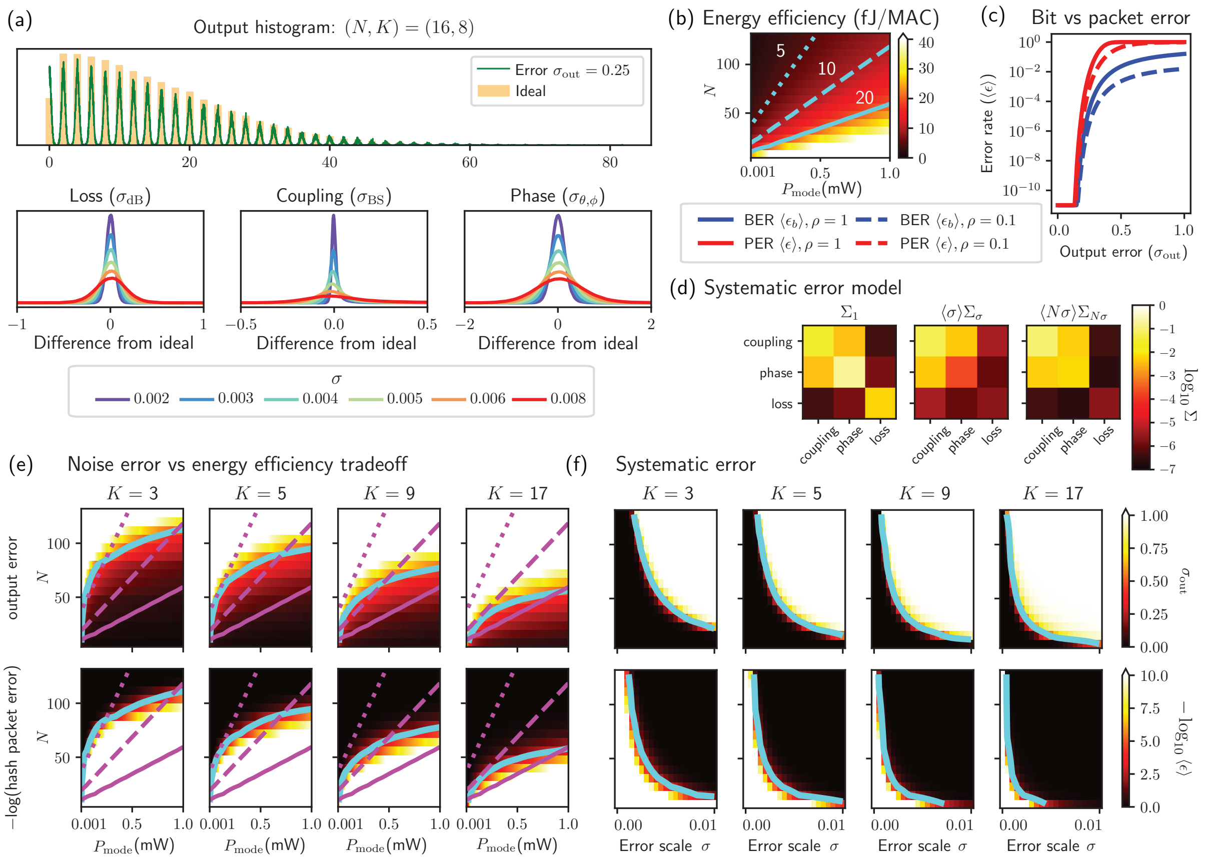

Fig. 2. Simulation of LightHash error-energy trade-off. (a) Example output histogram for $N = 16$ and $K = 8$ exceeds the capabilities of our device ($N = 4$) shown for both ideal and simulated implementations. Overall coupling error, loss error, and phase error contributions for error scale $\sigma < 0.01$ all are roughly Gaussian with coupling errors skewed slightly to the right. (b) Energy efficiency (fJ/MAC). Optical energy scaling relation shows that per-mode optical power ($P_{\mathrm{mode}} = 1$ µW to 1 mW) scales with N, with 5, 10, 20 fJ/MAC labeled contours shown in blue. (c) Bit versus packet error. The error prefactor ρ has a small effect on the bit error scaling; regardless of the scaling of ρ, we find that $\sigma_{\mathrm{out}} = 0.25$ is sufficient to ensure sufficiently low bit error (<1%) for $\rho = 0.1$ to 1. (d) Systematic error mode. Fitted log scale normalized systematic error model [Σ matrices of Eq. (1)], showing high coupling-phase correlation and higher-order contributions to overall output error proportional to σ, $N\sigma$. (e) Noise error versus energy efficiency trade-off. Given N from 8 to 128 and $K = 3, 5, 7, 17$ (respectively requiring 2, 3, 4, 5 bits), we estimate the error versus energy trade-off based (Supplement 1, Sections 4 and 5, Tables S1 and 2), overlaying the energy efficiency contours of (b), now in magenta, to find the intersection of feasible region and estimated photonic advantage given noise estimates of highly sensitive avalanche photodiodes [26]. (f) Systematic error. Bit threshold error profile shows sharp transition in overall hash error as a function of circuit size N and error scale $\sigma = 0$ to 0.01, sampled over 20 random $\boldsymbol{x}$, 10 random Q, and $1280/N$ random error samples scaled by σ and error-weighted by $\boldsymbol{w} = [1, 1, 3]$. Further simulations show the independent contributions of loss, coupling, and phase errors (Supplement 1, Fig. S1).

might be expected by a random walk based on our definitions of Q and $\boldsymbol{x}$. The field magnitudes $|\boldsymbol{y}|$ are more readily measured by output photodetectors and as expected form a discrete half-normal distribution as shown in Fig. 2(a), with a notable dip in histogram values for outcomes of zero.

In practice, we must design LightHash to operate at the limit of the trade-off between errors [Figs. 1 and 2(a)] and energy efficiency, which depends on input optical power (limitation of shot-noise-limited photoreceivers) and systematic fabrication errors giving the problem-scale error σ_{out}. Assuming the error is sufficiently small for some total input optical power P_{mode}, we can reasonably achieve energy efficiencies in fJ/MAC (multiply and accumulate), which can outperform digital platforms [Supplement 1 and Fig. 2(b)]. Compared to digital platforms scaling with MACs (N^2 per matrix multiply), the photonic mesh energy consumption scales with N input/output ports.

Given these errors, Fig. 2(c) indicates that the transition between feasibility and infeasibility occurs sharply when $\sigma_{\mathrm{out}} \approx 0.25$. As in Fig. 1, bit errors arise due to "bit threshold overlap" in the (approximately) Gaussian error distributions between successive values at threshold, given by $\mathcal{E}(\sigma_{\mathrm{out}}) = 0.5\mathrm{erfc}((\sigma_{\mathrm{out}}\sqrt{2})^{-1})$, where erfc denotes the complementary error function. The quadratic exponent in the erfc function's integrand, i.e., $\mathrm{erfc}(z) \propto \int_z^\infty e^{-t^2}\mathrm{d}t$, ultimately results in the sharp transition. To find the corresponding expected bit error, we multiply the overlap in error by twice the probability $\rho(N, K)$ that the values belong to the Gaussian spikes immediately before or after the threshold y_{th}. Assuming $\langle\varepsilon_b\rangle$ is small, we get the expression for expected hash error $\langle\varepsilon\rangle$: $\langle\varepsilon\rangle \approx 256\rho(N, K)\mathcal{E}(\sigma_{\mathrm{out}})$.

There are two major sources of errors contributing to σ_{out}: systematic (discussed here) and photodetector noise (Supplement 1). If the ratio S, given measured power P and Gaussian noise error δP, is defined as $S(P) = \sqrt{\mathrm{Var}[\delta P]}/P$, the contribution to σ_{out} corresponds to $\sigma_{\mathrm{noise}} \approx S(p_{\mathrm{th}})y_{\mathrm{th}}/2$ in the amplitude domain, given $\delta P \ll P$. For our systematic error analysis, caused by drift and fabrication error, we define error scaling σ varying from 0 to 0.01 and define error weights $\boldsymbol{w} = [w_{\theta,\phi}, w_{\mathrm{BS}}, w_{\mathrm{loss}}]$ such that $\boldsymbol{\sigma} = [\sigma_{\theta,\phi}, \sigma_{\mathrm{BS}}, \sigma_{\mathrm{loss}}] = \sigma \boldsymbol{w}$ for loss, phase, and coupling errors, respectively. We find that σ_{out} increases nearly linearly with N, K, σ. However, phase and coupling cross terms appear since MZI coupling errors can be reparametrized as phase errors and vice versa [24]. Ultimately, we achieve $\sim 3\%$ error in predicted σ_{out} compared to simulated data using [Fig. 2(d)]

$$\sigma_{\mathrm{out}}^2 \approx N^2 K^2 \sigma^2 [\boldsymbol{w}^T \Sigma(N, K, \sigma) \boldsymbol{w}] + \sigma_{\mathrm{noise}}^2 (P_{\mathrm{mode}}). \quad (1)$$

Here, the correlation is modeled as $\Sigma = \Sigma_1 + \Sigma_\sigma \sigma + \Sigma_{N\sigma} N\sigma \in \mathbb{R}^{3\times 3}$, where Σ is a symmetric matrix modeled given data sampled at $\boldsymbol{w} = [0, 0, 3], [0, 1, 0], [1, 0, 0], [1, 1, 0], [1, 1, 3]$, $\sigma = 0$ to 0.01. As is evident in Fig. 2(a), error distributions are roughly Gaussian, though our simulations suggest that coupling error σ_{BS} results in a right-skew compared to the more symmetric distributions from loss and phase error. The error increases monotonically with N, K, σ, and positive correlations in phase-coupling are observed [Fig. 2(d)], while the loss is relatively uncorrelated to the other two errors.

In Figs. 2(e) and 2(f), we explore trade-offs between error σ_{out} and PER $\langle \varepsilon \rangle$ or PER given $N, K, \sigma, P_{\mathrm{mode}}$. Due to error-energy trade-offs for optical receiver circuitry (Supplement 1, Tables S1 and 2), energy efficiency of LightHash may be dominated by optical power required in a GHz-scale chip. Large input optical power is required since loss scales exponentially with the circuit depth $2N$, requiring per-mode optical input powers P_{mode} near 1 mW. Photonic meshes have a sharp maximum N separating the feasible and infeasible regions to avoid an explosion of input optical power required for an idealized LightHash SVD chip [cyan curves, Fig. 2(e)]. Critically, the "feasible" and "photonic advantage" regions overlap; advantage is achieved above the dashed magenta curve at 10 fJ/MAC outperforming efficient, equivalent digital implementations requiring 80 fJ/MAC for 3-bit $K = 5$ [27] (Supplement 1, Sections 4 and 5). We additionally explore systematic error scalability in Fig. 2(f) for $\boldsymbol{w} = [1, 1, 3]$; while calibration and self-configuration may sidestep such issues in sufficiently small circuits [1,5], the phase, coupling and loss error scaling notably require $\sigma < 0.1\%$ for LightHash to work.

5. EXPERIMENTAL EVALUATION

Now that we have defined our protocol and simulated the scalability of the technique, we experimentally quantify errors in a 4 × 4 port MZI mesh network (i.e., $N = 4$, also used in Ref. [28]) as a function of bit resolution K using our custom designed chip and the experimental setup in Fig. 3(a). To estimate these errors, we record a distribution of output magnitudes at the network output given random $\boldsymbol{x}$, Q. Using this, we assume we can achieve an experimental estimate of σ_{out} measured across many devices. As expected and shown in Figs. 3(b) and 3(c), the distribution follows a discretized half-normal distribution with Gaussian-distributed spikes at each of the possible outputs.

Next, as shown in Figs. 3(d)–3(f), we perform an error mitigation analysis by singular value permutation as discussed in Section 3. The SVD is invariant given any permutation identically applied to the rows of $V^\dagger$, the columns of U, and the singular values of Σ, i.e., $Q = U\Sigma V^\dagger = (UP)(P\Sigma)(PV^\dagger)$, where P is a matrix that implements the permutation. Therefore, error mitigation is possible by applying different P to the R meshes implementing Q. The proof of invariance is that $Q_{ij} = \sum_{k=1}^{N} U_{ik} \Sigma_{kk} V_{kj}^\dagger$ and k can be relabeled in any order, resulting in the same Q by symmetric property of addition. In our case, we average the result over four cyclic permutations of the singular values, i.e., (1, 2, 3, 4), (4, 1, 2, 3), (3, 4, 1, 2), (2, 3, 4, 1). As expected in Fig. 3(e), error is roughly halved when $R = 4$; the slope is reduced by about 41%, not quite 50%, possibly due to noise (Supplement 1, Fig. S3). Therefore, averaging results over devices implementing Q with permuted singular values can significantly reduce the error at the expense of increased device footprint.

Finally, in Fig. 3(g), we consider the "error dispersion" relation $\varepsilon(\lambda)$, exploring the effect of wavelength on the error to explore the possibility of parallelizing the computation over multiple wavelengths in a 20 nm wide band at our empirically determined optimal wavelength $\lambda_c = 1560$ nm:

$$\varepsilon(\lambda) \approx \langle \varepsilon \rangle [1 + D_\varepsilon (\Delta\lambda)^2], \quad (2)$$

where $\Delta\lambda = \lambda - \lambda_c$, and D_ε is the "relative" error dispersion (that depends on N, K) evaluated at λ_c. Our results in Fig. 3(g) indicate that this relative dispersion coefficient as defined in Eq. (2) actually *decreases* slightly with K. Note that there is an increase in absolute dispersion, but a decrease in relative error dispersion. More broadband rapid adiabatic couplers (at the expense of size [29]) and multimode interferometers (at the expense of loss), along with long and broadband phase shifters, can improve the error dispersion, possibly allowing for parallelization or batching of multiple bitvector inputs through the same device—an idea left to future work.

The ultimate goal of our experimentally measured hash error rate prediction is to determine the conditions for which a photonic system could feasibly solve LightHash in the presence of realistic errors. Our findings suggest error mitigation results in a larger range of "feasible" K values for $N = 4$ [defined to be $<1\%$ hash rate error indicated by a dotted line in Fig. 3(f)], which would be the first, to our knowledge, experimental proof of feasible digitally verifiable photonic computing for cryptocurrency and potentially other blockchain applications.

6. DISCUSSION AND OUTLOOK

Our results suggest that a digitally verifiable photonic mesh for PoW applications such as cryptocurrency requires sufficient input optical power and well-calibrated, precisely calibrated photonic circuits to guarantee that the error rate is sufficiently small to verify transactions with high probability.

In this paper, we improve on past proposals (e.g., HeavyHash), which can be too sensitive to error. First, we propose LightHash to modulate the "difficulty" of a problem by changing the bit resolution K. If a photonic mesh is used to implement LightHash, for larger values of N and K, the likelihood of error is dramatically increased, and the output error scales roughly as $\sigma_{\mathrm{out}} \propto NK\sigma$, where σ is a component-wise phase or coupling error (in radians) or component-wise loss error (in dB) localized to the phase shifters. In addition, noise can play a major role in the energy-error trade-off [Fig. 2(e)], which may dominate energy consumption at large N due to optical losses in photonic circuits (Supplement 1, Section 4). We bound N to guarantee minimum state space size [large $(2K)^N$, Supplement 1], and optical power does not dominate the overall power [Fig. 2(e)]. Second, we propose hardware-agnostic error mitigation to reduce the error in addition to current error mitigation protocols such as self-configuration [1,22], hardware-aware error mitigation [24], and gradient-based approaches [30].

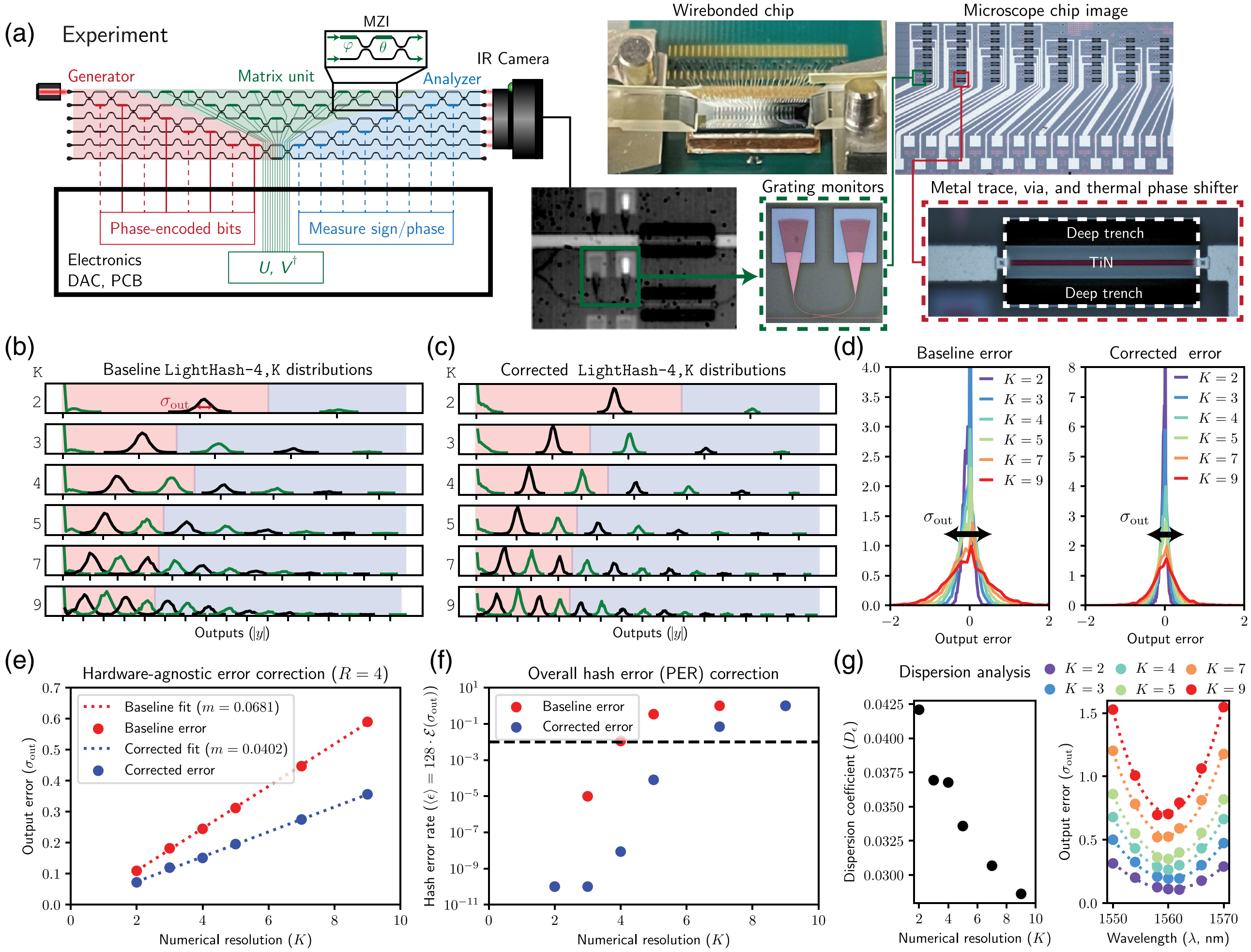

Fig. 3. Experimental evaluation of LightHash. (a) The experimental setup used to evaluate the LightHash protocol ($N = 4$, variable K) involves running U, V on-chip and multiplying singular values off-chip. The output measurements are made using IR camera readings of grating tap monitors placed along the output waveguides of the photonic mesh [28]. The physical wirebonded and thermally controlled setup along with a microscope image and image of integrated thermal phase shifters are also shown. (b) Outcome LightHash histograms for 250 random matrices Q given $N = 4$ for varying K for baseline and (c) hardware-agnostic error-mitigated implementations. We label alternating colors green and black to clearly delineate the overlap regions between successive values spaced 2 apart, labeled by tick marks. The red and blue regions correspond to a bit assignment of 0 and 1, respectively, by digital thresholding. (d) Comparison between the baseline and hardware-agnostic error mitigation error distributions [subtracting the ideal values from the outcome histograms in (b) and (c)]. (e) The standard deviation of the error, σ_{out}, is roughly proportional to K for both the baseline and corrected (error-mitigated) implementations, with the corrected implementation having a much smaller slope m. (f) Sharp transition in feasibility is demonstrated for the baseline and error-mitigated cases as a function of K (similar behavior is expected for N). (g) Dispersion of the error given calibration at the center wavelength 1560 nm shows parabolic increase in error around the center wavelength as expected, but the dispersion coefficient interestingly decreases with K.

To achieve feasible blockchain technology for cryptocurrency mining, we must reduce hash error rate $\langle\varepsilon\rangle$ and improve energy efficiency using sufficient reduction of systematic error σ_{out}, low-loss optical components, and development of robust, low-energy photodetector circuitry. Error mitigation resulting in a decrease of σ_{out} from 0.5 to 0.25 (using $R = 4$, which multiplies device footprint by four) can reduce $\langle\varepsilon_b\rangle$ by four orders of magnitude. This observation, in addition to Figs. 2(c) and 3(f), suggests that the feasibility barrier is sharp, so error mitigation reduces σ_{out} mostly in cases where feasibility is marginal. Other viable optical MVM architectures for LightHash (e.g., photoelectric multiplication using homodyne detector banks [18] or crossbar arrays [31]) may provide robust operation for accelerating low-bit precision MAC operations at large scale and are left to future work.

Our results justify the choice of photonic blockchain and oPoW over a digital alternative [27] to carry out the LightHash PoW scheme. First, LightHash miners would choose photonics since the energy efficiency (Fig. 2) and reduced latency for photonic matrix multiplies [17] lead to higher profits. With increased adoption, "mining pools" using photonic hardware can result in a consistent stream of income for a photonic versus digital miner (Supplement 1, Section 2). Our total photonic energy projection for $N = 64$ LightHash is less than 10 fJ/MAC [Fig. 2(e)], up to an order of magnitude less energy than digital hardware implementing equivalent MVM [27] (Supplement 1, Section 5). Assuming SHA-256 is handled by the Bitmain Antminer S19 Pro operating at 0.03 pJ per hash (pJ/H) efficiency [32], we estimate that the corresponding LightHash energy is roughly 7× lower for photonic implementations (roughly 0.194 nJ/H for photonic hardware versus 1.34 nJ/H for competing digital hardware [27]). Second, photonic hardware used in hash protocols can also be used for other applications, i.e., the hardware is not necessarily an application-specific device

(Supplement 1, Section 3). Importantly, the chip we use to explore cryptographic hash functions was used to perform inference tasks and backpropagation training in photonic neural networks [28]. Thus, in the context of LightHash, photonic mining hardware has key advantages over digital application-specific hardware that implements energy-efficient cryptography but serves no other purpose.

Funding. Air Force Office of Scientific Research (FA9550-17-1-0002, FA9550-18-1-0186, FA9550-21-1-0312).

Acknowledgment. We acknowledge Advanced MicroFoundries (AMF) in Singapore for their help in building the necessary circuitry for our demonstration. We also acknowledge funding from AFOSR in collaboration with UT Austin and UC Davis under Dr. Ben Yoo. Thanks also to Payton Broaddus for helping with wafer dicing, Nagaraja Pai for advice on electronics and thermal control packaging, Joseph Kahn for discussions on photodetector noise, and Carsten Langrock and Karel Urbanek for their help in designing our movable optical breadboard. Finally, we thank Dirk Englund, Ryan Hamerly, and Saumil Bandyopadhyay for fruitful discussions on error modeling as well as Mustafa Hammood, Vsevolod Hulchuk, and Maxim Karpov for illuminating feedback.

SP designed the photonic integrated circuit and ran all experiments with input from TP, NA, MM, FM, AM, OS, SF, and DM. SP, MM, FM, AM contributed to the photonic mesh design. SP conceptualized and simulated the LightHash protocol with input from BP, MD, and MB. TP designed the custom PCB with input from SP. SP wrote the manuscript with input from all coauthors.

Disclosures. SP, MD, MB, and BP all own a nominal amount of optical Bitcoin (oBTC), a cryptocurrency launched in 2021 that uses the HeavyHash (not LightHash) protocol for PoW and are involved with PoWx, a nonprofit organization dedicated to energy-efficient optical computing for cryptography. MB and BP also own a nominal amount of Kaspa. SP, MD, BP, MB, SF, OS, and DM are coinventors on a provisional patent application for LightHash, Prov. Appl. No.: 63/323727. DM holds patents on the SVD photonic mesh architecture. The authors declare no other conflicts of interest.

Data availability. All software and data for running the simulations and experiments are available through Zenodo [33] and Github through the Phox framework, including our experimental code via Phox [34], simulation code via Simphox [35], and automated photonic circuit design code via Dphox [36].

Supplemental document. See Supplement 1 for supporting content.

REFERENCES

1. D. A. B. Miller, "Self-configuring universal linear optical component (Invited)," Photon. Res. **1**, 1 (2013).
2. W. Bogaerts, D. Pérez, J. Capmany, D. A. Miller, J. Poon, D. Englund, F. Morichetti, and A. Melloni, "Programmable photonic circuits," Nature **586**, 207–216 (2020).
3. J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, "Machine learning method for state preparation and gate synthesis on photonic quantum computers," Quantum Sci. Technol. **4**, 024004 (2019).
4. J. Carolan, C. Harrold, C. Sparrow, E. Martín-López, N. J. Russell, J. W. Silverstone, P. J. Shadbolt, N. Matsuda, M. Oguma, M. Itoh, G. D. Marshall, M. G. Thompson, J. C. Matthews, T. Hashimoto, J. L. O'Brien, and A. Laing, "Universal linear optics," Science **349**, 711–716 (2015).
5. A. Annoni, E. Guglielmi, M. Carminati, G. Ferrari, M. Sampietro, D. A. Miller, A. Melloni, and F. Morichetti, "Unscrambling light—automatically undoing strong mixing between modes," Light Sci. Appl. **6**, e17110 (2017).
6. Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, "Deep learning with coherent nanophotonic circuits," Nat. Photonics **11**, 441–446 (2017).
7. B. Murmann, "Mixed-signal computing for deep neural network inference," IEEE Trans. Very Large Scale Integr. Syst. **29**, 3–13 (2021).
8. M. Dubrovsky, M. Ball, L. Kiffer, and B. Penkovsky, "Towards optical proof of work," in *Cryptoeconomic Systems* (2020).
9. Y. Sompolinsky, S. Wyborski, and A. Zohar, "PHANTOM GHOSTDAG: A scalable generalization of Nakamoto consensus: September 2, 2021," in *Proceedings of the 2021 3rd ACM Conference on Advances in Financial Technologies* (2021), pp. 57–70.
10. S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, https://bitcoin.org/bitcoin.pdf.
11. M. Dubrovsky and B. Penkovsky, "BIP 52: Durable, low energy bitcoin PoW," 2021, https://github.com/bitcoin/bips/blob/master/bip-0052.mediawiki.
12. J. Abou Jaoude and R. George Saade, "Blockchain applications—usage in different domains," IEEE Access **7**, 45360–45381 (2019).
13. A. Back, "Hashcash-A denial of service counter-measure," 2002, https://hashcash.org.
14. V. Kohli, S. Chakravarty, V. Chamola, K. S. Sangwan, and S. Zeadally, "An analysis of energy consumption and carbon footprints of cryptocurrencies and possible solutions," in *Digital Communications and Networks* (2022).
15. M. Shu, R. Song, and W. Zhu, "The 2021 Bitcoin bubbles and crashes: detection and classification," Stats **4**, 950–970 (2021).
16. A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: a provably secure proof-of-stake blockchain protocol," Katz, J., Shacham, H. (eds) in *Advances in Cryptology – CRYPTO 2017. Lecture Notes in Computer Science* Vol. **10401** (Springer, Cham, 2017), 357–388.
17. M. A. Nahmias, T. F. De Lima, A. N. Tait, H. T. Peng, B. J. Shastri, and P. R. Prucnal, "Photonic multiply-accumulate operations for neural networks," IEEE J. Sel. Top. Quantum Electron. **26**, 7701518 (2020).
18. R. Hamerly, L. Bernstein, A. Sludds, M. Soljačić, and D. Englund, "Large-scale optical neural networks based on photoelectric multiplication," Phys. Rev. X **9**, 021032 (2019).
19. M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," Phys. Rev. Lett. **73**, 58–61 (1994).
20. W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walsmley, "Optimal design for universal multiport interferometers," Optica **3**, 1460 (2016).
21. M. Miyahara, Y. Asada, D. Paik, and A. Matsuzawa, "A low-noise self-calibrating dynamic comparator for high-speed ADCs," in *Proceedings of 2008 IEEE Asian Solid-State Circuits Conference, A-SSCC* (2008), pp. 269–272.
22. R. Hamerly, S. Bandyopadhyay, and D. Englund, "Accurate self-configuration of rectangular multiport interferometers," Phys. Rev. Appl. **18**, 024019 (2022).
23. S. Bandyopadhyay, A. Sludds, S. Krastanov, R. Hamerly, N. Harris, D. Bunandar, M. Streshinsky, M. Hochberg, and D. Englund, "Single chip photonic deep neural network with accelerated training," arXiv, arXiv:2208.01623 (2022).
24. S. Bandyopadhyay, R. Hamerly, and D. Englund, "Hardware error correction for programmable photonics," Optica **8**, 1247 (2021).
25. S. P. Kumar, L. Neuhaus, L. G. Helt, H. Qi, B. Morrison, D. H. Mahler, and I. Dhand, "Mitigating linear optics imperfections via port allocation and compilation," arXiv, arXiv:2103.03183 (2021).
26. J. K. Perin, M. Sharif, and J. M. Kahn, "Sensitivity improvement in 100 Gb/s-per-wavelength links using semiconductor optical amplifiers or avalanche photodiodes," J. Lightwave Technol. **34**, 5542–5553 (2016).
27. R. Ni, H.-M. Chu, O. Castañeda, P.-Y. Chiang, C. Studer, and T. Goldstein, "WrapNet: neural net inference with ultra-low-resolution arithmetic," in *International Conference on Learning Representations* (2020).
28. S. Pai, Z. Sun, T. W. Hughes, T. Park, B. Bartlett, I. A. D. Williamson, M. Minkov, M. Milanizadeh, N. Abebe, F. Morichetti, A. Melloni, S. Fan, O. Solgaard, and D. A. B. Miller, "Experimentally realized in situ backpropagation for deep learning in nanophotonic neural networks," arXiv, arXiv:2205.08501 (2022).
29. J. M. Fargas Cabanillas, D. Onural, and M. A. Popović, "Rapid adiabatic 3 dB coupler with 50 ± 1% splitting over 200 nm including S, C and L bands in 45 nm CMOS platform," in *Frontiers in Optics + Laser Science* (Optica Publishing Group, 2021), paper FTu6B.2.
30. S. Pai, B. Bartlett, O. Solgaard, and D. A. B. Miller, "Matrix optimization on universal unitary photonic devices," Phys. Rev. Appl. **11**, 064044 (2019).
31. J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja, J. Liu, C. D. Wright, A. Sebastian, T. J. Kippenberg, W. H. Pernice, and H. Bhaskaran, "Parallel convolutional processing using an integrated photonic tensor core," Nature **589**, 52–58 (2021).

32. B. Bijelić, M. Hercog, G. Horvat, I. Ostheimer, and M. Vukobratović, “Modeling energy aspects of ASIC hardware for pow applications,” in *45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)* (2022), pp. 60–64.
33. S. Pai, “solgaardlab/photoniccrypto: data and code for the paper “Experimental evaluation of digitally-verifiable photonic computing for blockchain and cryptocurrency”,” Zenodo (2022), https://doi.org/10.5281/zenodo.6557372.
34. S. Pai, Z. Sun, and T. Park, “Phox: base repository for simulation and control of photonic devices,” Github (2022), https://github.com/solgaardlab/phox/.
35. S. Pai, “Simphox: Another inverse design library,” Github (2022), https://github.com/fancompute/simphox.
36. S. Pai and N. Abebe, “Dphox: photonic layout and device design,” Github (2022), https://github.com/solgaardlab/dphox.