ReFOCUS: Reusing Light for Efficient Fourier Optics-Based Photonic Neural Network Accelerator

Shurui Li
University of California, Los Angeles
Los Angeles, CA, USA
shuruili@ucla.edu

Hangbo Yang
University of California, Los Angeles
Los Angeles, CA, USA
yanghumble@ucla.edu

Chee Wei Wong
University of California, Los Angeles
Los Angeles, CA, USA
cheewei.wong@ucla.edu

Volker J. Sorger
University of Florida
Gainesville, FL, USA
volker.sorger@ufl.edu

Puneet Gupta
University of California, Los Angeles
Los Angeles, CA, USA
puneetg@ucla.edu

ABSTRACT

In recent years, there has been a significant focus on achieving low-latency and high-throughput convolutional neural network (CNN) inference. Integrated photonics offers the potential to substantially expedite neural networks due to its inherent low-latency properties. Recently, on-chip Fourier optics-based neural network accelerators have been demonstrated and achieved superior energy efficiency for CNN acceleration. By incorporating Fourier optics, computationally intensive convolution operations can be performed instantaneously through on-chip lenses at a significantly lower cost compared to other on-chip photonic neural network accelerators. This is thanks to the complexity reduction offered by the convolution theorem and the passive Fourier transforms computed by on-chip lenses. However, conversion overhead between optical and digital domains and memory access energy still hinder overall efficiency.

We introduce ReFOCUS, a Joint Transform Correlator (JTC) based on-chip neural network accelerator that efficiently reuses light through optical buffers. By incorporating optical delay lines, wavelength-division multiplexing, dataflow, and memory hierarchy optimization, ReFOCUS minimizes both conversion overhead and memory access energy. As a result, ReFOCUS achieves 2× throughput, 2.2× energy efficiency, and 1.36× area efficiency compared to state-of-the-art photonic neural network accelerators.

CCS CONCEPTS

• **Computer systems organization** → **Architectures**; • **Hardware** → **Emerging technologies**; **Emerging optical and photonic technologies**; • **Computing methodologies** → *Artificial intelligence*; *Machine learning*.

KEYWORDS

Photonic neural network, on-chip photonics, Fourier optics, 4F system, deep learning, neural network accelerator

MICRO '23, October 28-November 1, 2023, Toronto, ON, Canada

ACM ISBN 979-8-4007-0329-4/23/10.
https://doi.org/10.1145/3613424.3623798

ACM Reference Format:
Shurui Li, Hangbo Yang, Chee Wei Wong, Volker J. Sorger, and Puneet Gupta. 2023. ReFOCUS: Reusing Light for Efficient Fourier Optics-Based Photonic Neural Network Accelerator. In *56th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '23), October 28-November 1, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3613424.3623798

1 INTRODUCTION

Convolutional neural networks (CNNs) [18, 23, 27, 49, 54, 58] have become indispensable in modern Artificial Intelligence (AI) applications, forming the basis of numerous computer vision tasks such as image classification, object detection, and autonomous driving. Although vision transformers [10, 14, 59] are gaining popularity, CNNs still maintain an edge in terms of model compactness and the ability to achieve comparable accuracy to vision transformers with significantly fewer parameters [2]. Due to the complexity of convolution operations, executing them on general-purpose processors is not energy efficient. Therefore, researchers have focused on developing domain-specific accelerators employing parallel architectures for energy-efficient computation of neural networks [11, 29, 38, 45, 50]. However, the ever-increasing complexity of modern CNNs, the end of Dennard scaling, and the slowdown of Moore's law have imposed limitations on CMOS digital accelerators concerning energy consumption for data movement and computation. Silicon photonics emerging as a promising solution to this problem, which offers remarkable computational parallelism and efficiency.

Photonics components possess several unique advantages, including high frequency, relatively low power consumption, and no RC delay. These characteristics make photonics an unparalleled contender for low-latency and low-power computation. Generally, there are two types of photonic neural network accelerators: free-space and on-chip versions. While free-space optical neural network accelerators [8, 12, 22, 24, 34, 40] are often bulky and inflexible, on-chip photonics-based accelerators have gained significant interest due to their efficiency and flexibility. On-chip photonics can be further classified into two main categories. Most existing works compute dot products or vector-matrix multiplications using Mach-Zehnder Interferometers (MZI) and/or micro-ring resonators (MRR) [5, 20, 33, 36, 52, 53, 56, 71]. These MZI/MRR-based photonic neural network accelerators share similarities with compute-in-memory (CIM) analog accelerators but feature high clock frequencies (5-10

GHz) and the possibility of leveraging wavelength-division multiplexing (WDM) for extra parallelism. However, a major bottleneck of photonic and other analog neural network accelerators is the conversion cost between digital and analog domains, which can consume a significant amount of power. Unlike CIM accelerators which are typically designed to have tall columns to reduce the compute-to-conversion ratio, photonic neural network accelerators often have significantly smaller arrays because of relatively large photonic components and limitations of WDM. This results in lower compute-to-conversion ratios. Morever, the conversion overhead between digital and optical domains often prevents photonic neural networks from delivering their theoretical advantage over CMOS electronics.

The second category focuses on computing the convolution directly. This can be achieved through a pair of Fourier lenses that compute the Fourier transform passively. Fourier optics-based designs capitalize on the convolution theorem, which states that convolution in the space domain is equivalent to point-wise multiplication in the Fourier domain. These systems, commonly referred to as 4F systems, utilize time-of-flight Fourier transform via Fourier lenses to reduce convolution complexity from $O(N^2)$ to $O(N)$. Compared to conventional MZI/MRR-based photonic neural network accelerators, 4F systems require significantly fewer optical components to perform the same amount of computations, thanks to the complexity reduction. This type of photonic neural network accelerator was typically built as a free-space system, but recently, silicon photonics versions have been proposed, opening a new direction for designing efficient photonic neural network accelerators. [32] proposed a Joint Transform Correlator (JTC) based on-chip photonic neural network accelerator, which is a variant of the 4F system (still using Fourier optics), and achieved orders of magnitude better efficiency than previous state-of-the-art photonic neural network accelerators. JTC computes the auto-convolution of two input signals using a pair of Fourier lenses similar to 4F systems, but it uses spatial filters instead of complex-valued Fourier-domain filters. JTC addresses some limitations of conventional 4F systems, such as the support for complex filters and the large filter size (as Fourier-domain filters need to have the same size as inputs).

Although the JTC-based photonic neural network accelerator already demonstrates state-of-the-art efficiency, there is still substantial room for further optimizations. On one hand, the conversions between analog and digital domains still consume a large proportion of system power. On the other hand, as computation becomes even more efficient, memory access power becomes non-negligible. Both of these aspects could be optimized to further improve system efficiency.

In this work, we propose ReFOCUS, a JTC based on-chip photonic neural network accelerator that reuses light through optical buffers to minimize the conversion cost between optical and digital domains. With optical reuse and various optimizations, ReFOCUS is able to achieve significantly better energy efficiency compared to state-of-the-art photonic neural network accelerators. The main contributions can be summarized as follows:

- We propose optical reuse based on optical buffers constructed using optical delay lines, and incorporate corresponding dataflow and laser power optimization to significantly improve the power efficiency of the system.
- We adopt wavelength-division multiplexing (WDM) to improve the area efficiency by sharing on-chip lenses, which also reduces the area overhead of optical buffers.
- ReFOCUS can achieve 2× throughput, 2.2× energy efficiency and 1.36× area efficiency than previous state-of-the-art photonic neural network accelerator.

2 BACKGROUND

2.1 Background of JTC

Over the past couple of decades, JTC has found applications in a variety of fields, such as image filtering [25, 60] and object tracking [37, 57]. Recently, JTC systems have been used for accelerating neural networks [17, 32, 46, 67]. Theoretical analysis and experimental demonstrations of low-latency convolution operations using JTC systems have been presented in [17] and [46, 67] respectively, while [32] proposed the architecture-level design and optimizations.

The math behind JTC operations has been adequately discussed and analyzed in previous literature, so we will not go into too much detail in this paper as the focus is on architecture design and optimization. Still, we will provide a brief introduction to JTC operations for easier understanding.

Optical lenses can perform a Fourier transform $\mathcal{F}[\tilde{E}(x, y, f)]$ on their back focal plane when an input image $\tilde{E}(x, y, f)$, illuminated by a coherent light source, is placed at the front focal plane [19]. Utilizing the Fourier transform property of lenses, [63] introduced an optical JTC that generates optical convolution with both phase and amplitude components. A 1D on-chip photonic JTC can be derived from a traditional 2D optical JTC with minor modifications. There are five main components in a typical on-chip JTC system: (1) a 1D multi-channel input beam containing a signal $s(x + x_s)$ and a kernel $k(x - x_k)$ (with x_s and x_k representing the offsets of s and k from the origin in the x direction); (2) the first on-chip lens, which functions like a traditional free-space lens, to achieve the 1D Fourier transform $\mathcal{F}\left[s\left(x + x_s\right) + k\left(x - x_k\right)\right]$; (3) a *nonlinear function* unit (not the activation function of neural networks), realized using photodetectors and electro-optic modulators (EOM) or nonlinear materials to achieve a *square function* at the Fourier plane which is essential for JTC operation ; (4) the second on-chip lens, to transform the signal back to spatial domain; (5) photodetectors that detect the intensity pattern of the computed convolution by the JTC:

$$s(x + x_s + x_k) * k(-x) + s(-x) * k(x - x_s - x_k) + N(x) \quad (1)$$

, where $*$ denotes convolution. The first and second terms represent the computed auto-convolution between the two inputs. The third term $N(x)$, equals to $\mathcal{F}\left[|S\left(x\right)|^2 + |K\left(x\right)|^2\right]$, is a non-convolution term that can be spatially filtered out. Figure 1 illustrates the high-level diagram of a typical on-chip JTC system, which includes the five main components. Besides the 5 photonic components, DACs and ADCs are also required to convert the signals to and from the optical domain. The non-linear function in JTC, applied in the frequency domain after the first lens, is crucial for computing the

convolution, as the output would be identical to the input without it (Fourier transform followed by inverse Fourier transform). The primary difference between on-chip JTC and conventional free-space JTC is the replacement of 2D lenses with 1D on-chip lenses, which results in the computation of 1D convolutions instead of 2D convolutions. This will be further discussed in Section 2.2. In this work, we assume the non-linear function is achieved through passive non-linear materials [4, 6, 26, 41], which is also used in the NG version of [32].

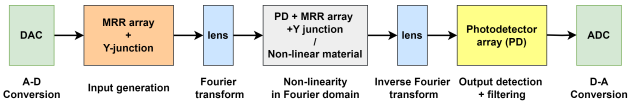

Figure 1: High-level diagram of a typical on-chip JTC system.

2.2 Computing 2D convolutions using 1D JTC

Unlike free-space 4F/JTC systems that naturally support 2D convolutions through the use of 2D Fourier lenses, their on-chip counterparts can only employ 1D on-chip metasurface-based lenses, and therefore, by default, only support 1D convolutions. To enable on-chip JTC systems to perform 2D convolutions, [32] proposed a generic algorithm for computing 2D convolutions using 1D convolutions, which is applicable to JTC systems. With this algorithm, 2D convolution can be computed using 1D convolution with no computation overhead for digital systems. For JTC-based systems, the supported 1D Fourier transform size needs to be large enough to avoid computation overhead. The core idea involves row tiling and partitioning, in which rows of 2D inputs and kernels are tiled with zero padding to form 1D inputs and kernels for 1D convolution. For $k \times k$ kernels, row tiling can be implemented if the JTC can accommodate at least K rows of inputs. This method can achieve identical results to conventional 2D convolutions when input rows are zero-padded with $k-1$ zeros per row and can closely approximate conventional 2D convolutions without zero-padding. While the 1D kernels needed to be zero-padded to the size of 1D input tiles, the zero-padding does not add overhead to JTC systems thanks to a unique property of JTC. For JTC, the actual convolution can be computed by the optical components passively, drawing almost no power. The computation cost comes from the input generation and output conversion part. For the zero-padding part, since all values are zero, the corresponding DACs and MRRs can be switched off so that no power will be consumed.

In cases where the JTC cannot hold k rows of inputs, 2D convolutions can still be computed by partially tiling or partitioning the input rows and taking multiple cycles to generate a single output row. The convolution results are identical to those obtained in the row tiling case but require more iterations. Since JTC can typically support a large number of input waveguides (>256), and CNNs usually incorporate multiple pooling layers to reduce activation size, partial row-tiling or row-partitioning generally occurs only during the execution of the first layer, where activation sizes are large. Therefore, the overhead of partial row-tiling and row-partitioning is negligible.

An example of performing 2D convolution with a 3×3 kernel using the on-chip JTC system is illustrated in Figure 2. In this example the input (activation) size is larger than the number of input waveguides in the JTC, therefore multiple iterations are required to compute the full convolution. The input is split into chunks and the rows in one chunk are tiled and loaded into the JTC. The kernel rows are padded to the same size as the input rows and also tiled and loaded into the JTC. The convolution between the tiled input rows and kernel rows completes in one cycle, and the output is received by the photodetectors and ADCs. Because of the circular padding nature of Fourier transform-based convolutions in JTC, only two output rows are valid in this example. Consequently, the invalid rows are discarded, constituting the primary source of computation overhead. The process is repeated multiple times to complete the convolution of the entire 2D input. The number of valid output rows is $R_i - k$ for $k \times k$ kernels, where R_i is number of input rows that can be tiled on the JTC. Therefore, the effective utilization is higher for larger JTCs and smaller input activations.

Comparing the amount of operations required for processing convolutions of digital systems (e.g., GPUs) and JTCs is non-trivial due to JTC's passive computation nature. However, if assuming the JTC's computational requirement is the number of input conversions needed, JTC with 256 input waveguides requires more than 5 times fewer computations than a GPU when computing a convolution between a 32×32 input and a 3×3 kernel. For JTC, each pass can tile 8 rows and generate 6 valid outputs $(8-2)$, thereby requiring 6 JTC passes to compute the actual value. This leads to 1590 conversions in total $(6 \times (256+9))$ while GPU typically requires 9216 multiply-and-accumulate operations $(32^2 \times 3^2)$.

3 A CASE STUDY FOR A TYPICAL JTC-BASED ACCELERATOR

In this section, we briefly introduce the baseline system of ReFOCUS, and analyze its bottlenecks while discussing how to further improve the efficiency of Fourier optics-based accelerators. We use a slightly modified version of PhotoFourier-NG (next-gen version) [32], the state-of-the-art Fourier-optics based photonic neural network accelerator, as our baseline system. The baseline system keeps the architecture of Photo Fourier-NG, which includes 16 JTCs in parallel, assumes monolithic integration of CMOS and photonics, and incorporates passive non-linear materials. The modification we made is to use citable sources for ADC and DAC power. The average power and the area of the baseline system are 15.7W and 116.3 mm^2 (90.7 mm^2 for photonic components) respectively.

The power breakdown comparison of a single JTC system (no optimizations) and our baseline system is illustrated in Figure 3 (a). It is evident that for the single JTC system, the overall power consumption is dominated by ADCs and DACs (> 85%). ReFOCUS-baseline exhibits reduced ADC power due to the implementation of an optimization technique called temporal accumulation introduced in [32], which accumulates convolution results before ADC readout using photodetectors, resulting in a significant reduction of ADC power consumption. The DAC and SRAM access power constitute a large proportion of the total system power, highlighting the need for further optimization. By reducing their power consumption, the efficiency of the baseline system can be enhanced. Area-wise, as

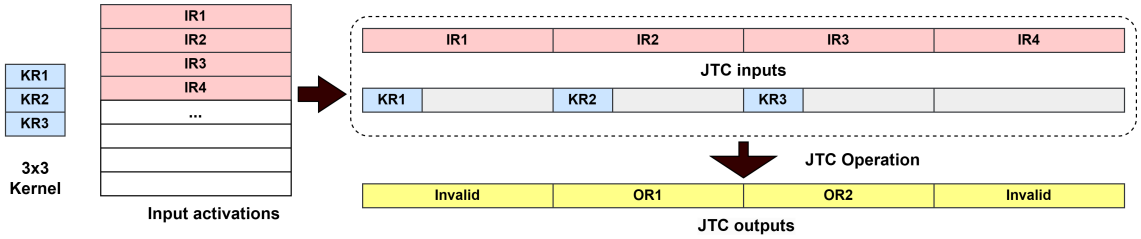

Figure 2: Illustration of how 2D convolution is computed using on-chip JTC system. IR, KR, and OR stand for input row, kernel row, and output row. The gray block represents zero-padding.

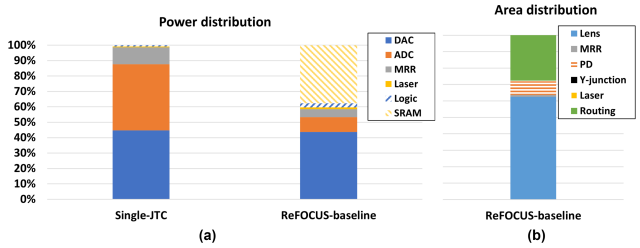

Figure 3: (a) Power breakdown of single JTC system and ReFOCUS-baseline. (b): Area breakdown of ReFOCUS-baseline, only photonic components are included.

demonstrated in Figure 3 (b), the lens area dominates, consuming more than 50% of the total area. Therefore, reducing lens area is crucial for achieving better area efficiency.

In ReFOCUS, we propose optical reuse and WDM to mitigate the power consumption of both the DACs and the memory accesses, which are the two most dominating factors in total power consumption, thereby enhancing overall energy efficiency. These two optimizations, along with the architecture-level optimizations, will be discussed in detail in Section 4 and 5.

4 REFOCUS COMPUTE UNIT

ReFOCUS comprises multiple compute units, which are named as ReFOCUS Compute Unit, or RFCU in short. Each RFCU is essentially a JTC system described in Section 2.1. The JTC configuration of the RFCU is kept the same as [32] unless related to optical reuse since this work focuses on optical reuse. Each RFCU has 256 input waveguides and 25 active weight waveguides (active means waveguides with DACs). On top of the baseline design, we introduce two main optimizations to improve energy efficiency and area efficiency.

4.1 Optical reuse

As discussed in Section 3, the ADC power can be reduced by temporal accumulation. However, this technique does not effectively reduce the DAC power, necessitating further optimization. One approach to decrease DAC power is to reuse the optical signals generated by the DACs. Reusing can be easily achieved in digital electronics through data buffers, but this proves to be non-trivial in photonics/optics due to the absence of optical memory. Despite this, optical buffers can be achieved through the use of optical delay lines. Optical delay lines essentially consist of spiral waveguides that require light signals to travel a relatively long distance within the delay line, consequently causing a delay. The delay line length can be calculated by multiplying the speed of light by the target delay time. The waveguides are placed in a spiral shape to minimize the area, as depicted in Figure 4 (the red square). The light signal is split into two parts, and one part travels through the delay line to be reused at a later time, such that DACs do not need to be active when light is reused from the delay line, effectively reducing the average DAC power. To accomplish optical reuse, we propose two versions of optical buffer design based on optical delay lines, which have different use cases. In this work, both optical buffer designs will be adopted and evaluated, hence forming two versions of ReFOCUS - ReFOCUS-FB (feedback) and ReFOCUS-FF (feedforward).

4.1.1 Feedback optical buffer. The schematic diagram of the feedback version of the optical buffer design is depicted in Figure 4 (a), which comprises a delay line module, a switch MRR, and a Y-junction. The input signals generated by the DAC are divided into two parts by a Y-junction. One part is used for JTC computation, while the other is designated for reuse. The reuse signal passes through the optical delay line module and returns to be reused N cycles later, where N is determined by the delay line length. An MRR is required as a switch to control whether the feedback should be used for computation since, when a new input signal is generated by the input MRRs, the reuse signal should be blocked to avoid corruption of the final input. For instance, if a second Y-junction is employed to replace the switch MRR, the delayed optical will be added to the main signal that goes to the first Y-junction and the JTC even when the JTC is supposed to receive new input activations, causing data corruption. A switch MRR can be turned off to block the feedback signal. When the switch MRR is turned on, the reuse signal will be coupled to the main waveguide connected to the Y-junction, and the input MRR should be turned off to avoid data corruption.

The advantage of this feedback approach is that, theoretically, the signals can be reused as many times as desired, which can maximize the reuse and significantly cut down the DAC power. However, one potential limitation of this design is that the signal power of the feedback signal will be lower with every iteration due to the Y-junction and the delay line loss. Define the power split ratio of Y-junction as α (percentage of input power directed to the JTC), and the delay line loss as l_d, the relationship between the signal power that goes into the JTC can be derived as:

$$X_i = (1 - l_d) \cdot (1 - \alpha) \cdot X_{i-1} \tag{2}$$

, where X_i is the signal power that goes into the JTC for the i^{th} iteration. The overall signal loss for every reuse iteration l_t is hence $(1 - l_d) \cdot (1 - \alpha)$. The signal power of a particular iteration can then be calculated as:

$$X_i = ((1 - l_d) \cdot (1 - \alpha))^i \cdot X_0 \tag{3}$$

, where X_0 is the signal power of the initial input to the JTC.

Assuming the input activations are reused, typically different convolution filters will be processed each time the input is reused. That means different filters will see inputs with different magnitudes, which are supposed to be the same. Since the power reduction of the signals for each iteration is fixed and can be pre-determined, a hardware-aware scheduler can be designed to adjust the weights of the filters according to Equation 1, and the convolution outputs will be scaled back in the digital domain. In this case, the number of times the same signal can be reused is determined by the laser power overhead (average laser power will be higher to compensate for the loss due to delay lines), the dynamic range of photodetectors, and ADCs. This will be further analyzed in Section 5.4.

4.1.2 Feedforward optical buffer. In addition to the software solutions, there is a hardware solution to address the issue of the reduction of power of the reused signals, at the cost of the amount of achievable reuse. This solution involves using a feedforward optical buffer, as depicted in Figure 4 (b). The difference between the feedback version is that the delayed signal is not connected back to the input of the Y-junction; instead, it goes directly to the JTC through a second Y-junction. The second Y-junction is used to connect the delayed signal back to the main waveguide. A switch MRR is not required in this design, as there are no signal loops - which means the delayed signal does not need to be blocked. In this design, the split ratio α of the Y-junction can be configured to make the signal power of the original signal and the delayed signal identical. The signal power that directly goes to the JTC (without delay) is $\alpha \cdot X$, where X is the signal power before the first Y-junction. The signal power of the delayed signal is $(1 - l_d) \cdot (1 - \alpha) \cdot X$, where l_d is the loss of the delay line module. By equating the two signal powers, the split ratio can be calculated:

$$\alpha = \frac{1 - l_d}{2 - l_d} \tag{4}$$

By configuring the split ratio according to Equation 4, the original signal and the delayed (reused) signal will have the same signal power. This design eliminates the need for weight and activation scaling. However, the signal in this design can only be reused once since there is no feedback loop, which is the main limitation of the feedforward optical buffer.

4.1.3 Reusing signal or weight. In the context of neural networks, the JTC receives two signals to compute their convolution: inputs (activation) and weights. Consequently, there is a choice of whether to reuse inputs or weights. Assuming the processing of a 3×3 kernel, the number of input DACs is 256, while the number of weight DACs is 9 for a single JTC. Even considering the entire system and assuming input is fully broadcasted to 8 or 16 RFCUs, the number of input DACs remains significantly larger than the number of weight DACs. Therefore, reusing inputs will have a greater impact on power efficiency than reusing weights.

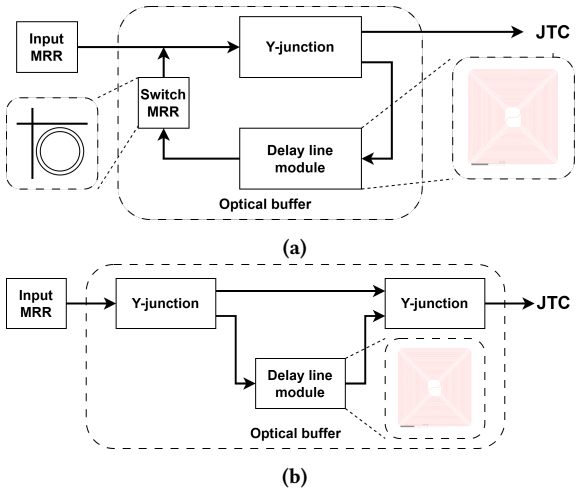

Figure 4: Schematic diagram of two versions of optical buffers used in ReFOCUS. (a): Feedback version. (b): Feedforward version.

Furthermore, reusing weights can lead to lower-than-expected performance improvement. For inference with a batch size of 1, if weights are reused, the only option is to process different input activation tiles for each iteration since they share the same weight. However, the JTC tile size is usually large (e.g., 256) for more inherent weight reuse within the JTC, while the input activation size can be small for later layers of CNNs due to pooling. For instance, ResNet-34 has 18 layers with input activation sizes small enough that the entire input can be loaded into a single JTC together, which means there will be no opportunity for temporal weight reuse at all. Reusing inputs will not have this problem, as the number of filters of modern CNNs is far larger than the number of filters that can be executed in parallel on ReFOCUS so that each cycle can process different filters.

4.1.4 Longer delay lines. Temporal accumulation can significantly reduce ADC frequency by accumulating the outputs of multiple cycles at photodetectors before the ADC readout, enabling the ADC to operate at much lower power. However, output stationary (OS) dataflow is required for temporal accumulation to function properly, as only the outputs of individual channels can be accumulated. The introduction of optical buffers to reuse inputs means the dataflow needs to be adjusted accordingly. Assuming the inputs are only delayed by 1 cycle, then input stationary dataflow will be enforced, and temporal accumulation cannot be implemented. If input reuse is achieved at the cost of removing temporal accumulation, it will not be an ideal design choice, as the increase in ADC power will have a greater impact than the reduction of DAC power.

Nevertheless, with a longer delay line and dataflow optimization, temporal accumulation can still be implemented by accumulating the results while the reused inputs are traveling through the delay

lines. An alternating dataflow (OS + input stationary (IS)) is required to implement temporal accumulation with a delay line. The maximum amount of temporal accumulation that can be achieved in terms of cycles is the same as the delay line length in terms of cycles. The alternating dataflow, choice of the exact length of the delay line, as well as how many times an input signal will be reused for ReFOCUS-FB, will be discussed in detail in Section 5.

4.1.5 Overhead of optical buffer. The components used in both versions of the optical buffers are passive, except for the switch MRR used in the feedback optical buffer, which consumes significantly less power compared to a high-speed DAC. Therefore, the power overhead of optical buffers is small (excluding laser power). However, the area overhead cannot be ignored, as the delay line modules are large in size, particularly if the signals need to be delayed for an extended period to implement temporal accumulation. Table 1 lists the length, area, and loss of the delay line which can delay the signal by one cycle for a 10 GHz system (0.1 ns). The delay line area is around $0.01mm^2$, which constrains the number of inputs that can be delayed and the number of cycles that can be delayed. The area overhead of optical buffers can be compensated through lens sharing and architecture-level optimizations, both of which will be discussed later.

In addition to the area overhead, delay lines also attenuate the signal. The total signal power loss is directly proportional to the delay line length. The average laser power will be higher compared to the case without optical buffers to compensate for the power loss caused by the delay line module, which will be discussed further in Section 5. With low-loss on-chip delay lines [28], the delay line loss is not significant for any reasonable delay line lengths.

Table 1: The length, area, and loss of a delay line with 0.1 ns delay (1 cycle in 10 GHz system).

# Length (mm)	Area (mm^2)	Loss (dB) [28]
8.57	0.01	6.94e-3

4.2 Lens sharing

4.2.1 Motivation. The optical buffer allows input reuse, which can reduce the DAC and memory access power of inputs. However, the delay line comes with a non-negligible area overhead. For a 16-RFCU system (each with 256 input waveguides), if the inputs of each RFCU are buffered individually for 8 cycles, the total delay line area is 327.7 mm^2, which is about 3× larger than the whole system area without the delay line and is clearly infeasible. The optical buffer area can be dramatically reduced through input broadcasting. By placing the optical buffer before the Y-junction tree that broadcasts the inputs, the total delay line area can be reduced to 20.48 mm^2 assuming full input broadcasting. Even in this case, the optical buffer still adds around 20% area overhead to the system.

To improve the overall area efficiency of the system, lens optimization is required, which accounts for around 50% of the total system area, as shown in Figure 3 (b). Wavelength-division multiplexing (WDM) is a common approach used in photonic designs to enhance parallelism and area efficiency, although it is primarily used in dot-product style on-chip photonic neural network accelerators and has not yet been applied to Fourier optics. WDM works by transmitting multiple data channels encoded into different wavelengths on a single waveguide, thus saving area. Furthermore, operations applied to the waveguide, such as phase change and delay, are effectively broadcasted to all wavelengths (data channels). For JTC, the Fourier transform implemented by the lens can also be broadcasted to the wavelengths through WDM, effectively sharing the lens. In this work, we leverage WDM to share lenses and photodetectors with different wavelengths, significantly improving the area efficiency of ReFOCUS.

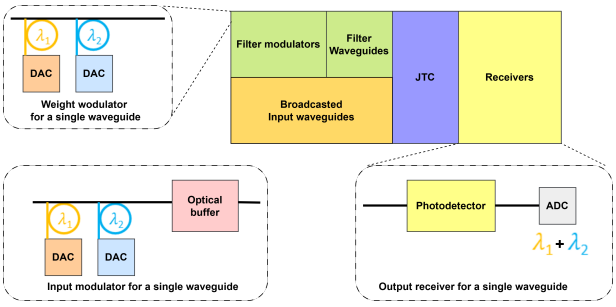

Figure 5: Illustration of how WDM is implemented in ReFOCUS (not drawn to scale). Two wavelengths are modulated and encoded into a single waveguide through MRRs with different wavelengths, for both filters and inputs generation. Photodetectors and ADCs receive the sum of the convolution output of the two wavelengths.

4.2.2 Implementation. When WDM is used for optical communications, encoders and decoders are required to encode and decode the signals. Both can be implemented with MRR arrays, with each MRR corresponding to one wavelength. For each wavelength, two MRRs are required for modulation/encoding and decoding, and one photodetector, ADC, and DAC. In the context of neural network acceleration, some of the components described above can be shared because of the reuse opportunity inside neural networks. Depending on the exact dataflow, either input DAC/MRR, weight DAC/MRR, or photodetector/ADC can be shared with different wavelengths. In ReFOCUS, each wavelength processes a single convolution channel, hence their convolution results can be directly accumulated. The decoder is no longer required in this case - the waveguide that contains multiple wavelengths is directly connected to a single photodetector and the convolution results of different wavelengths/channels are accumulated by the photodetector. The wavelengths should be selected to be close to each other so that their convolution results can be detected by a single photodetector. Figure 5 illustrates how WDM is implemented in ReFOCUS. In this example, 2 wavelengths are encoded into a single waveguide through MRRs with wavelength λ_1 and λ_2, for both inputs and weight generation. The photodetector receives the sum of the convolution results of both wavelengths.

In this implementation, the photodetector and ADC can be shared, and extra decoding MRRs are not required, which means WDM can improve area efficiency and power efficiency at the same

time. We choose to share ADCs rather than input or weight DACs for the following reasons: (1) As previously discussed, broadcasting weights to different activation tiles is not guaranteed, especially for later layers of CNNs, while inputs are already broadcasted to different RFCUs (and further reused through optical buffer). (2): Photodetectors can also be shared when sharing ADCs, which is not possible for the other two cases. Sharing photodetectors further improves area efficiency as they are around 10× larger than MRRs. Define N_λ as the number of wavelengths used, and in this dataflow, N_λ input channel needs to be generated. Since delay lines can also be shared by all wavelengths, processing multiple input channels will not cause excessive area overhead of optical buffers. WDM is applicable to both optical buffers, including the feedback version, as the switch MRR can react to a range of wavelengths.

4.2.3 Number of wavelengths. However, there is a limit on how many wavelengths can be used, and is relatively low for ReFOCUS. Having too many wavelengths can cause the spread of the convolution results of all wavelengths too large to be captured by a single photodetector, and our simulation suggests that the number of wavelengths should be less than 4. Besides, more wavelengths will make accessing inputs from memory/buffer challenging due to the huge number of data that needs to be accessed every cycle, as using temporal accumulation means inputs need to be accessed every cycle regardless of optical reuse. Considering both factors, we set $N_\lambda = 2$, that is using two wavelengths. With WDM, each RFCU essentially contains two 'virtual' JTCs and has 2× throughput, but only requires a single set of lenses and photodetectors.

Table 2: Area and normalized area efficiency in terms of frames per second per mm^2 of a 16-RFCU system with different wavelengths.

# wavelengths	Area (mm^2)	Normalized FPS/mm^2
1	111.3	1.00
2	115.2	1.93

Even with just two wavelengths, significant area efficiency can be achieved. Table 2 shows the area and normalized area efficiency of a 16-RFCU system with 1 or 2 wavelengths. Adding a second wavelength only increases the area by 3.5%, while doubling the throughput. Combining these together, a 1.93× area efficiency is achieved by WDM. The reason for this extremely low area overhead of WDM is that the Fourier lens and the photodetectors can be shared, which together consume a large proportion of the total system area.

5 REFOCUS ARCHITECTURE

The high-level architecture and configuration of ReFOCUS are introduced first in this section, followed by optimizations and design choices.

5.1 Overall architecture

ReFOCUS has two versions, ReFOCUS-FF (feedforward) and ReFOCUS-FB (feedback). The difference between the two versions is at the RFCU level - ReFOCUS-FB reuses inputs 15 times while ReFOCUS-FF reuses inputs once, and both versions share the same high-level architecture. The architecture diagram of ReFOCUS is illustrated in Figure 6. ReFOCUS operates at 10 GHz, supports 8-bit precision, and assumes monolithic integration of CMOS and photonics [47]. There are 16 RFCUs within ReFOCUS, with each RFCU containing 256 input waveguides and processing two wavelengths concurrently through WDM. Input signals first pass through optical buffers and then broadcast to all RFCUs, while weights are generated within each RFCU. ReFOCUS adopts 16-cycle temporal accumulation to reduce the frequency of ADC and the output processing CMOS circuits to 625 MHz. On the CMOS part, each RFCU has two corresponding CMOS processing units that are used to generate the inputs, process the outputs (reading from ADC, scaling and accumulating the results, and implementing the ReLU non-linearity), and communicate with memory. ReFOCUS has a 4MB global activation SRAM shared with all RFCUs, while each RFCU has its own 512 KB weight SRAM. Input and output data buffers are added to reduce the access energy of the shared activation SRAM. The design choices such as dataflow, number of RFCUs, data buffer configuration, and delay line lengths, will be further discussed in this section.

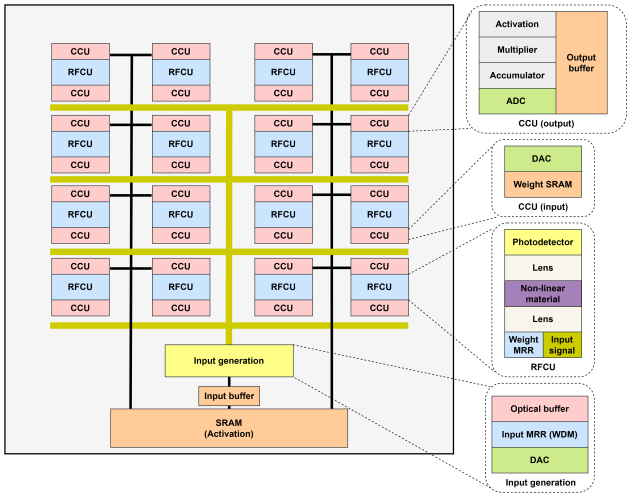

Figure 6: High-level architecture diagram of ReFOCUS. CCU stands for CMOS compute unit. Each RFCU has two CCUs, one for input generation and the other for output processing. The diagram is not drawn to scale.

5.2 Memory hierarchy

ReFOCUS adopts a similar top-level memory configuration as [32], with a 4MB shared activation SRAM and separate local weight SRAMs (one for each RFCU with 512KB size). he activation and weight SRAM sizes are configured to hold the entire activation/layer of weights of common CNNs [23, 54] to eliminate the need for writing to DRAMs during execution. This relatively large SRAM size also results in > 4× access energy compared to weight SRAM. Directly accessing and storing from/to the shared activation SRAM as [32] leads to excessive SRAM power, as shown in Figure 3 (a). In ReFOCUS, we add input and output data buffers to reduce the

memory access power. All RFCUs share a single input buffer because of input broadcasting, while each RFCU has its own output buffers. The size and relative access energy of the data buffers depends on the dataflow, and is further discussed in Section 5.3.

5.3 Dataflow

5.3.1 Parallelization scheme. Input broadcasting the default parallelization scheme in ReFOCUS, and the main reason is to reduce the input DAC power. Output reuse is achieved through temporal accumulation while weight reuse is inherently achieved by the JTC operation (kernel is 'broadcasted' to the entire input tile). Therefore, inputs are broadcasted to all RFCUs to achieve input reuse. Within an RFCU, WDM is implemented to compute two input channels in parallel, for reasons discussed in Section 4.2.

5.3.2 Alternating dataflow. Dataflow plays a critical role in ReFOCUS, as many optimizations have constraints or requirements related to dataflow. Temporal accumulation, which reduces ADC frequency and power, requires an OS dataflow to accumulate the output of different convolution channels using the photodetector. However, the optical buffers, which optically reuse the inputs, enforce an IS dataflow. While the two dataflow seem contradicting, they can be combined together to form an alternating dataflow with some modifications on the optical buffer.

The solution is to increase the delay line length so that inputs will be delayed by M cycles before being reused. Within the M cycles, there are no restrictions on the exact dataflow, and OS dataflow can be used to implement temporal accumulation for M cycles, by processing an input channel group of M channels. After M cycles, the same input channel group is reused, and another filter needs to be processed to achieve input reuse. The dataflow is illustrated in detail in Figure 7, which shows the dataflow of an example system with WDM and feedforward optical buffer with $M = 4$. Each RFCU processes a unique filter, and for the 8-RFCU system in this example, 8 filters will be processed in parallel, therefore when the input channel group is reused in RFCU1, filter 9 is processed. Within an RFCU, spatial accumulation is achieved by WDM, where each wavelength processes a different channel group of a filter. With this OS-IS alternating dataflow, output reuse (through temporal accumulation) and input reuse (through optical buffer) can be achieved concurrently.

5.3.3 Optimizing for efficient memory accesses. The input channel group can only be reused a limited number of times (reuse once for the ReFOCUS-FF). Thus, after reuse completes and new inputs need to be generated, there is a choice of what should be processed next. There are two dataflow choices: (1) follow the current pattern to process another filter until all filters are processed for the current input channel group, as illustrated in Figure 7 and (2) process another input channel group of the first filter being processed in the RFCU, until all the channels are fully processed for the current filters being processed. These two dataflow choices have different impacts on the SRAM data buffer design and the overall power efficiency. (1) requires relatively small input buffers and large output buffers while (2) requires relatively large input buffers and small output buffers.

Table 3: Notations and definitions of common terms used in the analysis.

Notation	Definition
M	Delay line length in terms of cycles
R	How many times the signal is reused
N_{RFCU}	Number of RFCUs
T	Input tile size (number of input waveguides)
N_λ	Number of waveguides.

Some common notations and their definitions used in the analysis are listed in table 3. For case (1), the input and output buffer size (per RFCU) in bytes can be calculated as (ignore ping-pong buffer for now):

$$B_{in1} = T \times M \times N_\lambda, B_{out1} = T \times \frac{N_F}{N_{RFCU}}$$

, where N_F is the maximum number of filters per layer of a neural network. For case (2), the input and output buffer size can be calculated as:

$$B_{in2} = T \times N_C \times N_\lambda, B_{out2} = T \times (R + 1)$$

, where N_C is the maximum number of channels per layer of a neural network. In ReFOCUS, we adopt (1) as our dataflow, which favors the input buffer over the output buffer. The reason behind this design choice is the input buffer needs to have a higher frequency than the output buffer and hence has higher constraints on access latency. The input buffer needs to be accessed every cycle (although when input is being reused the input buffer will not be accessed at all), while the output buffer only needs to be accessed once per M cycle. A large input buffer may not meet the latency requirement. Besides, for ReFOCUS-FF, the input buffer has more accesses overall compared to the output buffer, as there is more output reuse than input reuse (discussed more in Section 5.4). Thus, having a smaller input buffer reduces the cost of input buffer accesses, and improves the overall power efficiency of the ReFOCUS-FF.

5.4 Choice of design parameter

Some of the design choices such as which signal to reuse, how WDM is implemented, and the number of wavelengths used are already discussed in Section 4. This section discusses other design choices that cannot be determined individually as they have dependence or impact on each other and require system-level analysis, such as delay line length, number of RFCUs, and how many times the inputs should be optically reused in ReFOCUS-FB.

5.4.1 ReFOCUS-FF. From Equation 4, the Y-junction split ratio α for the feedforward buffer can be computed. Based on α, it can be derived that the average laser power needs to be $1/2\alpha \times$ larger (divided by 2 because the light is reused once). Based on the delay line loss from Table 1, and the fact that laser power per channel is much smaller than the DAC power, the increase in laser power caused by a longer delay line will have a negligible impact on overall power efficiency for any reasonable delay line lengths. Therefore, the primary overhead of longer delay lines is the increase in area.

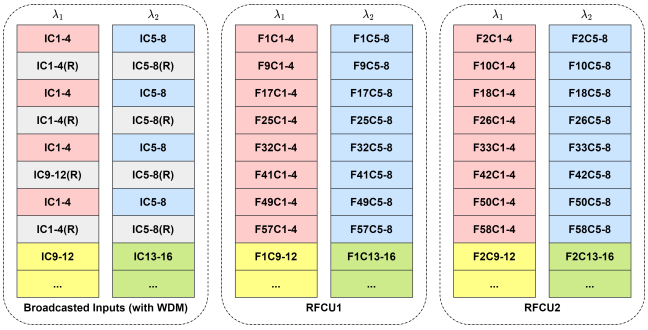

Figure 7: Dataflow used in ReFOCUS. An 8-RFCU system that implements feedforward optical buffers with 4-cycle delay lines and WDM with 2 wavelengths is assumed for this example. IC($a-b$) refers to input channel $a-b$, F(x)C($a-b$) refers to channel $a-b$ of filter x. λ_i refers to the i^{th} wavelength in WDM. R means reused input signal through the optical buffer. Difference channel groups are marked with different colors.

A longer delay line can result in fewer RFCUs that can be placed within a given area limit.

Nearly all previous studies on on-chip photonic neural network accelerators have reported chip areas of less than $200mm^2$ [32, 36, 52, 56, 71]. Increasing the chip area can lead to yield and cost issues, while providing diminishing returns in terms of performance. Therefore, we set the area budget of the photonic components of ReFOCUS to be $150mm^2$ (leaving some margin for CMOS components), and calculate the maximum number of RFCUs that can be placed for various delay line sizes within the area budget. Since optical buffer has impacts on both power and area, we develop a custom efficiency metric to take into account both power efficiency and area efficiency. The metric is simply the product of frames per second per watt and frames per second per mm^2, and is named PAP (power-efficiency-area-efficiency-product). The geo-mean of relative PAP on four different CNNs (VGG-16, ResNet-18, ResNet-34, ResNet-50) is calculated to determine the optimal delay line length and number of RFCUs, and the results are shown in Table 4, along with relative FPS/W and FPS/mm^2,. The change in laser power is modeled in the calculation. The results suggest that when the signals can be delayed by 16 cycles, the optimal efficiency can be achieved, with 18 RFCUs. Thus, we configure ReFOCUS-FF to have 16 RFCUs. We choose 16 rather than 18 as 16 is a power-of-two value and fits better with neural network execution.

5.4.2 ReFOCUS-FB. There is an additional design choice for ReFOCUS-FB, which is how many times the inputs are reused before generating new ones (R). The choice of R solely depends on the signal loss of the optical buffer, and the related change in average laser power and dynamic range of reused signals (ratio of the initial signal power and the power of the last reused signal).

Unlike ReFOCUS-FF, laser power overhead is not trivial without optimizations for ReFOCUS-FB, even with a low delay line loss. Since the signal power will be smaller for each reuse iteration due to the Y-junction, a relatively large initial laser power is required to make sure the last reused signal (the one with the lowest power) is detectable by the photodetector. In this scheme, all signals except for the last reused signal have higher than the required signal power, which makes the average laser power much higher than the case without optical buffers, especially when the split ratio α is 50%. The average laser power overhead and the dynamic range can be calculated based on Equation 3 and are reported in Table 5 for different number of reuses and α. Without optimizing the α, reuse 7 or more times is infeasible as it can lead to > 38× average laser power and > 153× dynamic range. Even ignoring the laser power overhead, the dynamic range is too large for an 8-bit ADC which has just 256 levels.

However, this issue can be resolved by setting α to $1/(R+1)$, the optimal Y-junction split ratio for the feedback optical buffer. As shown in Table 5, the relative laser power and the dynamic range are both 3.05 for reusing 7 times. Therefore, significantly more optical reuse can be achieved with this modification. If only power-of-2 values are considered (to fit the structure of CNNs better), reusing the signal higher than 15 leads to diminishing returns on overall power efficiency, while increasing the dynamic range of the signal. Thus, ReFOCUS-FB reuses the input signals optically 15 times, to achieve a balance between power efficiency and effective output precision. Once R is determined, the delay line length (M) and the number of RFCUs can be decided in the same way as ReFOCUS-FF, and the results are shown in Table 4. The optimal choices of M and N_{RFCU} are the same as ReFOCUS-FF, thus these two designs share the same system architecture.

Table 4: Number of RFCUs can be placed and relative FPS/W, FPS/mm^2, PAP for different delay line lengths in terms of cycles, for both ReFOCUS-FF and ReFOCUS-FB. The absolute values are shown for the baseline case where $M = 1$.

M	1	2	4	8	16	32
N_{RFCU}	25	24	23	21	18	11
FPS/W (FF)	1 (237)	1.92	2.83	3.71	4.51	4.72
FPS/mm^2 (FF)	1 (196)	1.00	0.97	0.91	0.80	0.53
PAP (FF)	1 (4.6e4)	1.92	2.75	3.39	**3.61**	2.52
FPS/W (FB)	1 (247)	2.00	3.07	4.18	5.20	5.17
FPS/mm^2 (FB)	1 (196)	0.99	0.96	0.91	0.80	0.53
PAP (FB)	1 (4.8e4)	1.98	2.96	3.80	**4.14**	2.75

Table 5: Relative laser power when compared to the system without optical buffer and the dynamic range of input signals for different R and α.

R	1	3	7	15	31	63
$\alpha = 1/(R+1)$						
relative LP.	2.05	2.56	3.05	3.87	5.96	13.7
dynamic range	2.05	2.56	3.05	3.87	5.96	13.7
$\alpha = 0.5$						
relative LP.	2.05	4.32	38.4	6.0e3	3.0e8	1.5e18
dynamic range	2.05	8.64	153	4.8e4	4.8e9	4.7e19

6 EVALUATION

We employ Cadence Genus along with a commercial 14nm library to model the power and area of the CMOS components. We use CACTI [43] to model the area, leakage power, and access energy for all the SRAM memory and buffers used in ReFOCUS. We develop a custom simulator based on Python to simulate the throughput and energy consumption of ReFOCUS on CNN inferences, and also model the area of ReFOCUS. The simulator integrates the CMOS and SRAM simulation results and then models the photonic part based on the characteristics of photonic components used in ReFOCUS. Table 6 lists the power and area of the components used in ReFOCUS. Since there are no reported ADCs/DACs that have the exact same specifications as we assumed in ReFOCUS, we find 8-bit, 14 nm ADC and DAC, with higher frequency than ReFOCUS required, and then linear scale down the power accordingly frequency, which is a conservative approach as the relationship between frequency and power is not linear. The average DAC power is calculated by multiplying the power reported in [35] with the duty cycle of DAC in ReFOCUS. Since JTC-based system can only process positive weights, ReFOCUS implements pseudo-negative processing, which splits a filter into two parts, one positive and one negative. The negative part is processed as a positive filter and the results are subtracted from the results of the positive part digitally. This approach addresses the positive-weight limitation, but doubles inference latency. We only benchmark the convolution layers of these networks, which correspond to more than 99% of total computation.

Table 6: Power of active components and the area of photonic components used in ReFOCUS.

Component power (mW)	
MRR	0.42 [42]
Laser (min) *	0.1 per waveguide
ADC @ 625 MHz	0.93 [35]
DAC @ 10 GHz	35.71 [7]
Optical component area (μm^2)	
MRR	255 [32]
Photodetector	1920 [32]
Y-junction	2.6 [69]
Laser	1.2e5 [13]
Delay line (0.1 ns delay)	1e4
Lens	2e6

*: Minimum laser power required. The average laser power will be higher to compensate for the loss of optical buffers.

6.1 Power and area

For power evaluation, we benchmark ReFOCUS on 5 CNNs (AlexNet [27], VGG-16 [54], ResNet-18,34,50 [23]), and the average system power is calculated. Overall, ReFOCUS-FF and ReFOCUS-FB consume 14.0W and 10.8W average power respectively. The difference is caused by the further reduction of input DAC energy, as ReFOCUS-FB has more optical reuse. Figure 8 shows the power breakdown of ReFOCUS-FF and ReFOCUS-FB. In both systems, DAC still consumes the most power, but the proportion is reduced in the FB version. For the FB version, DAC power is dominated by weight DAC, which consumes 90% of total DAC power, preventing further reduction of DAC power through input reuse. As a result of computation becoming more efficient, SRAM access energy consumes a large proportion of total power in both cases, which would be even larger without data buffers. ReFOCUS-FB has significantly higher laser power compared to ReFOCUS-FF, as the laser power needs to be scaled to compensate for the loss of the feedback optical buffer. Further improving the system power requires reducing the weight DAC power, and we will briefly discuss this in Section 7.3.

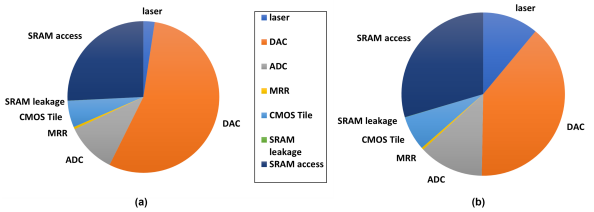

Figure 8: (a): Power breakdown of ReFOCUS-FF. (b): Power breakdown of ReFOCUS-FB. The same legend is applied to both pie charts.

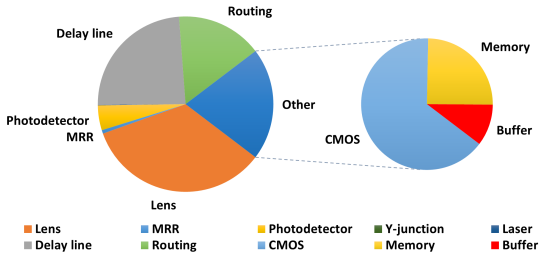

Figure 9: Area breakdown of ReFOCUS. The secondary pie chart shows the area breakdown of non-photonic components (CMOS, SRAM memory, and data buffers).

Figure 9 shows the area breakdown of ReFOCUS, which is applicable to both versions of ReFOCUS as they have the same area. ReFOCUS has a 171.1 mm^2 overall area, with 135.7 mm^2 contributed by the photonic components. SRAM memory and data buffers together consume 12.4 mm^2 area, and the rest chip area is contributed by CMOS logic and ADCs/DACs. On the photonic side, lenses (58.5 mm^2) and delay lines (41.0 mm^2) are the two largest contributors. WDM reduces the lens area of ReFOCUS by 2×, making it possible to fit 256 16-cycle delay lines with no area overhead. Further increasing the delay line length will make its area overhead too large to be compensated, and leads to lower system efficiency.

6.2 Effect of optimizations

Table 7 shows the potential reuse (spatial and temporal) that can be achieved through different optimizations for the baseline system and two versions of ReFOCUS. With the proposed WDM and

optical buffer, both outputs and inputs can be further reused when compared to the baseline, hence reducing the conversion energy.

Table 7: Potential reuse can be achieved by different optimizations. OB stands for optical buffer and TA stands for temporal accumulation.

	Input reuse		Output reuse	
	Broadcast	OB	WDM	TA
Baseline	16 ×	N/A	N/A	16×
ReFOCUS-FF	16 ×	2×	2×	16×
ReFOCUS-FB	16×	16×	2×	16×

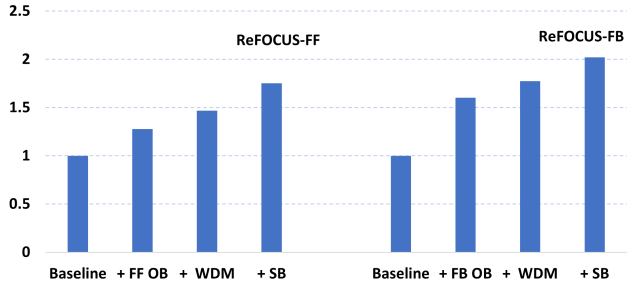

Figure 10: Relative FPS/W for ReFOCUS with different optimizations. Each column includes the optimizations that are reported on its left. Resnet-34 is used for this benchmark. OB stands for optical buffer while SB stands for SRAM buffer.

Figure 10 shows the relative FPS/W on ResNet-34[23] of ReFOCUS with different optimizations enabled compared to ReFOCUS-baseline with the same architecture (similar to [32]). All three optimizations proposed (optical buffers, WDM, and SRAM data buffers) improve the overall power efficiency noticeably. The SRAM buffers provide substantial benefits because the power of ADCs/DACs is optimized by optical buffers, WDM, and temporal accumulation, making SRAM power consume a larger proportion of total system power (36.9% for ReFOCUS-FB without data buffers). When comparing to the baseline system that scaled to the same throughput (with a much larger area), the absolute power of converters (ADC + DAC) for ReFOCUS-FB is 1.72× smaller, demonstrating the effectiveness of optical reuse to reduce the power overhead of A/D and D/A conversions. Overall, both ReFOCUS versions achieve significant performance improvement compared to the baseline system, with ReFOCUS-FB being 2× more efficient.

6.3 Comparison with prior work

We primarily compare ReFOCUS with PhotoFourier for two reasons - (1) PhotoFourier is the most closely related prior work as both PhotoFourier and ReFOCUS are based on JTC, and (2) PhotoFourier reports state-of-the-art efficiency results for on-chip photonic neural network accelerators. To make the comparison as fair as possible, we obtain the simulator from the authors of PhotoFourier, and implement a slightly modified version of PhotoFourier for comparison,

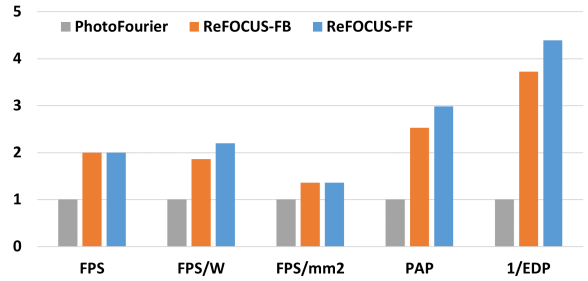

Figure 11: Two ReFOCUS versions compared to PhotoFourier, in terms of relative FPS, FPS/W, FPS/mm^2, PAP, and inverse of EDP. Benchmarked on 5 CNNs.

which uses our power and area number for individual components and adopts non-linear material for optical nonlinearity. We evaluate the systems on the 5 CNNs mentioned earlier, and the geometric mean of key metrics is calculated.

Figure 11 shows the relative improvements of ReFOCUS over PhotoFourier, in terms of throughput (FPS), power efficiency (FPS/W), area efficiency (FPS/mm^2), and two combined efficiency metrics - PAP (introduced earlier) and 1/EDP (inverse of energy-delay product). Both ReFOCUS-FF and ReFOCUS-FB achieve better results on all metrics compared to PhotoFourier, demonstrating the efficiency of ReFOCUS. The FPS of ReFOCUS is roughly doubled since ReFOCUS processes two wavelengths concurrently in each RFCU. For the same reason, ReFOCUS achieves better area efficiency even though delay lines add a large area overhead. Energy-wise, ReFOCUS-FB achieves more than 2× FPS/W compared to PhotoFourier, thanks to the extra input and output reuse achieved through the optical buffer and WDM. ReFOCUS-FF also has close to 2× efficiency. Since ReFOCUS has higher throughput, power, and area efficiency, naturally, ReFOCUS achieves significantly better PAP and 1/EDP.

We also compare ReFOCUS with two other 8-bit precision photonic neural network accelerators, Albireo [52] and Holylight-m [36] in terms of FPS and FPS/W on AlexNet, VGG-16, and ResNet-18. For reference purposes, we further compare ReFOCUS with a digital accelerator (UNPU) [29] and one RRAM-based accelerator [62]. The results are shown in Figure 13, some results are missing as some works did not report results on all three networks. Similarly, ReFOCUS achieves the best results on both metrics. ReFOCUS achieves up to 25× power efficiency compared to state-of-the-art MZI/MRR-based photonic neural network accelerator Albireo, and achieves up to 145× power efficiency compared to Holylight-m. The large performance gap between Fourier-optics based accelerators such as ReFOCUS and PhotoFourier and the MZI/MRR style accelerators demonstrates the superiority of Fourier-optics on CNN acceleration.

To better demonstrate the advantage of ReFOCUS, we conduct a comparison with some well-known digital accelerators from both industry and academia, namely, the NVIDIA H100 GPU [3], Google TPU V3 [1], Simba [51], and a design from JSSC 20 [70], on the relatively large ResNet-50 network. The FPS results of H100 and TPU V3 are collected from the MLPerf benchmark [48]. Figure 12 illustrates the FPS and FPS/W results. While H100 and TPU V3

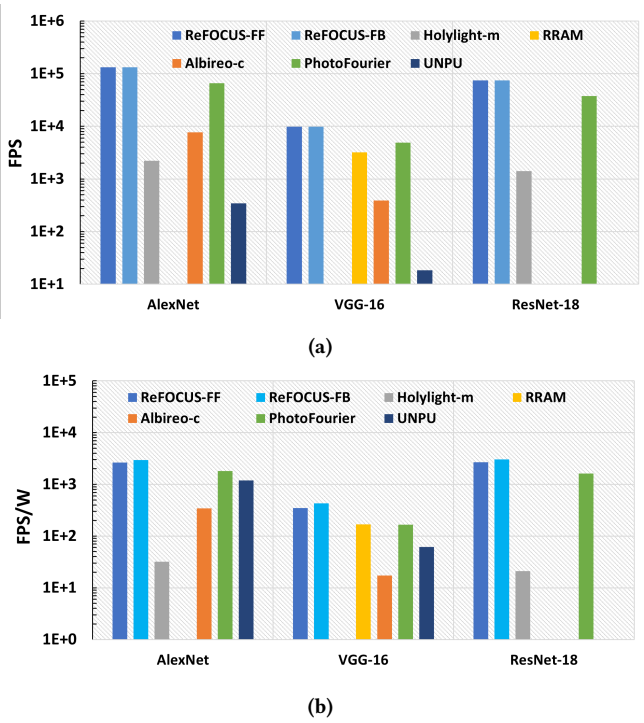

Figure 12: ReFOCUS compared with other accelerators. The logarithmic axis is used. (a): FPS. (b): FPS/W.

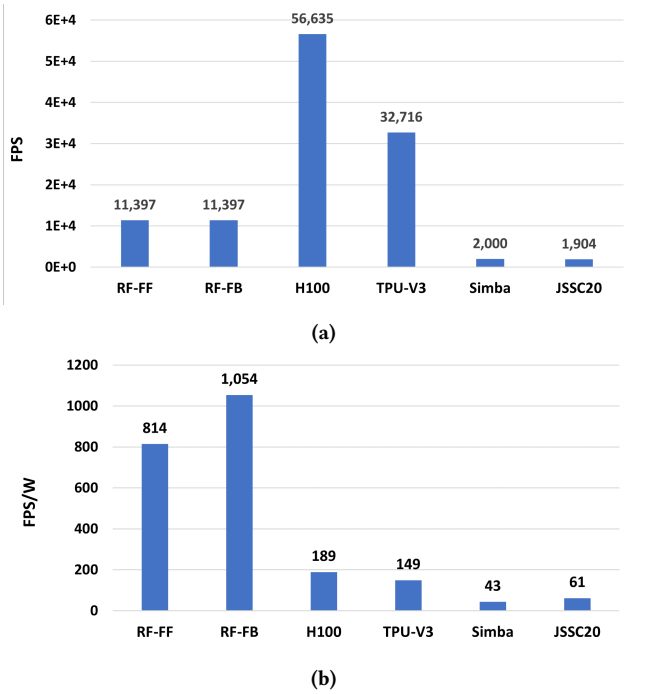

Figure 13: ReFOCUS compared with different digital accelerators on ResNet-50. RF stands for ReFOCUS. (a) FPS. (b) FPS/W.

exhibit better raw throughput compared to ReFOCUS, it is essential to consider their significantly larger footprint as a contributing factor. However, in terms of power efficiency (FPS/W), ReFOCUS has a clear advantage over existing GPUs and ASIC accelerators, boasting an efficiency that is 5.6 − 24.5 times higher.

The efficiency advantage over digital and other photonic accelerators mainly stems from the complexity reduction of JTC, the passive calculation of Fourier transforms, and the reduced conversion cost due to the reusing of light signals. When compared to RRAM-based analog accelerators that have limited write endurance, high write latency/energy, and are usually network-specific due to the necessity to unroll the network into numerous fixed crossbar arrays, ReFOCUS presents much better programmability and flexibility while still having more than 2× efficiency. Rather than being network-specific like RRAM-based accelerators, ReFOCUS allows weights to be fully programmable at high speed during runtime, akin to digital accelerators.

7 DISCUSSION

7.1 Instruction scheduling

While optical buffers introduce complexity to the system and dataflow, potentially complicating scheduling, the buffer size (delay line length) and latency in ReFOCUS are fixed. Given the strictly first-in-first-out behavior of the optical buffer, its behavior can be predetermined, allowing scheduling to be offloaded to the compiler. Consequently, the compiler can manage the instruction scheduling statically, akin to Very Long Instruction Word (VLIW).

7.2 Compensating system noise

Inherent in analog computing, noise and non-idealities cannot be entirely avoided in photonic neural network accelerators. However, system noise can be mitigated through careful design, placement, and calibration of photonic components. Moreover, the noise impact can be further compensated by modeling and injecting noise during training. This approach enables the trained neural network to learn and adapt to various noise behaviors and non-idealities.

7.3 DRAM, weight sharing, and weight DAC

Almost all prior works on photonic neural network accelerators did not report DRAM energy, which is often a major contributor to system power. We discover that when the computation and on-chip memory access are efficient enough, DRAM access power cannot be ignored. For example, DRAM access power can contribute more than 50% of total power in ReFOCUS-FB, when profiled with HBM2 access energy [44]. For neural network layers with small activation sizes but a large number of filters, DRAM energy dominates, even though ReFOCUS already minimizes DRAM accesses (no DRAM writes). Without reducing DRAM access energy, further optimizing computation or on-chip memory access leads to diminishing returns. Besides developing better DRAM technology (e.g., HBM3), there are also software solutions to reduce DRAM access energy, such as weight sharing.

Neural network weight sharing: Weight sharing [15, 16, 31, 55, 61, 65] is an effective compression technique for neural networks

that outperforms quantization and pruning while maintaining accuracy. It uses a smaller codebook and index matrix, reducing storage needs. In CNNs, various weight sharing methods exist [31, 55, 65]. Sharing 2D convolution kernels [55] with a trainable scaling factor can achieve a 4.5× compression ratio compared to 8-bit weights in ReFOCUS, with negligible accuracy loss. This method reduces DRAM access energy by 4.5× and overall energy by up to 52%. Weight SRAM access energy is also lowered due to smaller weight memory.

Channel Reordering: In ReFOCUS-FB and ReFOCUS-FF, the weight DAC accounts for 90% and 53% of the DAC power consumption, and 42% and 31% of the total system power on ResNet-34, respectively. Weight sharing in 2D convolution kernels presents an opportunity to decrease weight DAC power and thereby enhance system efficiency. To capitalize on this, we reorder the input channels and group those that are assigned to the same kernel. This minimizes the weight DAC operations, although the degree of reduction is constrained by factors like input broadcasting and reuse. We further introduce a Simulated Annealing-based algorithm for channel reordering, achieving a 15% reduction in weight DAC power for ReFOCUS-FF under a typical setup and boosting the overall power efficiency by 4.7%.

7.4 Non-CNN tasks

While we primarily focus on accelerating CNNs in this work, recently there are many works proposed Fourier-transform based transformer [21, 30] and convolution-based transformer [64, 68], which can be potentially accelerated by JTC-based systems as they share similar underlying operations as CNNs. Further work is required to adapt JTC-based architecture for these transformer models, which will be a part of our future work.

7.5 Slow light

One concept that has been used to design area-efficient optical delay lines is called 'slow light'. The speed of light is significantly reduced as it propagates through a medium in this type of delay line, achieved by manipulating the properties of the medium. With a lower light speed, the length of the waveguide, and hence the delay line area can be greatly reduced. There are works that reported 'slow light' based delay lines with promising area efficiency [9, 66]. Given the number of cycles that inputs can be delayed is constrained by the delay line area, having more compact delay lines will further improve the system efficiency. Slow slight-based delay lines are not used in ReFOCUS as they currently have relatively large loss [9] and require further development.

8 RELATED WORK

As mentioned in Section 1, on-chip photonic neural network accelerators can be roughly split into two categories - dot product or matrix multiplication accelerators based on MRRs/MZIs, and convolution accelerators based on Fourier-optics. PhotoFourier [32], being the only published Fourier-optics based accelerator so far, is the most closely related prior work. Hence, we extensively compare ReFOCUS to PhotoFourier. PhotoFourier proposed the first on-chip JTC based neural network accelerator and demonstrated state-of-the-art power efficiency. It uses plain JTCs as building blocks that do not feature WDM or optical buffer and the performance advantage mostly comes from the complexity reduction of JTC. In contrast, ReFOCUS innovatively integrates two versions of optical buffers and WDM. This distinct approach substantially enhances both the area and power efficiency of JTC-based accelerators, thus differentiating ReFOCUS from PhotoFourier. Besides, ReFOCUS further optimizes the dataflow and memory hierarchy to improve power efficiency. Other on-chip photonic neural network accelerators [36, 39, 52, 53, 56, 71] are fundamentally different than ReFOCUS as ReFOCUS leverages the convolution theorem to reduce the complexity of CNNs through Fourier optics.

9 CONCLUSION

In this paper, we introduce ReFOCUS, a Fourier-optics on-chip photonic neural network accelerator featuring optical reuse. We present two innovative optical buffer designs tailored to enhance light reuse and energy efficiency. To mitigate the area overhead of optical buffers, we incorporate WDM in ReFOCUS, significantly improving the area efficiency of the system. Compared to state-of-the-art photonic neural network accelerators, ReFOCUS demonstrates remarkable gains: 2× throughput, 2.2× energy efficiency, and 1.36× area efficiency. Furthermore, ReFOCUS achieves over 25× power efficiency when compared to photonic neural network accelerators not utilizing Fourier optics, highlighting its potential for future high-performance computer-vision applications.

ACKNOWLEDGMENTS

We would like to extend our gratitude to the Office of Naval Research (ONR) for providing the funding for this research.

REFERENCES

[1] 2023. *Google Cloud TPU.* https://cloud.google.com/tpu/docs/system-architecture-tpu-vm
[2] 2023. *ImageNet Benchmark.* https://paperswithcode.com/sota/image-classification-on-imagenet
[3] 2023. *NVIDIA H100 Tensor Core GPU.* https://www.nvidia.com/en-us/data-center/h100
[4] M Zahirul Alam, Israel De Leon, and Robert W Boyd. 2016. Large optical nonlinearity of indium tin oxide in its epsilon-near-zero region. *Science* 352, 6287 (2016), 795–797.
[5] Viraj Bangari, Bicky A. Marquez, Heidi Miller, Alexander N. Tait, Mitchell A. Nahmias, Thomas Ferreira de Lima, Hsuan-Tung Peng, Paul R. Prucnal, and Bhavin J. Shastri. 2020. Digital Electronics and Analog Photonics for Convolutional Neural Networks (DEAP-CNNs). *IEEE Journal of Selected Topics in Quantum Electronics* 26, 1 (2020), 1–13. https://doi.org/10.1109/JSTQE.2019.2945540
[6] Qiaoliang Bao, Jianqiang Chen, Yuanjiang Xiang, Kai Zhang, Shaojuan Li, Xiaofang Jiang, Qing-Hua Xu, Kian Ping Loh, and T Venkatesan. 2015. Graphene nanobubbles: a new optical nonlinear material. *Advanced Optical Materials* 3, 6 (2015), 744–749.
[7] Pietro Caragiulo, Oscar Elisio Mattia, Amin Arbabian, and Boris Murmann. 2020. A Compact 14 GS/s 8-Bit Switched-Capacitor DAC in 16 nm FinFET CMOS. In *2020 IEEE Symposium on VLSI Circuits*. 1–2. https://doi.org/10.1109/VLSICircuits18222.2020.9162776
[8] Julie Chang, Vincent Sitzmann, Xiong Dun, Wolfgang Heidrich, and Gordon Wetzstein. 2018. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Scientific reports* 8, 1 (2018), 1–10.
[9] Alexander Chen, Amir Begović, Stephen Anderson, and Zhaoran Rena Huang. 2022. On-Chip Slow-Light SiN Bragg Grating Waveguides. *IEEE Photonics Journal* 14, 6 (2022), 1–6. https://doi.org/10.1109/JPHOT.2022.3220540
[10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *International conference on machine learning*. PMLR, 1691–1703.
[11] Yu-Hsin Chen, Joel Emer, and Vivienne Sze. 2016. Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks. In *2016*

ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA). 367–379. https://doi.org/10.1109/ISCA.2016.40

[12] Shane Colburn, Yi Chu, Eli Shilzerman, and Arka Majumdar. 2019. Optical frontend for a convolutional neural network. *Applied optics* 58, 12 (2019), 3179–3186.

[13] A Descos, C Jany, D Bordel, H Duprez, G Beninca de Farias, P Brianceau, S Menezo, and B Ben Bakir. 2013. Heterogeneously integrated III-V/Si distributed Bragg reflector laser with adiabatic coupling. In *39th European Conference and Exhibition on Optical Communication (ECOC 2013)*. IET, 1–3.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[15] Etienne Dupuis, David Novo, Ian O'Connor, and Alberto Bosio. 2022. A Heuristic Exploration of Retraining-free Weight-Sharing for CNN Compression. In *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*. 134–139. https://doi.org/10.1109/ASP-DAC52403.2022.9712487

[16] Etienne Dupuis, David Novo, Ian O'Connor, and Alberto Bosio. 2020. On the Automatic Exploration of Weight Sharing for Deep Neural Network Compression. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1319–1322. https://doi.org/10.23919/DATE48585.2020.9116350

[17] Jonathan K George, Maria Solyanik-Gorgone, Hangbo Yang, Chee Wei Wong, and Volker J Sorger. 2022. Nonlinear Optical Joint Transform Correlator for Low Latency Convolution Operations. *arXiv preprint arXiv:2202.06444* (2022).

[18] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR* abs/1311.2524 (2013). arXiv:1311.2524 http://arxiv.org/abs/1311.2524

[19] J.W. Goodman. 2005. *Introduction to Fourier Optics*. W. H. Freeman. https://books.google.com/books?id=ow5xs_Rtt9AC

[20] Jiaqi Gu, Zheng Zhao, Chenghao Feng, Mingjie Liu, Ray T. Chen, and David Z. Pan. 2020. Towards Area-Efficient Optical Neural Networks: An FFT-based Architecture. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. 476–481. https://doi.org/10.1109/ASP-DAC47756.2020.9045156

[21] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. 2021. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587* (2021).

[22] Puneet Gupta and Shurui Li. 2022. 4F optical neural network acceleration: an architecture perspective. In *Proc. of SPIE Vol*, Vol. 12019. 120190B–1.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[24] Zibo Hu, Shurui Li, Russell LT Schwartz, Maria Solyanik-Gorgone, Mario Miscuglio, Puneet Gupta, and Volker J Sorger. 2022. High-Throughput Multichannel Parallelized Diffraction Convolutional Neural Network Accelerator. *Laser & Photonics Reviews* (2022), 2200213.

[25] Bahram Javidi. 1990. Comparison of nonlinear joint transform correlator and nonlinearly transformed matched filter based correlator for noisy input scenes. *Optical Engineering* 29, 9 (1990), 1013–1020.

[26] Hojoong Jung and Hong X. Tang. 2016. Aluminum nitride as nonlinear optical material for on-chip frequency comb generation and frequency conversion. *Nanophotonics* 5, 2 (2016), 263–271. https://doi.org/doi:10.1515/nanoph-2016-0020

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[28] Hansuek Lee, Tong Chen, Jiang Li, Oskar Painter, and Kerry J Vahala. 2012. Ultra-low-loss optical delay line on a silicon chip. *Nature communications* 3, 1 (2012), 867.

[29] Jinmook Lee, Changhyeon Kim, Sanghoon Kang, Dongjoo Shin, Sangyeob Kim, and Hoi-Jun Yoo. 2019. UNPU: An Energy-Efficient Deep Neural Network Accelerator With Fully Variable Weight Bit Precision. *IEEE Journal of Solid-State Circuits* 54, 1 (2019), 173–185. https://doi.org/10.1109/JSSC.2018.2865489

[30] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2021. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824* (2021).

[31] Shurui Li and Puneet Gupta. 2022. Bit-serial Weight Pools: Compression and Arbitrary Precision Execution of Neural Networks on Resource Constrained Processors. *Proceedings of Machine Learning and Systems* 4 (2022), 238–250.

[32] Shurui Li, Hangbo Yang, Chee Wei Wong, Volker J Sorger, and Puneet Gupta. 2023. Photofourier: A photonic joint transform correlator-based neural network accelerator. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 15–28.

[33] Yuan Li, Ahmed Louri, and Avinash Karanth. 2022. SPACX: Silicon Photonics-based Scalable Chiplet Accelerator for DNN Inference. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 831–845. https://doi.org/10.1109/HPCA53966.2022.00066

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

[35] Juzheng Liu, Mohsen Hassanpourghadi, and Mike Shuo-Wei Chen. 2022. A 10GS/s 8b 25fJ/c-s 2850um² Two-Step Time-Domain ADC Using Delay-Tracking Pipelined-SAR TDC with 500fs Time Step in 14nm CMOS Technology. In *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, Vol. 65. 160–162. https://doi.org/10.1109/ISSCC42614.2022.9731625

[36] Weichen Liu, Wenyang Liu, Yichen Ye, Qian Lou, Yiyuan Xie, and Lei Jiang. 2019. Holylight: A nanophotonic accelerator for deep learning in data centers. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1483–1488.

[37] Chye-Hwa Chye Hwa Loo and Mohammad S Alam. 2004. Invariant object tracking using fringe-adjusted joint transform correlator. *Optical Engineering* 43, 9 (2004), 2175–2183.

[38] Tao Luo, Shaoli Liu, Ling Li, Yuqing Wang, Shijin Zhang, Tianshi Chen, Zhiwei Xu, Olivier Temam, and Yunji Chen. 2017. DaDianNao: A Neural Network Supercomputer. *IEEE Trans. Comput.* 66, 1 (2017), 73–88. https://doi.org/10.1109/TC.2016.2574353

[39] Armin Mehrabian, Yousra Al-Kabani, Volker J Sorger, and Tarek El-Ghazawi. 2018. PCNNA: a photonic convolutional neural network accelerator. In *2018 31st IEEE International System-on-Chip Conference (SOCC)*. IEEE, 169–173.

[40] Mario Miscuglio, Zibo Hu, Shurui Li, Jonathan K. George, Roberto Capanna, Hamed Dalir, Philippe M. Bardet, Puneet Gupta, and Volker J. Sorger. 2020. Massively parallel amplitude-only Fourier neural network. *Optica* 7, 12 (Dec 2020), 1812–1819. https://doi.org/10.1364/OPTICA.408659

[41] Mario Miscuglio, Armin Mehrabian, Zibo Hu, Shaimaa I. Azzam, Jonathan George, Alexander V. Kildishev, Matthew Pelton, and Volker J. Sorger. 2018. All-optical nonlinear activation function for photonic neural networks]. *Opt. Mater. Express* 8, 12 (Dec 2018), 3851–3863. https://doi.org/10.1364/OME.8.003851

[42] Sajjad Moazeni, Sen Lin, Mark Wade, Luca Alloatti, Rajeev J Ram, Miloš Popović, and Vladimir Stojanović. 2017. A 40-Gb/s PAM-4 transmitter based on a ring-resonator optical DAC in 45-nm SOI CMOS. *IEEE Journal of Solid-State Circuits* 52, 12 (2017), 3503–3516.

[43] Naveen Muralimanohar, Rajeev Balasubramonian, and Norman P Jouppi. 2009. CACTI 6.0: A tool to model large caches. *HP laboratories* 27 (2009), 28.

[44] Mike O'Connor, Niladrish Chatterjee, Donghyuk Lee, John Wilson, Aditya Agrawal, Stephen W Keckler, and William J Dally. 2017. Fine-grained DRAM: Energy-efficient DRAM for extreme bandwidth systems. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. 41–54.

[45] Angshuman Parashar, Minsoo Rhu, Anurag Mukkara, Antonio Puglielli, Rangharajan Venkatesan, Brucek Khailany, Joel Emer, Stephen W Keckler, and William J Dally. 2017. SCNN: An accelerator for compressed-sparse convolutional neural networks. *ACM SIGARCH computer architecture news* 45, 2 (2017), 27–40.

[46] Nicola Peserico, Jiawei Meng, Hangbo Yang, Xiaoxuan Ma, Shurui Li, Hamed Dalir, Puneet Gupta, Chee Wei Wong, and Volker J Sorger. 2023. Design and testing of silicon photonic 4F system for convolutional neural networks. In *Integrated Optics: Devices, Materials, and Technologies XXVII*, Vol. 12424. SPIE, 112–121.

[47] Michal Rakowski, Colleen Meagher, Karen Nummy, Abdelsalam Aboketaf, Javier Ayala, Yusheng Bian, Brendan Harris, Kate Mclean, Kevin McStay, Asli Sahin, Louis Medina, Bo Peng, Zoey Sowinski, Andy Stricker, Thomas Houghton, Crystal Hedges, Ken Giewont, Ajey Jacob, Ted Letavic, Dave Riggs, Anthony Yu, and John Pellerin. 2020. 45nm CMOS — Silicon Photonics Monolithic Technology (45CLO) for Next-Generation, Low Power and High Speed Optical Interconnects. In *2020 Optical Fiber Communications Conference and Exhibition (OFC)*. 1–3.

[48] Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou, et al. 2020. Mlperf inference benchmark. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 446–459.

[49] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788. https://doi.org/10.1109/CVPR.2016.91

[50] Ananda Samajdar, Yuhao Zhu, Paul Whatmough, Matthew Mattina, and Tushar Krishna. 2018. Scale-sim: Systolic cnn accelerator simulator. *arXiv preprint arXiv:1811.02883* (2018).

[51] Yakun Sophia Shao, Jason Clemons, Rangharajan Venkatesan, Brian Zimmer, Matthew Fojtik, Nan Jiang, Ben Keller, Alicia Klinefelter, Nathaniel Pinckney, Priyanka Raina, Stephen G. Tell, Yanqing Zhang, William J. Dally, Joel Emer, C. Thomas Gray, Brucek Khailany, and Stephen W. Keckler. 2019. Simba: Scaling Deep-Learning Inference with Multi-Chip-Module-Based Architecture. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture* (Columbus, OH, USA) *(MICRO '52)*. Association for Computing Machinery, New York, NY, USA, 14–27. https://doi.org/10.1145/3352460.3358302

[52] Kyle Shiflett, Avinash Karanth, Razvan Bunescu, and Ahmed Louri. 2021. Albireo: Energy-efficient acceleration of convolutional neural networks via silicon photonics. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 860–873.

[53] Kyle Shiflett, Dylan Wright, Avinash Karanth, and Ahmed Louri. 2020. PIXEL: Photonic neural network accelerator. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 474–487.

[54] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[55] Sanghyun Son, Seungjun Nah, and Kyoung Mu Lee. 2018. Clustering convolutional kernels to compress deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 216–232.

[56] Febin Sunny, Asif Mirza, Mahdi Nikdast, and Sudeep Pasricha. 2021. CrossLight: A Cross-Layer Optimized Silicon Photonic Neural Network Accelerator. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. 1069–1074. https://doi.org/10.1109/DAC18074.2021.9586161

[57] Eddy Chi-Poon Tam, TS Francis, Don A Gregory, and Richard D Juday. 1990. Autonomous real-time object tracking with an adaptive joint transform correlator. *Optical Engineering* 29, 4 (1990), 314–320.

[58] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[59] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*. PMLR, 10347–10357.

[60] Renu Tripathi and Kehar Singh. 1998. Pattern discrimination using a bank of wavelet filters in a joint transform correlator. *Optical Engineering* 37, 2 (1998), 532–538.

[61] Karen Ullrich, Edward Meeds, and Max Welling. 2017. Soft weight-sharing for neural network compression. *arXiv preprint arXiv:1702.04008* (2017).

[62] Qiwen Wang, Xinxin Wang, Seung Hwan Lee, Fan-Hsuan Meng, and Wei D. Lu. 2019. A Deep Neural Network Accelerator Based on Tiled RRAM Architecture. In *2019 IEEE International Electron Devices Meeting (IEDM)*. 14.4.1–14.4.4. https://doi.org/10.1109/IEDM19573.2019.8993641

[63] CS Weaver and Joseph W Goodman. 1966. A technique for optically convolving two functions. *Applied optics* 5, 7 (1966), 1248–1249.

[64] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. 2021. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 22–31.

[65] Junru Wu, Yue Wang, Zhenyu Wu, Zhangyang Wang, Ashok Veeraraghavan, and Yingyan Lin. 2018. Deep k-means: Re-training and parameter sharing with harder cluster assignments for compressing deep convolutions. In *International Conference on Machine Learning*. PMLR, 5363–5372.

[66] Xingyuan Xu, Jiayang Wu, Thach G. Nguyen, Tania Moein, Sai T. Chu, Brent E. Little, Roberto Morandotti, Arnan Mitchell, and David J. Moss. 2018. Photonic microwave true time delays for phased array antennas using a 49 GHz FSR integrated optical micro-comb source. *Photon. Res.* 6, 5 (May 2018), B30–B36. https://doi.org/10.1364/PRJ.6.000B30

[67] Hangbo Yang, Shurui Li, Xiaoxuan Ma, Jonathan K. George, Puneet Gupta, Volker J. Sorger, and Chee Wei Wong. 2022. Programmable On-chip Photonic Machine Learning System Based on Joint Transform Correlator, In Conference on Lasers and Electro-Optics. *Conference on Lasers and Electro-Optics*, JW3B.19. https://opg.optica.org/abstract.cfm?URI=CLEO_SI-2022-JW3B.19

[68] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. 2021. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 579–588.

[69] Yi Zhang, Shuyu Yang, Andy Eu-Jin Lim, Guo-Qiang Lo, Christophe Galland, Tom Baehr-Jones, and Michael Hochberg. 2013. A compact and low loss Y-junction for submicron silicon waveguide. *Optics express* 21, 1 (2013), 1310–1316.

[70] Brian Zimmer, Rangharajan Venkatesan, Yakun Sophia Shao, Jason Clemons, Matthew Fojtik, Nan Jiang, Ben Keller, Alicia Klinefelter, Nathaniel Pinckney, Priyanka Raina, Stephen G. Tell, Yanqing Zhang, William J. Dally, Joel S. Emer, C. Thomas Gray, Stephen W. Keckler, and Brucek Khailany. 2020. A 0.32–128 TOPS, Scalable Multi-Chip-Module-Based Deep Neural Network Inference Accelerator With Ground-Referenced Signaling in 16 nm. *IEEE Journal of Solid-State Circuits* 55, 4 (2020), 920–932. https://doi.org/10.1109/JSSC.2019.2960488

[71] Farzaneh Zokaee, Qian Lou, Nathan Youngblood, Weichen Liu, Yiyuan Xie, and Lei Jiang. 2020. LightBulb: A photonic-nonvolatile-memory-based accelerator for binarized convolutional neural networks. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1438–1443.