Inference analysis of optical transformers

Chenchen Wang*
Department of Electrical Engineering, Tsinghua University, Beijing, China
wcc20@mails.tsinghua.edu.cn

Djamshid A. Damry
Lumai Ltd.,
Wood Centre for Innovation, Quarry Road, Headington, Oxford, OX3 8SB, UK
djamshid.damry@lumai.co.uk

Xianxin Guo
Lumai Ltd.,
Wood Centre for Innovation, Quarry Road, Headington, Oxford, OX3 8SB, UK
xianxin.guo@lumai.co.uk

Abstract

This paper explores the utilization of optical computing for accelerating inference in transformer models, which have demonstrated substantial success in various applications. Optical computing offers ultra-fast computation and ultra-high energy efficiency compared to conventional electronics. Our findings suggest that optical implementation has the potential to achieve a significant 10-100 times improvement in the inference throughput of transformer models.

1 Introduction

Since its invention [Vaswani et al., 2017], the transformer model has been widely applied in natural language processing and various other machine learning tasks, such as computer vision, graph, and multi-modal processing. Recently, we have witnessed an exponential increase in the size of transformer models, enabling remarkable performances across many fields. Despite the advent of modern GPUs facilitating the processing of continuously growing large language models (LLMs), the time and resource cost in model training and inference remains a significant barrier to the deployment of LLMs.

The training of LLMs is mostly compute limited, hence it can be efficiently accelerated by adding more compute power, either with more compute nodes or faster processors. The inference, however, poses a greater challenge to the compute hardware [Pope et al., 2023, Aminabadi et al., 2022]. On the one hand, the degree of parallelism that can be exploited at the inference phase is significantly lower than that at the training phase, which limits the computing efficiency. On the other hand, the two key performance metrics of inference, *throughput* and *latency*, have different requirements on the hardware and therefore it is difficult to balance the performance. In this paper, we focus on the inference analysis of LLMs. More specifically, we investigate the potential of optical computing for throughput enhancement during LLM inference.

37th First Workshop on Machine Learning with New Compute Paradigms at NeurIPS 2023(MLNPCP 2023).

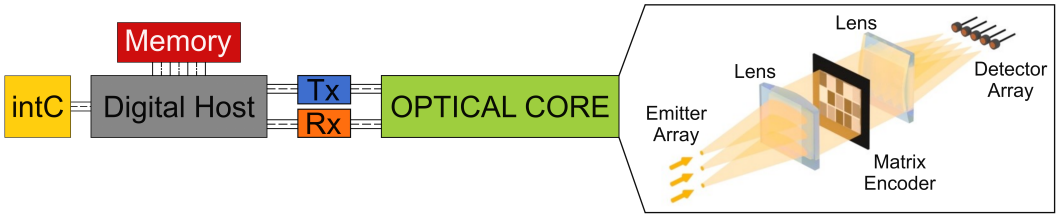

Figure 1: Conceptual diagram of free-space optical matrix-vector multiplication. intC: interconnect

2 Optical matrix multiplication

Matrix multiplication is ubiquitous in modern machine learning, and it accounts for over 90% of the computation in LLMs [Anderson et al., 2023]. This operation can be implemented in different forms: vector-vector multiplication (VVM), matrix-vector multiplication (MVM) and matrix-matrix multiplication (MMM), and they can all be optically realized [Tamura and Wyant, 1979, Spall et al., 2020, Wang et al., 2022]. Here we consider optical processors (OPs) that perform MVM implemented using free-space optics, as illustrated in Fig. 1.

In this scheme, an input vector is encoded onto the amplitude of a light emitter array. The amplitude-modulated light beams fan out during propagation, whilst being shaped with the aid of optical lenses such that the beams pass through a pixelated display properly whereby the transmission pattern is mapped from the input matrix. Therefore, we achieve element-wise multiplication between the vector and matrix when the beams pass through the display. MVM result is obtained by simply introducing a converging lens after the display to accumulate the element-wise multiplication values.

This scheme has several key advantages as compared to other optical implementations:

- **Matrix core size:** The optical matrix core size can reach 2000×2000 or above [Chen et al., 2018], limited by the pixel resolution of the display. This core size is much larger than that achievable with integrated photonics. As explained later, a larger optical matrix core improves the overall computing performance, including compute speed, cost-effectiveness and energy efficiency.
- **Precision:** Calculation precision of 6-8 bit integer-equivalent have been achieved [Spall et al., 2022, Anderson et al., 2023], and this precision is largely independent of the matrix core size, as already demonstrated. This is because light beams propagate in parallel and independently with minimal crosstalk in this scheme.
- **Hardware cost:** System cost is dominated by the costs of the light emitters, receivers, data converters, amplifiers, and vector memory chips, which increase linearly with the channel number (vector size). The cost of matrix-related parts such as high-resolution display and the driver can be very low. Since compute speed scales quadratically with the channel number, cost-effectiveness (Tera operations per second (TOPS) per dollar) increases with matrix core size.
- **Energy cost:** Energy cost of the full system is also dominated by light emitters, receivers, data converters, amplifiers, and vector memory chips. Therefore, the energy cost scales linearly with the vector size, while the compute speed scales quadratically. Therefore, energy efficiency (TOPS per Watt) increases with matrix core size. A detailed energy breakdown can be found in [Anderson et al., 2023].

The MVM scheme can be adapted to perform VVM or MMM, but neither is as hardware efficient as MVM. In VVM, the compute speed and all the costs scale linearly with the vector size, which means that the cost-effectiveness, energy efficiency, and arithmetic intensity cannot be improved. Therefore, VVM has low practical value. In MMM, the compute speed scales as $O(N^3)$, where $N \times N$ is the matrix size, and both the hardware cost and energy cost scales as $O(N^2)$. Although the overall computing performance improves with the matrix size, optical MMM is less resource-efficient as compared to MVM. With current technology, we expect that a single optical core capable of 80,000 TOPS is within reach using a vector rate of 10 GHz and a matrix size of 2000×2000. Both the memory and communication bandwidth need to be improved correspondingly to deliver this ultra-high compute speed in practice.

*Work done while at Lumai

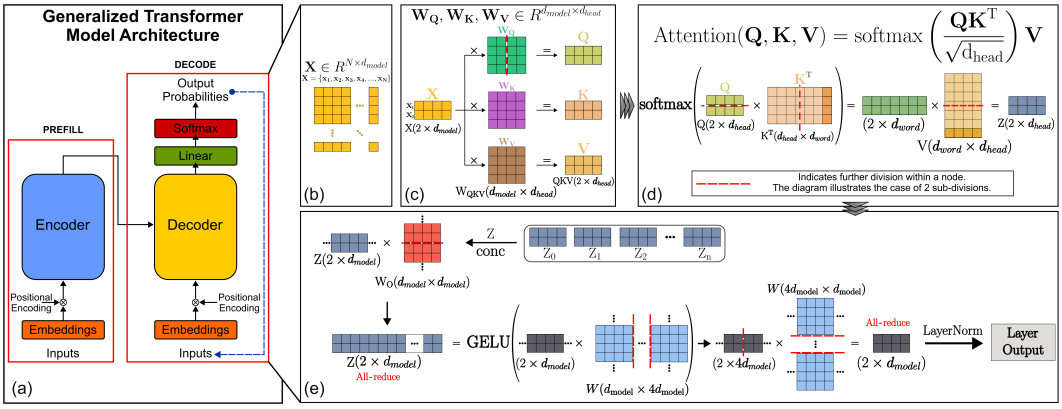

Figure 2: Architecture of a transformer block.

3 Challenges in LLM inference

LLMs can be broadly categorized as encoder-only, decoder-only and encoder-decoder models [Zhao et al., 2023]. Here we focus on decoder-only models such as GPT series. These decoder-only models actually include one encoder block that processes user input information (known as the prefill or prompt processing phase), and multiple auto-regressive decoder blocks that output tokens one by one (known as the decode or token generation phase), as illustrated in Fig. 2(a). These encoder and decoder blocks are all transformer blocks, and each transformer block consists of an attention block and a feed-forward network (FFN). The attention mechanism is essential in LLMs as it captures long-range dependencies of input tokens and enables contextual understanding. Early versions of LLMs employed multi-head attention (MHA) to focus on different parts of input data simultaneously, but recently grouped-query attention (GQA) and multi-query attention (MQA) have gained popularity due to their abilities to process much longer input sequences with reduced memory requirement.

As LLMs are hard to be fully loaded onto a single hardware device with limited capacity, efficient inference needs to be operated with multiple compute nodes. Distributed inference can be accelerated with Tensor Parallelism (TP) and other parallel strategies. TP is a model parallelism strategy, and it splits a tensor into several smaller chunks, with different chunks processed on different hardware devices. After completing the tensor processing for each part, partial results are aggregated to yield the correct result. Other parallelism includes Data Parallelism (DP), Pipeline Parallelism (PP), Fully Sharded Data Parallelism (FSDP) and so on. In practice, different parallel strategies can be combined flexibly. In this paper, we mainly use TP for the inference analysis, and we leave other strategies for future works.

The basic architecture of a transformer block and its TP implementation at the decode phase is illustrated in Fig. 2 (b-e). The attention layer consists of the following operations: 1) Calculation of the Query, Key and Value (QKV) vectors for an input token; 2) Calculation of the multi-head attention scores; 3) Projection of the attention output. As shown by the Megatron work [Shoeybi et al., 2019], MHA can be parallelized by splitting the attention heads onto N_{TP1} different nodes, and inter-node communication is only invoked at the last projection stage. Within each node, we can further split the QKV vectors and projection matrices into N_{TP2} chunks along the directions shown in Fig. 2. Intra-node communication is needed during the attention calculation step to obtain the correct attention scores through the 'all reduce' communication process [Pope et al., 2023]. For the feed-forward layers, we can perform TP and split the weight matrices similarly.

Now that we have explained the model architecture and parallelism, we can analyze the costs and various tradeoffs of inference. Given that different layers of the model share the same structure, we only consider the throughput and latency of a single layer. Table. 1 lists the parameter byte counts and arithmetic operations per processor, as well as communication volume at different steps of a MHA-based transformer.

KV cache: At the decode phase, we need to calculate the attention score between the current token and all the previous tokens. The re-calculation of KV matrices for all the previous tokens is extremely time consuming, hence it is common practice to cache the previous KV matrices and load from

Table 1: Inference cost breakdown of a MHA-based transformer. B: bytes per parameter.

Operation	Memory cost	Arithmetic operations	Communication cost
QKV calculation	$\frac{3Bd_{\text{model}}d_{\text{head}}n_{\text{heads}}}{N_{\text{TP1}}N_{\text{TP2}}}$	$\frac{6d_{\text{model}}d_{\text{head}}n_{\text{heads}}N_{\text{batch}}}{N_{\text{TP1}}N_{\text{TP2}}}$	0
KV attention	$\frac{2Bd_{\text{head}}n_{\text{words}}N_{\text{batch}}}{N_{\text{TP1}}N_{\text{TP2}}}$	$\frac{4d_{\text{head}}n_{\text{words}}N_{\text{batch}}}{N_{\text{TP1}}N_{\text{TP2}}}$	0
Attention projection	$\frac{Bd_{\text{model}}d_{\text{head}}n_{\text{heads}}}{N_{\text{TP1}}N_{\text{TP2}}}$	$\frac{2d_{\text{model}}d_{\text{head}}n_{\text{heads}}N_{\text{batch}}}{N_{\text{TP1}}N_{\text{TP2}}}$	$\frac{Bd_{\text{model}}(N_{\text{TP1}}-1)N_{\text{batch}}}{N_{\text{TP1}}}$
FFN	$\frac{8Bd_{\text{model}}^2}{N_{\text{TP1}}N_{\text{TP2}}}$	$\frac{16d_{\text{model}}^2N_{\text{batch}}}{N_{\text{TP1}}N_{\text{TP2}}}$	$\frac{Bd_{\text{model}}(N_{\text{TP1}}-1)N_{\text{batch}}}{N_{\text{TP1}}}$

memory instead of re-calculation. Although KV cache reduces the calculation time, it still presents a major challenge to inference, because the KV cache size grows with sequence length n_{words} and data batch size N_{batch}. In some LLMs the KV cache size can easily exceed the model size, making the memory bandwidth a significant bottleneck.

Memory cost: The primary factors that contribute to memory usage are typically the model size and the KV cache. When dealing with small batch sizes, loading the model itself incurs significant overhead, while with larger batch sizes, the KV cache becomes the dominant factor. To address this challenge, one effective solution is MQA, where one set of KV matrix is shared across multiple heads.

Communication cost: In our TP implementation, when a matrix is distributed across N_{TP} compute nodes, partial MVM results are summed once calculations are completed and then the results broadcast to all nodes. This communication process is efficiently carried out by the 'all reduce' protocol, where the total communication volume is approximately the full matrix size when N_{TP} is large. In this work we consider TP across multiple nodes as well as within each node, hence both inter-node and intra-node communication bandwidth need to be considered properly.

Compute cost: Although LLMs are matrix multiplication-intensive, if the hardware or the model is not designed properly, LLMs can easily be memory or communication limited. To improve the compute efficiency, we can increase the batch size and adop MQA instead of MHA. To make sure that communication also is not a bottleneck, we use a small number of TP nodes. With these configurations, LLMs are mainly compute limited, and they can be efficiently accelerated with ultra-fast optical computing hardware.

4 Optically accelerated inference

4.1 Inference on a GPU cluster

Following the previous discussion, we now proceed to the analysis of actual performance on a GPU cluster as an illustration. We examine two LLM models: (1) GPT-175B as a typical example of current models, and (2) a future model where both d_{head} and n_{heads} are $\times 10$ larger, and d_{model} is $\times 100$ larger. Model parameters of GPT-175B are: $d_{\text{model}} = 12288$, $d_{\text{head}} = 128$, $n_{\text{heads}} = 96$. We also set the context window size n_{words} to 100k which has been demonstrated by several LLMs.

For the GPU cluster, we use NVIDIA H100 DGX as our compute node. Each node includes 8 NVIDIA H100 GPUs, and a single GPU has 80 GB of memory, 2000 TOPS INT8 non-sparse peak speed, and 3.35 TB/s of memory bandwidth. The GPU-to-GPU bidirectional communication bandwidth as supported by NVLink is 900 GB/s, and node-to-node bidirectional bandwidth as supported by Infiniband is 1000 GB/s. Here we use TP as our parallel strategy, with $N_{\text{TP1}} = 6$ nodes for different heads, $N_{\text{TP2}} = 8$ for TP within each node. In total, the cluster consists of 48 GPUs.

In Fig. 3 we list the time costs of four models in the decode phase: current model based on MHA (GPT-175B-MHA), current model based on MQA (GPT-175B-MQA), future model based on MHA (FM-MHA), and future model based on MQA (FM-MQA). From the figures we can clearly see that memory time is the dominant factor in many cases, particularly in MHA-based models. The bottom row shows the breakdown of memory loading time, and we can see that the memory bottleneck is primarily caused by the KV cache, the loading time of which grows linearly with batch size. This

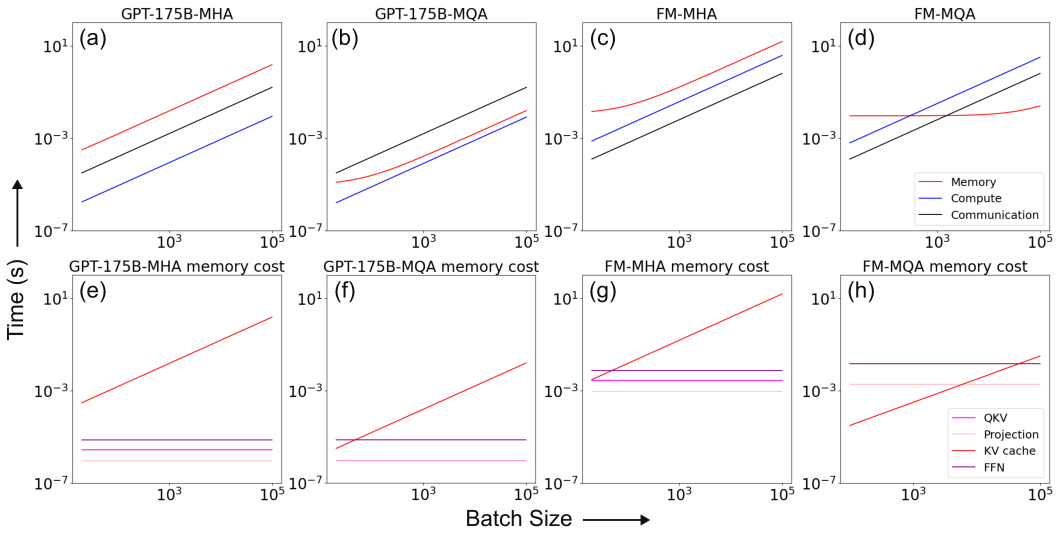

Figure 3: Time cost of inference on a GPU cluster. The top panel shows the overall memory, compute and communication times for four models, and the bottom panel shows the memory cost breakdown of each model.

memory loading time can be effectively reduced by MQA, and the reduction effect is much more significant for future larger models. Fig. 3(d) shows that FM-MQA will be compute limited over a broad range of batch size.

4.2 Inference on an optical cluster

Now we analyze the situation where MHA-based models are deployed on optical accelerators benchmarked against the H100 with the same batch number (64) and number of chips (4). Three optical processors (OPs) are considered: OP1 with a matrix size of 1024×1024 and a vector rate of 1GHz, OP2 with a matrix size of 2048×2048 and a vector rate of 1GHz, and OP3 with a matrix size of 2048×2048 and a vector rate of 4GHz. With the same display refresh rate of 100KHz, the peak TOPS for OP1, OP2, and OP3 are 2097, 8389, and 33554 respectively as shown in Fig. 4.

The bar charts in Fig. 4 showcase various throughputs of the OPs at both prefill and decode stages of the LLaMA-70B model with 2048 input tokens. We see that all the OPs achieve higher throughput than the H100 cluster, with OP3 throughput being 11 times higher at the prefill phase and 4 times higher at the decode phase. The prefill phase is more compute intensive so the optical speedup effect is more significant. Besides, we also see the throughput enhancement is below the peak compute speed enhancement. This is because existing model such as LLaMA-70B cannot saturate faster OPs with larger optical matrix cores. Future larger models can achieve higher TOPS utilization efficiency, thus achieving better throughput improvement. At the decode phase, since tokens are generated one by one, model TOPS utilization efficiency on both OPs and GPUs are typically very low. Our result shows that although OPs can also achieve higher throughput at the decode phase, the optical utilization efficiency is still too low. More optimization techniques need to be developed to fully deliver the optical advantage in the decode phase.

5 Discussion

In this work, we explored optical speedup advantages in distributed inference of LLMs. The efficient serving of LLMs is a systematic engineering challenge with many practical complications. By analyzing the tradeoffs of compute, memory and communication costs, we identify scenarios where LLMs are compute limited: models that are larger in size, and models that adopt GHA or MHA to enable a large context window. Importantly, we see a clear trend of LLMs moving towards these directions. Throughput of these LLMs can be enhanced by up to 11 times in the prefill stage and 4 times in the decode stage using optical processors with 16 times higher peak speed. As the optical

Processor	Matrix size	Vector rate	Display	Peak TOPS	VMem BW	IO BW
OP1	1024x1024	1 GHz	100 kHz	2097	1 TB/s	600 GB/s
OP2	2048x2048	1 GHz	100 kHz	8389	2 TB/s	1.2 TB/s
OP3	2048x2048	4 GHz	100 kHz	33554	8 TB/s	2.4 TB/s

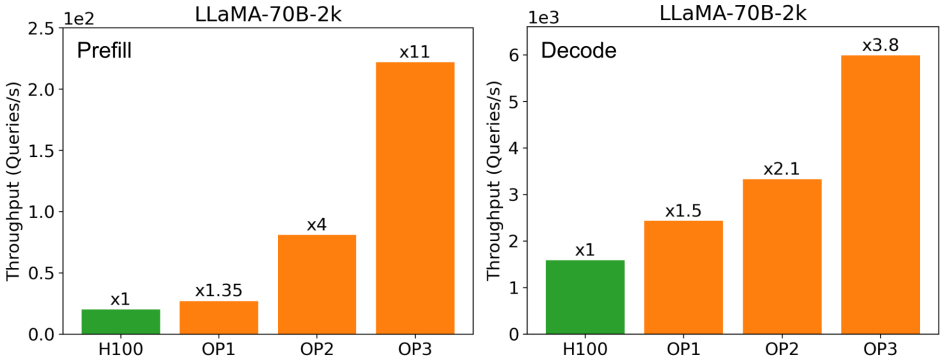

Figure 4: Inference performance of an optical cluster.

core is capable of 80,000 TOPS peak speed with current technology, we expect that about 100 times throughput enhancement from optics is possible, but both the memory and communication bandwidths need to be improved as well.

To deliver these optical advantages, optical hardware developers need to consider not only the peak compute speed, but also the hardware utilization efficiency and the matrix refresh rate. In the free-space optical matrix multiplier, the compute speed increases quadratically with the matrix core size. However, a larger core size might lead to lower utilization efficiency. One way to mitigate this is to allow for flexible matrix partitioning within each optical processor such that one large optical core can be configured as multiple smaller parallel cores. Regarding the matrix update speed, although during inference the matrices do not need to be updated as fast and frequently as the vectors, the optical cores may be reconfigured to handle different layers. In our analysis we find that a matrix refresh rate of 100 kHz is sufficient in many cases. Although existing displays are usually not faster than 10 KHz, faster displays may be achieved using phase-change materials [Shields et al., 2023, Zhang et al., 2019], electro-optical modulation [Trajtenberg-Mills et al., 2023] and so on. Furthermore, even slow displays are not necessarily a bottleneck, since the displays can be refreshed row-by-row with proper driver circuits, and multiple displays can work in pipelines, much higher effective refresh rate is possible.

As of today, LLMs are still rapidly evolving with many new optimization techniques, and we expect that both our memory and communication costs can be further reduced to yield more drastic optical computational speedup. For example, sparse attention [Child et al., 2019] can be applied to significantly reduce the memory footprint of attention layers. Advanced parallel strategies such as FSDP or DeepSpeed [Aminabadi et al., 2022] can be used for more efficient model partitioning and memory footprint reduction. Although we haven't applied all the advanced optimization techniques in our analysis, this work provides a framework for proper evaluation of optical computing advantages in LLM inference.

References

R. Y. Aminabadi, S. Rajbhandari, A. A. Awan, C. Li, D. Li, E. Zheng, O. Ruwase, S. Smith, M. Zhang, J. Rasley, et al. Deepspeed-inference: enabling efficient inference of transformer models at unprecedented scale. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE, 2022.

M. G. Anderson, S.-Y. Ma, T. Wang, L. G. Wright, and P. L. McMahon. Optical transformers. *arXiv preprint arXiv:2302.10360*, 2023.

H.-M. P. Chen, J.-P. Yang, H.-T. Yen, Z.-N. Hsu, Y. Huang, and S.-T. Wu. Pursuing high quality phase-only liquid crystal on silicon (lcos) devices. *Applied Sciences*, 8(11):2323, 2018.

R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

R. Pope, S. Douglas, A. Chowdhery, J. Devlin, J. Bradbury, J. Heek, K. Xiao, S. Agrawal, and J. Dean. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5, 2023.

J. Shields, C. R. De Galarreta, H. Penketh, Y.-Y. Au, J. Bertolotti, and C. D. Wright. A route to ultra-fast amplitude-only spatial light modulation using phase-change materials. *Advanced Optical Materials*, page 2300765, 2023.

M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

J. Spall, X. Guo, T. D. Barrett, and A. Lvovsky. Fully reconfigurable coherent optical vector–matrix multiplication. *Optics Letters*, 45(20):5752–5755, 2020.

J. Spall, X. Guo, and A. I. Lvovsky. Hybrid training of optical neural networks. *Optica*, 9(7):803–811, 2022.

P. N. Tamura and J. C. Wyant. Two-dimensional matrix multiplication using coherent optical techniques. *Optical Engineering*, 18(2):198–204, 1979.

S. Trajtenberg-Mills, M. ElKabbash, C. Brabec, C. Panuski, I. Christen, and D. Englund. Lithium niobite on silicon high speed spatial light modulator. In *CLEO: Science and Innovations*, pages SF1E–4. Optica Publishing Group, 2023.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

T. Wang, S.-Y. Ma, L. G. Wright, T. Onodera, B. C. Richard, and P. L. McMahon. An optical neural network using less than 1 photon per multiplication. *Nature Communications*, 13(1):123, 2022.

Y. Zhang, J. B. Chou, J. Li, H. Li, Q. Du, A. Yadav, S. Zhou, M. Y. Shalaginov, Z. Fang, H. Zhong, et al. Broadband transparent optical phase change materials for high-performance nonvolatile photonics. *Nature communications*, 10(1):4279, 2019.

W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.