Energy-Efficient Inference Scaling of Optical Transformers

Maxwell G. Anderson
Cornell University
mga58@cornell.edu

Shi-Yuan Ma
Cornell University

Tianyu Wang
Cornell University

Logan G. Wright
Cornell University
NTT Research

Peter L. McMahon
Cornell University
pmcmahon@cornell.edu

Abstract

The rapidly increasing size of deep-learning models has renewed interest in alternatives to digital-electronic computers as a means to dramatically reduce the inference energy cost of running state-of-the-art neural networks. Optical matrix-vector multipliers are best suited to performing computations with very large operands, which suggests that large Transformer models could be a good target for them. However, the ability of optical accelerators to run efficiently depends on the model being run, and if the model can be run at all when subject to the noise, error, and low precision of analog-optical hardware. Here we investigate whether Transformers meet the criteria to be efficient when running optically, what benefits can be had for doing so, and how worthwhile it is at scale. We found using small-scale experiments on and simulation of a prototype hardware accelerator that Transformers may run on optical hardware, and that elements of their design — the ability to parallel-process data using the same weights, and trends in scaling them to enormous widths — allow them to achieve an asymptotic energy-efficiency advantage running optically compared to on digital hardware. Based on a model of a full optical accelerator system, we predict that well-engineered, large-scale optical hardware should be able to achieve a 100× energy-efficiency advantage over current digital-electronic processors in running some of the largest current Transformer models, and if both the models and the optical hardware are scaled to the quadrillion-parameter regime, optical accelerators could have a > 8,000× energy-efficiency advantage.

This is an abridged version of our full paper at `https://arxiv.org/abs/2302.10360`.

1 Introduction

Deep learning models' exponentially increasing scale is both a key driver in advancing the state-of-the-art and a cause of growing concern about their energy usage, speed, and practicality. This has led to the development of hardware accelerators and model training/compression/design techniques for efficient and fast inference on them. Because they still perform all the underlying operations using the same physical mechanisms, most digital-electronic accelerators [1–5] can improve performance by constant factors. This is because in digital systems there is an energy cost for every computation [6], so improvements do not change the way costs scale with the number of computations to perform.

Analog accelerators can differ from digital ones in that the energy cost of performing computations may fundamentally scale differently than digital systems. For example, in optics or analog-electronic

First Workshop on Machine Learning with New Compute Paradigms at NeurIPS 2023(MLNCP 2023).

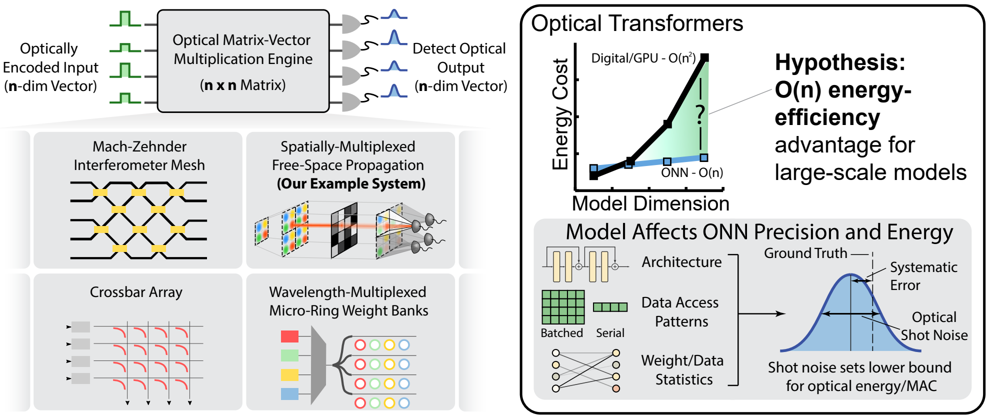

Figure 1: **Can Transformers benefit from running on optical hardware?** Optical Neural Networks (ONNs) have been proposed as an alternative computing platform that can achieve asymptotic energy-efficiency advantages over digital computers running neural networks. This is not a guarantee; their behavior is affected by model architecture, statistics, and resilience to the noise/imprecision of analog hardware. Thus, while there are many implementations of general-purpose optical matrix accelerators (such as those depicted in the inset), there are still model-dependent challenges/tradeoffs in realizing their purported advantages. We seek here to answer the question of how much today's enormous Transformer models can benefit from this technology. We hypothesize that Transformers' architecture allows for ONN-enabled benefits that scale.

crossbar arrays, a common heuristic is that the energy of a matrix-vector product scales *linearly* with vector size, rather than the $\sim d^2$ of digital systems (assuming all dimensions are $\sim d$). This is a key intuition for why alternative analog computing platforms using optics have been proposed as a new paradigm for better scalability [7–13]. Ideally, the scaling is asymptotically better than digital systems in energy per MAC [6, 14, 15, 10].

But these optical neural networks (ONNs) have additional complexities and limitations of their own such as low precision, noise, and analog/digital data conversion overheads which depend on the access patterns of the model running. Benefits can only be realized when operands reach a certain scale and level of parallelism, because optics-based systems have additional overheads such as analog-digital conversion whose costs must be amortized. Thus, advantageously accelerating any neural network architecture with ONNs is in practice hard, and DNNs without the necessary activation statistics and model architecture may not achieve this scaling.

But Transformers' efficient data-access patterns (wide layers, parallel/batched token processing, etc.) and trends in methods for scaling them make them an especially attractive match to leverage this analog optical scaling advantage for asymptotic energy-efficiency. Here, our goal is to investigate if this Optical Transformer hypothesis is true in realistic settings (Figure 1, right) — with real noise, hardware imperfections, memory and digital-analog-conversion costs, and state-of-the-art models.

Here we demonstrate how Transformers run on ONN systems, and estimate the potential benefits of doing so. To first verify that Transformers may run on these systems despite their imprecision, we sampled operations from a Transformer and ran them on a real spatial light modulator (SLM) based experimental system, and used the results to create a calibrated simulation of the optical hardware, with the systematic error, noise, and imprecision of weights/inputs we observed. Transformers running on the simulated hardware could perform nearly as well as those running digitally, and could be far more efficient.

2 Background and Related Work

2.1 Large-Scale Deep Learning

In the past few years, it has been found in particular that Transformer architectures significantly improve when sized up to billions or even trillions of parameters [16–21], causing an exponential growth of deep learning compute usage [22, 23]. These large-scale Transformers achieve ever

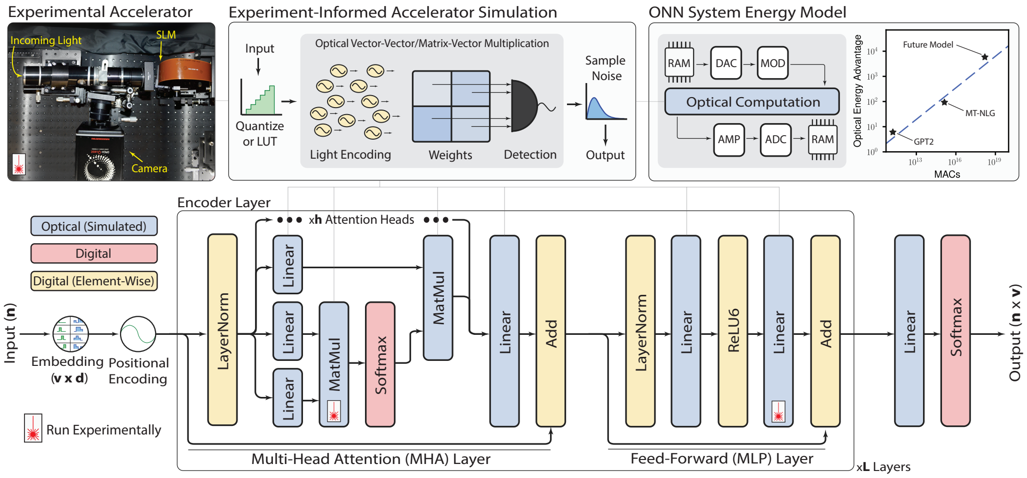

Figure 2: **Optical Transformer evaluation: prototype hardware; simulator model; Transformer architecture.** Bottom: typical Transformer architecture, but with ReLU6 activation. Top Left: experimental spatial light modulator (SLM)-based accelerator setup. From some layers—marked with a laser icon—we sampled dot products to run on real hardware. Top Middle: Linear operations, in light blue, run on a simulated accelerator with noise/error. Lookup tables (LUT) allow simulation using our setup's supported weight/activation values. Top right: our model of energy consumption for optical accelerators, based on assumptions and results from our experiment/simulations. The model accelerator system consists of random-access memory (RAM), a analog/digital conversion (DAC/ADC), light modulation (MOD), amplification (AMP).

more impressive results in not only natural language processing, but also in other domains such as computer vision [24, 25], graphs [26], and in multi-modal settings [27–32], making them a popular but expensive solution for many tasks—digital hardware's energy efficiency (ie. per-flop or per-inference cost) has not kept up with the growing FLOP requirements of state-of-the-art deep learning models [23]. They also have transfer learning capabilities [33–35, 16, 36, 24], allowing them to easily generalize to specific tasks, in some cases in a zero-shot setting where no further training is necessary [16, 30, 37].

2.2 Traits of Optical Accelerators

Researchers have explored a wide variety of controllable optical systems to implement linear operations on optical fields, such as arbitrary matrix-vector multiplications, vector-vector dot products [38, 39, 6, 40, 41, 14, 42–44], or convolutions [45–49]. In this work, we adopt one kind of free-space multiplier [14, 40, 42] (Figure 2, top left) to demonstrate Transformer operations in optical experiments and for our simulations. We selected this system because it has many of the same characteristics as other ONN implementations (photon detection noise, free optical data transport and reuse, systematic errors), and aim to draw conclusions that could generally be useful for those working with other ONN designs. Many ONN systems, including ours, share the following typical traits (Figure 1):

Optical Shot Noise Noise in photon detection that causes outputs to be Poisson-distributed around the ground truth value. This limits the precision of the accelerator's outputs. The signal-to-noise — the effective precision — depends only on the photon totals at the *output* of an operation.

Device Imprecision and Systematic Errors Systematic errors, on the other hand, are not noise but rather errors resulting from deficiencies of the hardware. These errors occur consistently; they are not random. ONN devices also have precision constraints, often only supporting a number of mappable transmission/emission levels.

Free Data Transport and Reuse Transport and copying of data encoded in light is free when performed optically. However, when splitting a signal in this way, the total amount of light is divided by the number of copies. The ability to reuse data in this way is particularly important because it allows for a device to pay the cost of loading data only once (such as a vector) to be used as much as is necessary (such as for multiple vector-vector dot products in a matrix-vector product).

Efficient Photon Usage Shot noise, and therefore an optical dot product's signal-to-noise ratio is related to the mean number of photons at the *output*. The efficiency of photon usage can therefore grow with increasing multiply-accumulate operations (MACs), because the total amount of optical energy needed for a dot product is independent of the dot product size. Work on ONNs has studied this behavior in a variety of scenarios [6, 10, 14, 15]. For example, if a dot product of size d requires P photons to compute with precision b, then a dot product of size $2d$ only requires the same P photons if b does not change. However, this only holds true when wider models (ie. models with larger dot products in their matrix-vector computations) do not have precision requirements that scale with the model size. Thus the efficient scaling is not a guarantee—the required number of photons may be influenced by a model architecture's activation/weight distributions, encoding schemes, precision requirements, etc [50].

2.3 Optical Accelerator Energy Usage

ONNs' energy consumption is modelled as follows: the energy cost is broken down into the optical costs of performing MACs and the electrical costs of loading/detecting data (including conversion to/from digital/analog), which are usually dominant. Consider a matrix product involving $A \in \mathbb{R}^{n \times d}$, $B \in \mathbb{R}^{d \times k}$. Such a product results in loading (detecting) $nd + dk$ (nk) scalars, and performing ndk MACs. If the energy to electrically load (detect) a scalar is E_{load} (E_{det}), and to perform a MAC optically is E_{optical}, then the total energy is: $E = (nd + dk)E_{\text{load}} + nkE_{\text{det}} + ndkE_{\text{optical}}$. This illustrates how ONNs may have asymptotic energy advantages over digital computers. Notice that regardless of the number of reuses, all data is only loaded once (and partial products are accumulated at a detector before converting and storing the data digitally). Meanwhile, E_{optical} ideally scales as $1/d$. These properties make energy cost disproportional to the number of MACs, ndk. In other words, $\frac{E_{\text{digital}}}{E_{\text{ONN}}} \sim \min(n, d)$ (or just d for weights-in-place systems — systems where weights may be loaded once, and kept in the hardware to be reused for long periods of time). The electrical overheads include the costs of memory read/write, conversion (analog-digital and vice-versa), modulation, and detection. Further details about assumed values, based on existing hardware capabilities, are in Appendix E.

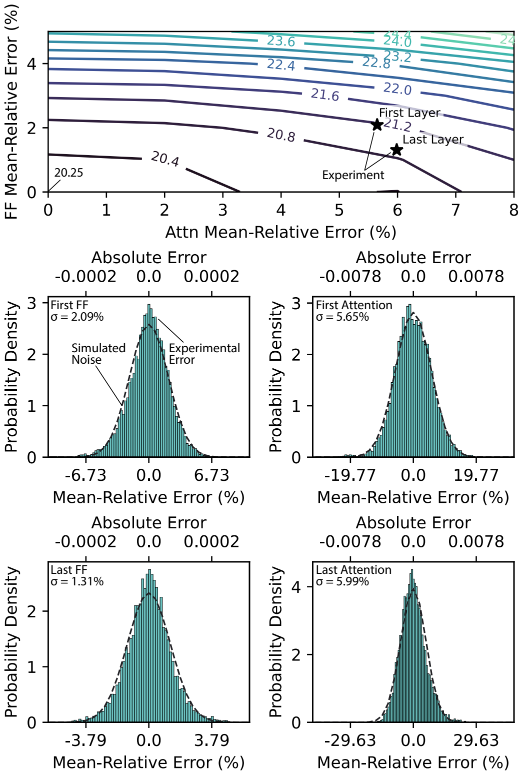

Figure 3: **Comparison of experimental and simulated noise models and simulated Optical Transformer noise tolerance.** Top: Simulated performance (Wikitext-103 validation perplexity (PPL)) versus percent mean-relative simulated noise in feed-forward (FF) and attention (Attn) layers. Systematic errors from experimental data marked with a star. Bottom: comparison of simulated noise model to error from experimental data. The Gaussian shape of the simulated error behavior matches experiment accurately.

3 Optical Transformers

We designed models that are intentionally similar to other Transformers, with the goal of simulating their behavior (informed by some experimental measurements) and energy consumption on optical hardware. A summary of our approach and model is in Figure 2. We created optical Transformer models with a GPT2-like [35] architecture that replaces the GELU [51] activation with ReLU6, which is known to improve low-precision model performance [52–54]. For language

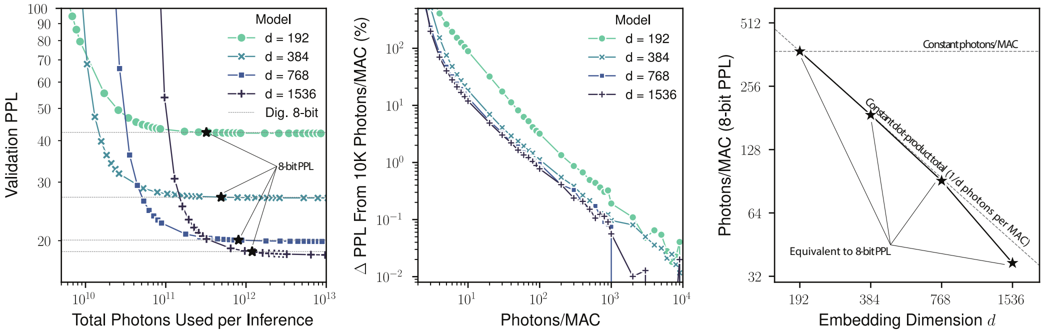

Figure 4: **Simulations of Optical Transformer behavior with varying photon usage.** Left: Wikitext-103 validation-set perplexity (PPL) versus embedding dimension d and total photons used for a single forward pass/inference. 8-bit digital model performance is shown with dashed lines. Middle: perplexity degrades from ideal with fewer photons-per-MAC; the plot exhibits truncated power-law scaling. Right: Scaling of number of photons needed for an Optical Transformer to achieve the same perplexity as an 8-bit digital-electronic processor, versus model size.

modelling, we used the raw Wikitext-103 dataset [55]. The models we simulated have 12 layers (consisting of multi-head attention and feed-forward blocks), operate on a context length of 1024 tokens, use 12 attention heads, and have embedding dimension d varying from 192 to 1536. The full details of the training technique, architecture, and hyperparameters are in Appendix A.

3.1 Transformers Are Resilient To ONN Systematic Errors

We ran experiments using a real Transformer's (we used the base-sized model with $d = 768$) weights in order to characterize the behavior of an ONN system. We adopted as a representative example of an optical accelerator a spatial light modulator (SLM) based system which computes vector-vector dot products [14]. Vectors are encoded on a display, and copies are shone through the SLM which has varying transmission corresponding to some data (ie. a weight matrix). The outputs of this operation—element-wise products—are collected at detectors as the resultant dot products (Figure 2, top left).

Informed by our experiments, we constructed a simulation of the optical hardware, incorporating collected calibration, noise, and error data.We also evaluated the digital, 8-bit-QAT-trained model for comparison purposes.

We found that Transformer operations can be run on real hardware without severely degraded performance from systematic errors. The bottom four panels of Figure 3 are histograms of the experimental differences from correct values. The simulated noise distributions (dotted lines) match well with the experimental data, which confirms that they are an accurate representation of the real systematic error behavior. Figure 3 (top) is a map of the performance of the simulated model over different configurations of the mean-relative (in percent) noise at every layer of feed-forward and attention blocks. The model performs well with significant noise (experimental noise levels marked with stars), within 1 perplexity from noise-free performance unless the noise is very high.

3.2 Transformers' Shot-Noise Resilience and Optical Scaling Laws

We simulated the Transformer models running on optical hardware in the presence of shot noise. To isolate shot noise from other effects, we excluded systematic errors from these simulations. We define photons/MAC as the total photon budget (amount at input) divided by total MACs.

Optical Transformers achieve language modelling performance close to their digital counterparts' when shot-noise-limited at photon budgets where optical energy is negligible. The perplexities on the Wikitext-103 validation set of various optical Transformer models simulated with different total photon usage (amount used for input data) are shown in Figure 4 (left). The curves illustrate a tradeoff: larger models need larger photon totals to function well, and there are different optimal model choices based on the photon budget.

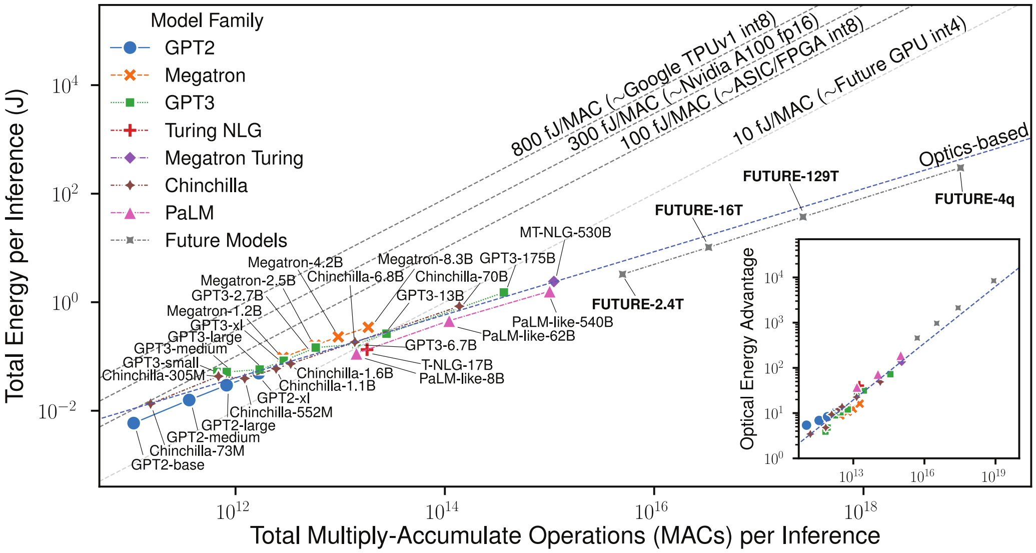

Figure 5: **Estimated energy usage of Transformer models on optical hardware for a single forward pass/inference.** Hypothetical future model designs are labelled **FUTURE-***. Estimated energy/MAC for digital systems is based on [1]. Trend for energy usage in optical systems (blue) computed based on real models only. Inset: energy advantage of running on optics over estimated NVIDIA A100 usage. The advantage grows with the model compute. $\mathrm{M} = 10^6$, $\mathrm{G} = 10^9$, $\mathrm{T} = 10^{12}$, $\mathrm{q} = 10^{15}$ parameters.

The models use fewer photons/MAC as they scale, achieving the theoretical efficient scaling where the total per-dot-product photons needed is constant. To study how photon usage scales, we determined how many photons it takes to reach the performance of 8-bit digital models. These values, in Figure 4 (right), decrease nearly as $\frac{1}{d}$—the total photons needed per dot product is constant (bottom dashed line). The Transformer architecture clearly takes advantage of efficient optical scaling with larger model sizes, suggesting that required output SNR does not increase with scale, and that model statistics and dynamic ranges do not become unmanageable.

3.3 Estimated Energy Usage

The efficient photon scaling trend we observed suggests that Transformers running on optical hardware could achieve significant energy efficiency advantages over running on digital hardware. To understand the efficiency of Transformers on optical hardware, we designed an ONN system based on current hardware that is like our experimental setup, with our measured precision and photon scaling (see: Figure 2, top right). It is an inference system with in-place weights which are loaded once and reused forever, activations read from and written to SRAM for every layer, a 10 GHz light modulator array, and an optical "core" which can perform 10M multiplications per cycle (this can be thought of as a 10 megapixel SLM). The electrical energy costs are calculated using the approach in Section 2.3. We assume that the element-wise nonlinear layers (and softmax) must be run digitally, but none scale as d^2 — so an advantage is still possible in sufficiently large models. Because they are data-access-heavy, we estimate their costs as the cost of storing/loading all operands to/from memory. All linear operations (matrix products, both in attention and MLP) are assumed to run optically, as verified by our experiments.

As models grow, running Transformers on optical hardware has a large and asymptotic efficiency advantage over running on digital hardware. In Figure 5 we chart estimates of the forward pass energy required for various models, which include our calculations for electrical overheads, optical energy, and digital operation overheads[1]. We include a hypothetical family of large, dense Transformer models designed in a similar fashion, which we label **FUTURE-***. For comparison, we also chart various digital systems [1] in different performance regimes, and a hypothetical "next generation"

[1] The recent PaLM [56] models used a modified architecture. A For simpler comparison, we make our estimates using a model with GPT-like architecture but with the PaLM model dimensions, which we call PaLM-Like.

GPU that can use $\sim$10 fJ/MAC. For small models, the optics-based system uses about the same energy, but eventually gains an advantage that scales asymptotically with the number of MACs. For the larger models, MT-NLG-530B and FUTURE-4q, the optics-based approach would have $\sim 140\times$ and $\sim 8500\times$ energy advantages over the current state-of-the-art GPU (NVIDIA A100) respectively.

4 Conclusion

We have demonstrated the ability of Transformer models to run accurately and efficiently on optical hardware through optical experiments and an experiment-informed simulation of future optical hardware. We examined Transformers' scaling behavior with optics and used our findings to show that optical systems could have a large and asymptotic energy advantage over digital-electronic ones that *grows* with the model size. For example, we showed that optical hardware may achieve an over $100\times$ energy advantage when performing inference with the largest Transformer models today ($\sim$500 billion parameters) and that larger, future Transformers ($\sim$4 quadrillion parameters) may be realized with an $>8000\times$ optical energy advantage. We believe our findings about the potential energy-efficiency of optical accelerator hardware strongly motivate the development of optical processors for large-scale deep learning with Transformers.

5 Acknowledgements

The authors wish to thank NTT Research for their financial and technical support. Portions of this work were supported by the National Science Foundation (awards CCF-1918549 and CBET-2123862) and a David and Lucile Packard Foundation Fellowship. We acknowledge helpful discussions with and feedback from Alen Senanian, Benjamin Malia, Fan Wu, Federico Presutti, Jeremie Laydevant, Sridhar Prabhu, and Vladimir Kremenetski.

References

[1] Albert Reuther, Peter Michaleas, Michael Jones, Vijay Gadepally, Siddharth Samsi, and Jeremy Kepner. Survey of machine learning accelerators. *arXiv:2009.00993*, 2020. URL https://arxiv.org/abs/2009.00993.

[2] Graphcore. The data center architecture for graphcore computing. Technical report, Apr 2021. URL https://www.graphcore.ai/hubfs/Graphcore-Mk2-IPU-System-Architecture-GC.pdf.

[3] Cerebras Systems. Cerebras systems: Achieving industry best AI performance through a systems approach. Technical report, Apr 2021. URL https://8968533.fs1.hubspotusercontent-na1.net/hubfs/8968533/Whitepapers/Cerebras-CS-2-Whitepaper.pdf.

[4] Michael Andersch, Greg Palmer, Ronny Krashinsky, Nick Stam, Vishal Mehta, Gonzalo Brito, and Sridhar Ramaswamy. NVIDIA Hopper architecture in-depth. Technical report, March 2022. URL https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/.

[5] Habana Labs. HABANA® GAUDI®2 white paper. Technical report, June 2022. URL https://habana.ai/wp-content/uploads/pdf/2022/gaudi2-whitepaper.pdf.

[6] Ryan Hamerly, Liane Bernstein, Alexander Sludds, Marin Soljačić, and Dirk Englund. Large-scale optical neural networks based on photoelectric multiplication. *Physical Review X*, 9(2): 021032, 2019. URL https://doi.org/10.1103/PhysRevX.9.021032.

[7] Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature Nanotechnology*, 15(7): 529–544, March 2020. doi: 10.1038/s41565-020-0655-z. URL https://doi.org/10.1038/s41565-020-0655-z.

[8] H John Caulfield and Shlomi Dolev. Why future supercomputing requires optics. *Nature Photonics*, 4(5):261–263, 2010. URL https://doi.org/10.1038/nphoton.2010.94.

[9] Gordon Wetzstein, Aydogan Ozcan, Sylvain Gigan, Shanhui Fan, Dirk Englund, Marin Soljačić, Cornelia Denz, David A. B. Miller, and Demetri Psaltis. Inference in artificial intelligence with deep optics and photonics. *Nature*, 588(7836):39–47, Dec 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2973-6. URL https://doi.org/10.1038/s41586-020-2973-6.

[10] Mitchell A Nahmias, Thomas Ferreira De Lima, Alexander N Tait, Hsuan-Tung Peng, Bhavin J Shastri, and Paul R Prucnal. Photonic multiply-accumulate operations for neural networks. *IEEE Journal of Selected Topics in Quantum Electronics*, 26:1–18, 2020. URL https://doi.org/10.1109/JSTQE.2019.2941485.

[11] Pascal Stark, Folkert Horst, Roger Dangel, Jonas Weiss, and Bert Jan Offrein. Opportunities for integrated photonic neural networks. *Nanophotonics*, 9(13):4221–4232, 2020. URL https://doi.org/10.1515/nanoph-2020-0297.

[12] Chaoran Huang, Volker J. Sorger, Mario Miscuglio, Mohammed Al-Qadasi, Avilash Mukherjee, Lutz Lampe, Mitchell Nichols, Alexander N. Tait, Thomas Ferreira de Lima, Bicky A. Marquez, Jiahui Wang, Lukas Chrostowski, Mable P. Fok, Daniel Brunner, Shanhui Fan, Sudip Shekhar, Paul R. Prucnal, and Bhavin J. Shastri. Prospects and applications of photonic neural networks. *Advances in Physics: X*, 7(1), October 2021. doi: 10.1080/23746149.2021.1981155. URL https://doi.org/10.1080/23746149.2021.1981155.

[13] Bhavin J Shastri, Alexander N Tait, T Ferreira de Lima, Wolfram HP Pernice, Harish Bhaskaran, C David Wright, and Paul R Prucnal. Photonics for artificial intelligence and neuromorphic computing. *Nature Photonics*, 15(2):102–114, 2021. URL https://doi.org/10.1038/s41566-020-00754-y.

[14] Tianyu Wang, Shi-Yuan Ma, Logan G. Wright, Tatsuhiro Onodera, Brian C. Richard, and Peter L. McMahon. An optical neural network using less than 1 photon per multiplication. *Nature Communications*, 13(1), January 2022. doi: 10.1038/s41467-021-27774-8. URL https://doi.org/10.1038/s41467-021-27774-8.

[15] Alexander Sludds, Saumil Bandyopadhyay, Zaijun Chen, Zhizhen Zhong, Jared Cochrane, Liane Bernstein, Darius Bunandar, P. Ben Dixon, Scott A. Hamilton, Matthew Streshinsky, Ari Novack, Tom Baehr-Jones, Michael Hochberg, Manya Ghobadi, Ryan Hamerly, and Dirk Englund. Delocalized photonic deep learning on the internet's edge. *Science*, 378(6617):270–276, 2022. doi: 10.1126/science.abq8271. URL `https://www.science.org/doi/abs/10.1126/science.abq8271`.

[16] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL `https://arxiv.org/abs/2005.14165`.

[17] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL `https://arxiv.org/abs/2001.08361`.

[18] Aidan Clark, Diego De Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, George Bm Van Den Driessche, Eliza Rutherford, Tom Hennigan, Matthew J Johnson, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Marc'Aurelio Ranzato, Jack Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. Unified scaling laws for routed language models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4057–4086. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/clark22a.html`.

[19] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL `https://arxiv.org/abs/2203.15556`.

[20] Marcos Treviso, Tianchu Ji, Ji-Ung Lee, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Pedro H. Martins, André F. T. Martins, Peter Milder, Colin Raffel, Edwin Simpson, Noam Slonim, Niranjan Balasubramanian, Leon Derczynski, and Roy Schwartz. Efficient methods for natural language processing: A survey, 2022. URL `https://arxiv.org/abs/2209.00099`.

[21] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, June 2022.

[22] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[23] Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. doi: 10.1109/IJCNN55064.2022.9891914.

[24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[26] Jinwoo Kim, Tien Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. Pure Transformers are powerful graph learners. *arXiv*, abs/2207.02505, 2022. URL `https://arxiv.org/abs/2207.02505`.

[27] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/jaegle21a.html`.

[28] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs, 2021. URL `https://arxiv.org/abs/2107.14795`.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/radford21a.html`.

[30] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/ramesh21a.html`.

[31] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=Ee277P3AYC`.

[32] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=1ikK0kHjvj`. Featured Certification.

[33] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. URL `https://openai.com/blog/language-unsupervised/`.

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

[35] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL `https://openai.com/blog/better-language-models/`.

[36] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Pretrained transformers as universal computation engines. *arXiv preprint arXiv:2103.05247*, 2021.

[37] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL `https://arxiv.org/abs/2206.14858`.

[38] Yichen Shen, Nicholas C Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, and Marin Soljačić. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7):441, 2017. URL https://doi.org/10.1038/nphoton.2017.93.

[39] William Andregg, Michael Andregg, Robert T Weverka, and Lionel Clermont. Wavelength multiplexed matrix-matrix multiplier, April 19 2019. URL https://patents.google.com/patent/US10274989B2/en. (U.S. Patent No. 10,274,989). U.S. Patent and Trademark Office.

[40] James Spall, Xianxin Guo, Thomas D Barrett, and AI Lvovsky. Fully reconfigurable coherent optical vector–matrix multiplication. *Optics Letters*, 45(20):5752–5755, 2020. URL https://doi.org/10.1364/OL.401675.

[41] Wim Bogaerts, Daniel Pérez, José Capmany, David A B Miller, Joyce Poon, Dirk Englund, Francesco Morichetti, and Andrea Melloni. Programmable photonic circuits. *Nature*, 586 (7828):207–216, 2020. URL https://doi.org/10.1038/s41586-020-2764-0.

[42] Yoshio Hayasaki, Ichiro Tohyama, Toyohiko Yatagai, Masahiko Mori, and Satoshi Ishihara. Optical learning neural network using Selfoc microlens array. *Japanese Journal of Applied Physics*, 31(5S):1689, 1992. URL https://doi.org/10.1143/JJAP.31.1689.

[43] Charis Mesaritakis, Vassilis Papataxiarhis, and Dimitris Syvridis. Micro ring resonators as building blocks for an all-optical high-speed reservoir-computing bit-pattern-recognition system. *J. Opt. Soc. Am. B*, 30(11):3048–3055, Nov 2013. doi: 10.1364/JOSAB.30.003048. URL https://opg.optica.org/josab/abstract.cfm?URI=josab-30-11-3048.

[44] Alexander N. Tait, John Chang, Bhavin J. Shastri, Mitchell A. Nahmias, and Paul R. Prucnal. Demonstration of WDM weighted addition for principal component analysis. *Opt. Express*, 23(10):12758–12765, May 2015. doi: 10.1364/OE.23.012758. URL https://opg.optica.org/oe/abstract.cfm?URI=oe-23-10-12758.

[45] Changming Wu, Heshan Yu, Seokhyeong Lee, Ruoming Peng, Ichiro Takeuchi, and Mo Li. Programmable phase-change metasurfaces on waveguides for multimode photonic convolutional neural network. *arXiv preprint arXiv:2004.10651*, 2020.

[46] Johannes Feldmann, Nathan Youngblood, Maxim Karpov, Helge Gehring, Xuan Li, Maik Stappers, Manuel Le Gallo, Xin Fu, Anton Lukashchuk, Arslan Sajid Raja, et al. Parallel convolutional processing using an integrated photonic tensor core. *Nature*, 589(7840):52–58, 2021.

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

[48] Xingyuan Xu, Mengxi Tan, Bill Corcoran, Jiayang Wu, Andreas Boes, Thach G Nguyen, Sai T Chu, Brent E Little, Damien G Hicks, Roberto Morandotti, Arnan Mitchell, and David J Moss. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature*, 589(7840): 44–51, 2021. URL https://doi.org/10.1038/s41586-020-03063-0.

[49] Lingling Fan, Zhexin Zhao, Kai Wang, Avik Dutt, Jiahui Wang, Siddharth Buddhiraju, Casey C. Wojcik, and Shanhui Fan. Multidimensional convolution operation with synthetic frequency dimensions in photonics. *Phys. Rev. Appl.*, 18:034088, Sep 2022. doi: 10.1103/PhysRevApplied.18.034088. URL https://link.aps.org/doi/10.1103/PhysRevApplied.18.034088.

[50] Alexander N Tait. Quantifying power use in silicon photonic neural networks. *arXiv:2108.04819*, 2021. URL https://arxiv.org/abs/2108.04819.

[51] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs), 2016. URL https://arxiv.org/abs/1606.08415.

[52] Alex Krizhevsky. Convolutional deep belief networks on cifar-10. 2010. URL https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf.

[53] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.

[54] Hyungjun Kim, Jihoon Park, Changhun Lee, and Jae-Joon Kim. Improving accuracy of binary neural networks using unbalanced activation distribution. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7858–7867, 2021. doi: 10.1109/CVPR46437.2021.00777.

[55] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017. URL `https://openreview.net/forum?id=Byj72udxe`.

[56] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with Pathways, 2022. URL `https://arxiv.org/abs/2204.02311`.

[57] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

[58] Tijmen Tieleman, Geoffrey Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2): 26–31, 2012.

[59] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL `https://proceedings.mlr.press/v9/glorot10a.html`.

[60] Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joseph E. Gonzalez. Train large, then compress: Rethinking model size for efficient training and inference of transformers. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[61] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Understanding and overcoming the challenges of efficient Transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.emnlp-main.627`.

[62] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism, 2019. URL `https://arxiv.org/abs/1909.08053`.

[63] Corby Rosset. Turing-nlg: A 17-billion-parameter language model by Microsoft. *https://www.microsoft.com*, Feb 2020. URL `https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/`.

[64] Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang,

Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530B, a large-scale generative language model, 2022. URL `https://arxiv.org/abs/2201.11990`.

[65] Yaosheng Fu, Evgeny Bolotin, Niladrish Chatterjee, David Nellans, and Stephen W. Keckler. GPU domain specialization via composable on-package architecture. *ACM Trans. Archit. Code Optim.*, 19(1), dec 2021. ISSN 1544-3566. doi: 10.1145/3484505. URL `https://doi.org/10.1145/3484505`.

[66] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017. URL `https://doi.org/10.1109/JPROC.2017.2761740`.

[67] Flavio Ponzina, Miguel Peon-Quiros, Andreas Burg, and David Atienza. E^{2}CNNs: Ensembles of convolutional neural networks to improve robustness against memory errors in edge-computing devices. *IEEE Transactions on Computers*, 70(8):1199–1212, August 2021. doi: 10.1109/tc.2021.3061086. URL `https://doi.org/10.1109/tc.2021.3061086`.

[68] Benoît W. Denkinger, Flavio Ponzina, Soumya S. Basu, Andrea Bonetti, Szabolcs Balási, Martino Ruggiero, Miguel Peón-Quirós, Davide Rossi, Andreas Burg, and David Atienza. Impact of memory voltage scaling on accuracy and resilience of deep learning based edge devices. *IEEE Design & Test*, 37(2):84–92, 2020. doi: 10.1109/MDAT.2019.2947282.

[69] Norman P. Jouppi, Doe Hyun Yoon, Matthew Ashcraft, Mark Gottscho, Thomas B. Jablin, George Kurian, James Laudon, Sheng Li, Peter Ma, Xiaoyu Ma, Thomas Norrie, Nishant Patil, Sushma Prasad, Cliff Young, Zongwei Zhou, and David Patterson. Ten lessons from three generations shaped Google's TPUv4i : Industrial product. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 1–14, 2021. doi: 10.1109/ISCA52012.2021.00010.

[70] Pietro Caragiulo, Clayton Daigle, and Boris Murmann. DAC performance survey 1996-2020. URL `https://github.com/pietro-caragiulo/survey-DAC`.

[71] Mengyue Xu, Yuntao Zhu, Fabio Pittalà, Jin Tang, Mingbo He, Wing Chau Ng, Jingyi Wang, Ziliang Ruan, Xuefeng Tang, Maxim Kuschnerov, et al. Dual-polarization thin-film lithium niobate in-phase quadrature modulators for terabit-per-second transmission. *Optica*, 9(1):61–62, 2022.

[72] Maruf N. Ahmed, Joseph Chong, and Dong Sam Ha. A 100 Gb/s transimpedance amplifier in 65 nm CMOS technology for optical communications. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1885–1888, 2014. doi: 10.1109/ISCAS.2014.6865527.

[73] David A B Miller. Attojoule optoelectronics for low-energy information processing and communications. *Journal of Lightwave Technology*, 35(3):346–396, 2017. URL `https://doi.org/10.1109/JLT.2017.2647779`.

[74] Juzheng Liu, Mohsen Hassanpourghadi, and Mike Shuo-Wei Chen. A 10GS/s 8b 25fJ/c-s 2850um^2 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)*. IEEE, February 2022. doi: 10.1109/isscc42614.2022.9731625. URL `https://doi.org/10.1109/isscc42614.2022.9731625`.

[75] Sony Corporation. *LCX027AKB Datasheet (PDF) - Sony Corporation*. Sony Corporation. URL `https://pdf1.alldatasheet.com/datasheet-pdf/view/47550/SONY/LCX027AKB.html`.

[76] Matthias Wuttig, Harish Bhaskaran, and Thomas Taubner. Phase-change materials for non-volatile photonic applications. *Nature photonics*, 11(8):465–476, 2017. URL `https://doi.org/10.1038/nphoton.2017.126`.

[77] Boris Murmann. ADC performance survey 1997-2020. `http://web.stanford.edu/~murmann/adcsurvey.html`, 2020. Online Accessed: 2021-02-18.

[78] Saumil Bandyopadhyay, Alexander Sludds, Stefan Krastanov, Ryan Hamerly, Nicholas Harris, Darius Bunandar, Matthew Streshinsky, Michael Hochberg, and Dirk Englund. Single chip photonic deep neural network with accelerated training. *arXiv preprint arXiv:2208.01623*, 2022. URL `https://doi.org/10.48550/arXiv.2208.01623`.

[79] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *arXiv:2211.05102*, 2022.

Table 1: Model configurations for optical Transformers. $\mathrm{M} = 10^6$.

Model	n	d	h	L	Non-emb. Params
Tiny	1024	192	12	12	15M
Small	1024	384	12	12	40.6M
Base	1024	768	12	12	123.7M
Large	1024	1536	12	12	416.3M

Table 2: Pretraining hyperparameters for optical Transformer models. All models were trained with the **AdamW** [57] optimizer.

Model	Steps	Batch	lr	β_1	β_2	ϵ	Weight decay	Dropout	Schedule	Warmup	Stop
Tiny	90000	32	2e-4	0.9	0.999	1e-8	0.02	0.1	Cosine	2500	-
Small	90000	32	2e-4	0.9	0.999	1e-8	0.02	0.1	Cosine	2500	-
Base	90000	32	2e-4	0.9	0.999	1e-8	0.02	0.1	Cosine	2500	-
Large	90000	32	2e-4	0.9	0.999	1e-8	0.02	0.1	Cosine	2500	82500

A Appendix

B Optical Transformer Training Hyperparameters

The optical Transformer models were pretrained on the Wikitext-103 [55] dataset and used the same tokenizer as GPT2 [35]. All models used **Xavier uniform initialization** [59]. The architectures are in Table 1. Embedding layers were initialized with a normal distribution with $\sigma = 0.02$. We used the AdamW [57] optimizer, with weight decay applied to parameters which were not embedding, gains, or biases. Dropout was applied after every linear layer (including those in attention), as well as on the attention matrix and after the softmax$(\frac{QK^T}{\sqrt{d_h}})V$ product in the attention calculation. The values of the parameters used for the training scheme are in Table 2.

After pretraining the models were quantized via our 8-bit QAT scheme. For QAT we used the RMSProp optimizer [58]. The parameters we used for the training are in Table 3. To clamp weights and activations we employ two different approaches: first, we kept running statistics of minimum and maximum values with an exponential moving average (EMA, with parameter α) for every layer and use those to clamp. Second, we recorded the minimum/maximum statistic throughout the network for a forward pass to apply a clipping scheme. Specifically, we clamped weights and activations to percentiles of the maximum values collected for each layer. The outputs were either rounded to the nearest integer during QAT, or stochastically rounded to nearby values. Finally, for the Base-sized model we used to run the experiments, we directly used the lookup tables (LUT) instead of "simulating" the quantization of inputs and weights (though outputs are still quantized). Table 4 details our use of these various techniques in the models.

For evaluation we used the perplexity (PPL) metric to measure the language modelling performance on Wikitext-103. We evaluated the perplexity over the entire validation set, and ran the model with context length 1024 (the same as in training) and a 1024-token stride length.

C ONN Experimental Procedure

C.1 Experimental Setup

Our setup is a SLM-based matrix-vector/vector-vector multiplier. The setup is shown in Figure 6 with a simplified illustration in Figure 7, and works as follows: Vectors corresponding to the inputs and weights are rearranged into squares of pixels and loaded onto the display and SLM respectively. They are aligned such that the light from display pixels will reach the corresponding pixels on the SLM. First, light from the display enters into the polarizing beam splitter (PBS), and reaches the SLM through a half-wave plate (HWP) which rotates its polarization. The phase is then modified by the SLM and reflected back through the half-wave plate, rotating the polarization again based on the phase difference. Then, the PBS only admits light of a certain polarization along one of its arms, aimed at a camera for detection. Summation of the output pixels is performed digitally. This

Table 3: Quantization aware training hyperparameters for optical Transformer models. All models were trained with the **RMSProp** [58] optimizer. Quantization parameters are in Table. 4.

Model	Steps	Batch	lr	α	ϵ	Weight decay	Dropout	Schedule	Warmup	Stop
Tiny	7327	64	1e-5	0.99	1e-8	1e-5	0.1	Cosine	2500	-
Small	7327	64	1e-5	0.99	1e-8	1e-5	0.1	Cosine	2500	-
Base	7327	64	1e-5	0.99	1e-8	1e-5	0.1	Cosine	2500	5500
Large	7327	32	1e-5	0.99	1e-8	1e-5	0.1	Cosine	2500	5500

Table 4: Hyperparameters for optical Transformer Quantization. We perform QAT with both a percentile-clipping approach and by clamping based on an exponential moving average (EMA) of model statistics with factor γ. For the Base-sized model that is used in our experiments (LUT-Base), we use lookup tables (LUT) for inputs and weights instead of quantization.

	Overall Config		EMA	Attention Clipping			Feed-Forward Clipping		
Model	Precision	Rounding	γ	$\texttt{Input}_1$	$\texttt{Input}_2$	Output	Input	Weights	Output
Tiny	8-bit	Stochastic	-	99.99%	99.9%	99.9999%	99.99%	99.9%	99.9999%
Small	8-bit	Stochastic	-	99.99%	99.9%	99.9999%	99.99%	99.9%	99.9999%
Base	8-bit	Stochastic	-	99.99%	99.9%	99.9999%	99.99%	99.9%	99.9999%
Large	8-bit	Stochastic	-	99.99%	99.9%	99.9999%	99.99%	99.9%	99.9999%
LUT-Base	LUT	Stochastic	-	99.99%	98%	99.9999%	99.99%	99%	99.9999%
Tiny	8-bit	Deterministic	0.999	-	-	-	-	-	-
Small	8-bit	Deterministic	0.999	-	-	-	-	-	-
Base	8-bit	Deterministic	0.999	-	-	-	-	-	-
Large	8-bit	Deterministic	0.999	-	-	-	-	-	-

SLM–HWP–PBS arrangement effectively creates an amplitude modulating SLM, where the output at each pixel is the element-wise product of the input pixel and corresponding weight pixel.

The OLED display has multiple color channels and a broad spectrum. For easier modulation by the SLM, we used a band-pass filter and only green light.

The components we used are:

- Organic light-emitting diode (OLED) display (Google Pixel 2016)
- Reflective liquid-crystal modulator (1920-500-1100-HDMI, Meadowlark Optics)
- Half-wave plate (PH10ME-532, Thorlabs)
- Polarizing beam splitter (CCM1-PBS251, Thorlabs)
- Zoom lens for imaging onto SLM (Resolv4K, Navitar)
- Zoom lens and objective lens for imaging onto detector (1-81102, Navita and XLFLUOR4x/340, Olympus)
- Band-pass filter (FF01-525/15-25, Semroc)
- Camera for detection (Prime 95B Scientific CMOS Camera, Teledyne Photometrics)

This setup works as a good bench for testing the precision of optical Transformers by performing optical dot products involved in attention and feed-forward layers. Even though the optical dot products were performed one at a time, it is sufficient for showing that Transformer operations can run with the accuracy of ONNs, since matrix-vector and matrix-matrix products are merely collections of many dot products run in parallel.

C.2 Calibration and Lookup Tables

We used several techniques to reduce errors, map inputs to SLM/display values, and to convert detected outputs back to neural network values.

First, we developed a specialized data-pixel encoding scheme to reduce systematic errors. We noticed that a large source of error was with a limitation of our hardware—in particular the SLM pixels have cross-talk (pixels may affect their neighbors if they have very different values) and misalignment in the experimental setup may lead to corrupted outputs. To help with these issues, we created "macropixels"—each input element (and weight) does not occupy one pixel on the display (SLM) but rather is mapped to a 3x3 grid of pixels, all with the same value. For the attention layers, we used 5x5

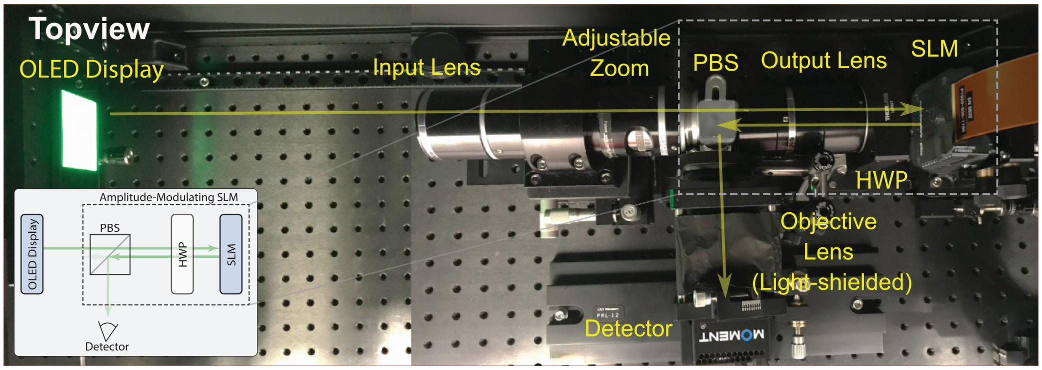

Figure 6: Photo of experimental setup used for running Transformer dot-product operations. Inset: simplified illustration of the experimental system. Spatial light modulator (SLM) + half-wave plate (HWP) + polarizing beam splitter (PBS) arrangement is effectively an amplitude-modulating SLM. The system works as follows: in our experiments, a vector is loaded as pixels on the organic light-emitting diode (OLED) display, and weights on the SLM. The input light enters through the PBS towards the SLM, passing through the HWP twice as the SLM reflects it. The SLM and HWP together rotate the polarization of the light, such that the amount reflected by the PBS towards the detector for each pixel is roughly the product between the pixel value and the corresponding weight on the SLM. The summation of these element-wise products by the detector yields the dot product.

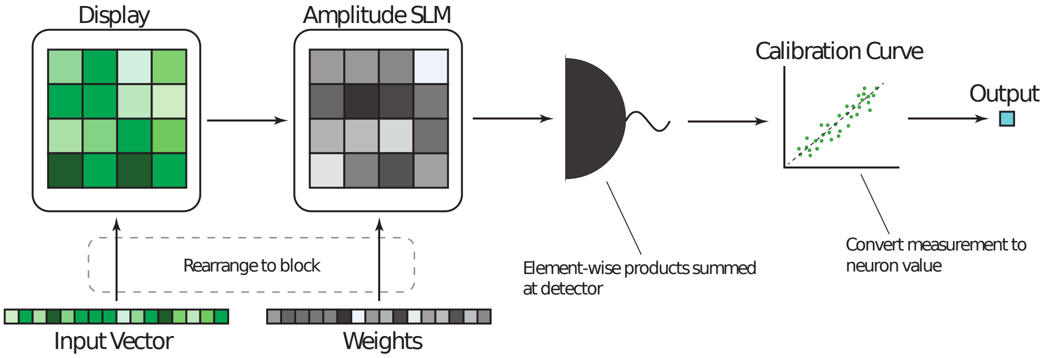

Figure 7: Simplified illustration of experimental setup operation. Weights are loaded and rearranged into a block on spatial light modulator (SLM) to prevent crosstalk between pixels of drastically different values. Data is rearranged on display accordingly. Measurements are looked up against calibration curve to obtain the final output value.

macropixels for the results we report, but later discovered that with 3x3 the performance is essentially the same. We also rearranged vectors into square blocks of pixels so that significantly nonzero weights are nearby each other Figure 7. For the vectors to better fit in the center of the field of view (where there is less distortion/misalignment) we computed the dot products using only the 400 largest weight elements (the corresponding input elements are loaded). While this may introduce some inaccuracy in the final results, we found that the benefits of computing the element-wise products more accurately outweigh the drawbacks of pruning the weights; the outputs were still quite accurate to the ground-truth dot-product values (see main text, Figure 3). We suspect that this was the case because:

- Transformer weights are not entirely dense; some weights were already zero.
- Because our setup only supports non-negative data anyway, we use the four-pass approach (Section 3.3, main text). This means that for any given dot product, roughly half the weights and activations will be zero before considering the previously mentioned sparsity.
- Meanwhile, a second consequence of this four-pass approach is that roughly half of activations will be zero as well, possibly rendering some of the pruned weights irrelevant.

- Transformers still perform well when pruned, and luckily larger models can be pruned more heavily [60]. While our pruning method is quite basic, the number of weights pruned was light (ie. $< 75\%$) compared to what is possible with more advanced methods.

This approach was not necessary for attention calculations, since the dot products were sufficiently small to fit them entirely (64 elements).

Next, we consider the lookup tables (LUTs) of the display and SLM in the setup. In order to optimize the experimental results, the model used for experiment was trained to be aware of the realistic, discrete mappable values supported by the system. The display has a LUT with 256 unique levels (1000 levels total, but many are the same as others) and the SLM has roughly 128 unique levels (256 total). So they are roughly capable of 7 and 8 bit precision. The SLM also cannot fully extinguish input light—the minimum modulation is 2% of the maximum transmission. Thus, the minimum absolute values of the weights were mapped to 0.02 instead of 0.

After applying these approaches, we finally collected the calibration curve, which maps the output intensity measurements to neuron values in the neural network and allows us to determine the experimental setup's systematic error. To do this we sampled randomly inputs and outputs of the layers we wished to run, computed their dot products both digitally and in experiment, and created a data set of experimental measurements and ground-truth-digital dot-product outputs. We then performed linear regression to find a mapping between experimental output and the correct values, effectively creating another lookup table. Then when future dot products were computed experimentally, the output was passed to this linear regression model (or it can literally be stored as a lookup table) to get the output. We used many photons and averaged outputs across multiple shots for each input, eliminating shot noise—any remaining error in this calibration scheme we defined as the system's systematic error.

It is important to note that in general other optical systems might have different causes of error from ours, but the overall accuracy of our system is representative of a typical ONN nowadays.

C.3 Model Design Optimization

Transformers tend to have large dynamic ranges in their activations and weights [61]. In particular, we found that systematic error is proportional to some characteristic amplitude of the output. So, because it scales roughly with the sizes of outputs, having large outlier values can increase the systematic error and worsen the calibration for all other values in the representable range. Furthermore, after quantization in a naive, linear scheme, large outliers mean that huge ranges of outputs which are seldom used are assigned to many of the quantization levels, while the rest of the small, common outputs are squashed into few buckets—so the model precision is poor. This can be an issue when quantizing any deep learning model, but was exacerbated here by those systematic errors and the fact that the lowest levels of the weights are 0.02 and not 0.0. Therefore, we opted for an aggressive clipping scheme and the clamped activation ReLU6 when training the model to be run (Appendix B, **LUT-base** model); they reduce the dynamic range of inputs and weights and we found that they drastically improved the ONN's ability to run Transformer operations with smaller error. Having fewer values in the 0.02 bucket of the SLM LUT also improved QAT training stability significantly. Even though the non-zero light extinction at 0.02 is caused by the specific SLM in our setup, such issues may happen with other optical implementations made of elements with finite extinction or resolution, and here we described a method to mitigate such issues by modifying training methods.

C.4 Transformer Dot Product Samples

While the speed and parallelism limitation of our setup made it intractable to run an entire Transformer model on it, we attempted to sample dot products to run that were representative of the range of possible activation/weight statistics in the model. That way, our results would be very representative of what running the full model would be like. In particular, we found two ways in which statistics throughout the model vary: the statistics change with depth (shallow and deep layers behave differently) and operation type (matrix-matrix multiplication in attention has different statistics from MLP layers). So, given our limited ability to run operations on the setup, we sampled roughly 10000 dot products from the first (QK^T) attention operation and second MLP layer of the first and last encoder layers of the model. The inputs to the whole model were samples from the Wikitext-103 dataset. Our approach captures the range of statistics throughout a model's different components, over its depth,

Table 5: Simulated optical Transformer precision ablation. Input precision is degraded by subsampling from lookup table (LUT), while output is quantized. Input precision is approximate, as LUT has 1000 levels, not 1024. Bold: most compressed model found in our ablation with performance very close to the baseline.

Input Precision (LUT)	Output Precision	Val. Loss
$\sim$ 10 bits	32 bits	3.0059
$\sim$ 9 bits	32 bits	3.0057
$\sim$ 8 bits	32 bits	3.0054
$\sim$ 7 bits	32 bits	3.0039
$\sim$ 6 bits	32 bits	3.0034
$\sim$ 5 bits	32 bits	3.0017
$\sim$ 4 bits	32 bits	3.0111
$\sim$ 3 bits	32 bits	3.1223
$\sim$ 5 bits	8 bits	3.0032
$\sim$ 5 bits	**7 bits**	**3.0074**
$\sim$ 5 bits	6 bits	3.0335
$\sim$ 5 bits	5 bits	3.3966

and when processing a real task's data. The second MLP layer has dot product size $4d$, making it the hardest to run experimentally.

In sampling the dot products, we tried to sample from both operands equally. For example, one could sample 1000 dot products by taking a single input vector and 1000 weight matrix vectors, and vice-versa, but choosing random vector pairs captures dot products involving different tokens and weights. This is important because Transformer output sizes, particularly the outlier activation values, are token-dependent [61]. To maintain this balance, we sample equal rows/columns for both operands. For attention layers we sample 100 from each; For linear layers, we sampled 56 rows from the input data and 200 columns from the weight matrix W^T, where the product being computed is xW^T.

D Simulated Precision Ablation Study

To further study how the optical Transformer can perform inference at lower precisions, we conducted a simple ablation on the input and output precisions used at inference, on the 8-bit-QAT base-sized model with LUT. We opted to leave the weights at 8-bit precision, since in-place weights are not a significant energy cost, and do not take more space/memory in these analog optical systems. In Table 5 is the performance of the model at lower precisions. With 5-bit input and 7-bit output precision, the model performs as well as the baseline. The reported precision values for the LUT are approximate, since the LUT has 1000 levels instead of $2^{10} = 1024$ levels.

When using the LUT, it is also not possible to directly change the precision of the input. Instead, we employed a subsampling scheme where the precision is degraded by rounding to every n'th integer level before using the LUT, where n is a power of 2 and represents a reduction in the effective bit precision. The LUT of our display has 1000 levels, some levels have the same value, and we simulate the model without added noise. So we say that the original precision is initially *at most* 10 bits ($2^{10} = 1024$).

E ONN Energy Calculation

The models we used to estimate the energy use of ONN systems are in Table 6. We used a variety of real models that have been introduced by other works, and then designed our family of hypothetical future models **FUTURE-*** in a similar fashion, keeping a reasonable sequence length, increasing the embedding dimension drastically, and following the trend of recent large models like PaLM [56] and MT-NLG [64] of increasing the ratio d/h, which results in favorable energy calculations due to the lower fraction of memory operations in attention.

The calculation of energy costs for ONNs requires consideration of the entire system design and the costs of the surrounding electronics—since the optical computation itself is so cheap the electronics

Table 6: Designs of models used for energy estimates. Transformers have embedding dimension d, process sequence length n, use h attention heads, and have L layers. $\mathrm{M} = 10^6$ parameters.

Model	n	d	h	L	Parameters	Reference
GPT2	1024	768	12	12	117M	[35]
GPT2	1024	1024	16	24	345M	
GPT2	1024	1280	20	36	762M	
GPT2	1024	1600	25	48	1.5B	
Megatron	2048	1536	16	40	1.2B	[62]
Megatron	2048	1920	20	54	2.5B	
Megatron	2048	2304	24	64	4.2B	
Megatron	2048	3072	32	72	8.3B	
GPT3	2048	768	12	32	125M	[16]
GPT3	2048	1024	16	24	350M	
GPT3	2048	1536	16	24	760M	
GPT3	2048	2048	24	24	1.3B	
GPT3	2048	2560	32	32	2.7B	
GPT3	2048	4096	32	32	6.7B	
GPT3	2048	5140	40	40	13B	
GPT3	2048	12288	96	96	175B	
Turing-NLG	1024	4256	28	78	17B	[63]
MT-NLG	2048	20480	128	105	530B	[64]
Chinchilla	2048	640	10	10	73M	[19]
Chinchilla	2048	1024	16	20	305M	
Chinchilla	2048	1280	10	24	552M	
Chinchilla	2048	1792	14	26	1.1B	
Chinchilla	2048	2048	16	28	1.6B	
Chinchilla	2048	3584	28	40	6.8B	
Chinchilla	2048	8192	64	80	70B	
PaLM-like	2048	4096	16	32	8B	[56]
PaLM-like	2048	8192	32	64	62B	
PaLM-like	2048	18432	48	118	540B	
FUTURE	2048	40960	80	120	2.4T	This work
FUTURE	2048	81920	128	200	16T	
FUTURE	2048	163840	160	400	129T	
FUTURE	2048	655360	512	800	4q	

account for nearly all of the energy cost. The way the energy is accounted for is as follows: The energy E_{load} can be broken down into three components, related to the energy of the cost of reading from memory E_{read}, digital-to-analog conversion (DAC) E_{DAC}, and modulation to generate the light E_{mod}:

$$E_{\mathrm{load}} = E_{\mathrm{read}} + E_{\mathrm{DAC}} + E_{\mathrm{mod}}. \tag{1}$$

Detection energy consumption E_{det} can broken down in a similar fashion, where

$$E_{\mathrm{det}} = E_{\mathrm{detector}} + E_{\mathrm{amp}} + E_{\mathrm{ADC}} + E_{\mathrm{write}} \tag{2}$$

represent the costs of detecting a signal, amplifying the detected signal, performing analog-to-digital conversion, and writing to memory respectively. There is also a cost of maintaining the weights in a weights-in-place system, which we call E_{maintain}. Because this cost scales per element, it is a per-MAC cost. But based on values from efficient commercial SLM systems, it is sufficiently small (and amortized by a large clock rate) that even the largest models we do estimations for are not bottlenecked. For optical energy, we take $1\,\mathrm{eV}$ (single-photon energy at $1240\,\mathrm{nm}$). We started with using our measured 8-bit-performance photon count of 1500/MAC for the smallest model ($d = 192$) and rescaled the value for larger ones using the constant-per-dot-product trend which we know our simulated models can match or beat.

The assumptions we used were that weights would be loaded from off-chip memory like DRAM (in the case of a chunked-weights strategy; for a full weights-in-place, one-shot approach this cost does not exist), and that the system uses large amounts of SRAM for activations [65]. We assumed that the system only needs 5 bits worth of input precision and 7 bits worth of output precision, per the results of our ablation on the base-sized model. We still assumed 8-bit memory accesses for convenience. The actual costs for the data access and weight maintenance were assumed to be these values:

- E_{read} = 1 pJ/bit for off-chip memory [66], and 0.3 pJ/bit for SRAM. The SRAM estimate is based on results for DNN accelerator measurements with 9.55 pJ/32-bit access [67, 68], and cutting edge/near-future assumptions for data transport from SRAM/cache [65]. [69] estimates 14 pJ per 64-bit access, or roughly 0.22 pJ/bit, for a recent TPU architecture.
- E_{DAC} = 10 pJ per 5-bit sample @ $10\,\text{GHz}$—this is achievable with $100\,\text{mW}$ at 30.1dB SFDR [70].
- E_{mod} = 1 fJ/bit @ $110\,\text{GHz}$ with thin-film lithium-niobate modulators [71].
- E_{amp} = 2.4 pJ per access. A transimpedance amplifier can run at 24 mW at 70 GHz [72]. We will just assume 10 GHz. 24mW / 10^{10} = 2.4 pJ per element.
- E_{detector} is negligible compared to E_{amp}. For example, [73] calculates the cost of detection as the capacitive discharge, $\frac{1}{2}CV^2$, with capacitance $C \sim 1$ fF and voltage $V = 0.5$ V. This results in <500 aJ of energy consumption per detection. The cost is therefore negligible compared to amplification (E_{amp}).
- $E_{\text{ADC}} = 3.17\,\text{pJ}$ per 7-bit sample. 10 Ghz, need 7-bits of precision, so 128 conversion steps per sample – Achievable with 24.8 fJ/c-s [74] (24.8 fJ × 128 = 3.17 pJ per 7-bit sample).
- $E_{\text{write}} = E_{\text{read}}$. Actually, write access was measured to be cheaper than read access in [68], but we use $E_{\text{write}} = E_{\text{read}}$ as a simple, conservative assumption.
- $E_{\text{maintain}} = 0.002\,\text{fJ/MAC}$. Assuming 2W for operation of a 10MP SLM, with inputs shone at 10 GHz (each pixel performs one MAC every cycle). There is not much information SLM power consumption for maintenance of a fixed pattern on the LCD panel, though more typical LCD displays which update can operate in the ∼1W regime. For example, [75] consumes $30\,\text{mW}$ with 180000 pixels, which would scale to $1.67\,\text{W}$ with 10MP (at worst, multiple SLMs/LCDs could be used in order to scale up).

F Breakdown-Of-Costs For Estimated ONN Energy Usage

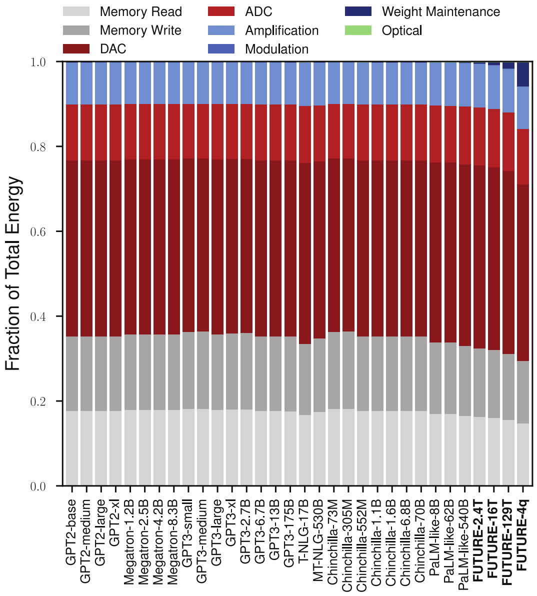

Figure 8: Breakdown of optical Transformer energy costs by energy type at 8-bit operation. Data access costs are dominant due to the high costs of DAC/ADC, but weight maintenance becomes important for large models.

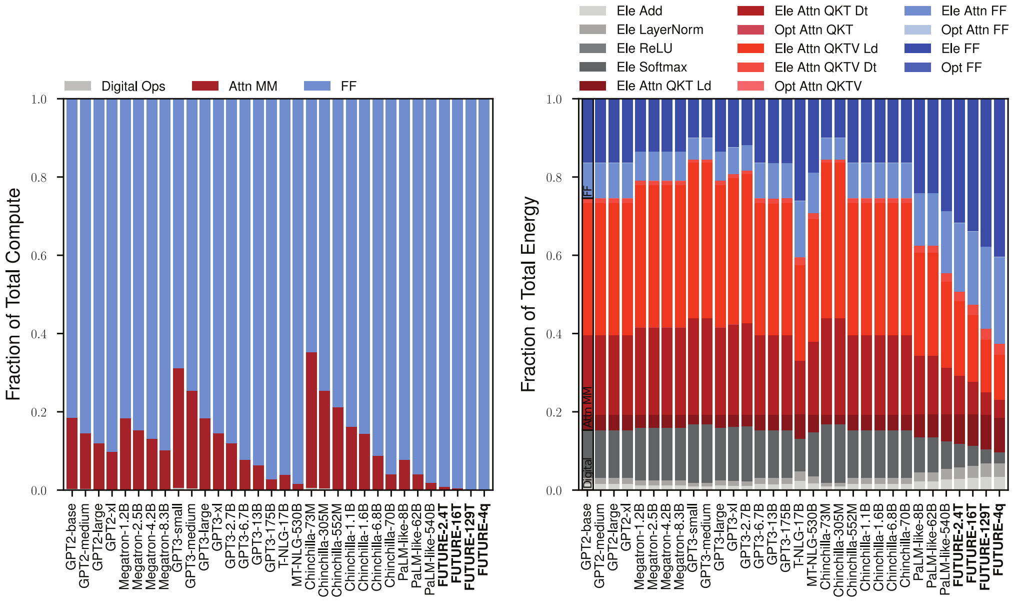

Figure 9: Breakdown of computing costs for optical Transformer models. Left: fraction of total compute used by digital operations, attention, and feed-forward components. Feed-forward layers account for most of the compute. Right: breakdown-of-costs for models by layer. The energy costs of attention operations is expensive. “Ele *” operations: electrical costs of loading (Ld), detecting (Dt), or both for data for the operation. Operations related to attention computation (ie. QK^T) are expensive for little compute. Functions computed digitally have their energy costs estimated as the cost of reading and writing to memory the required data.

In Figure 8 we see that data access costs, that is costs per element loaded/stored in memory, are most expensive. In particular, the cost of ADC and DAC are the leading contributors to the access costs, though since their cost is exponential in the bit precision, one might imagine that a future, optimized Transformer running at lower precision than our assumptions would have energy costs dominated by the actual SRAM memory costs. Also, for very large models, since the energy from weight maintenance scales with the number of MACs, it eventually will dominate if model sizes scale past that of FUTURE-4q. But future hardware would reduce E_{maintain} through improved electronics or higher clock speeds allowing for lower energy per MAC. Finally, the contribution from optical energy is vanishingly small, a consequence of the efficient photon usage scaling that we found Transformers can leverage. Were it not for this, the cost of actually performing the MACs would be orders of magnitude larger than everything else, resulting in energy usage that scales the same way as digital systems’.

Breaking down the sources of compute and energy costs in Transformer models running optically further illustrates how aspects of model/system design affect energy usage. The breakdown of compute and energy costs by source is in Figure 9. We find that as models get larger the feed-forward layers require most of the computation, but that the energy of data access in attention is still very expensive. This is because of the need to save/load many attention matrices from memory, and the fact that a weights-in-place scheme cannot be used for the matrix-matrix products because the products are between activations. Of course, this also means that there are more activations to load. In total, this means that attention layers have high energy costs for small amounts of computation. Thankfully, and interestingly, existing model design trends have moved towards focusing much harder on feed-forward layers, and so for the largest real (and our hypothetical future) models the fraction of energy cost taken by attention is low. Finally, we note that the operations we assume run on digital computers - such as nonlinear functions, in gray - do not account for much of the total energy cost (though they too are a small fraction of the total compute).

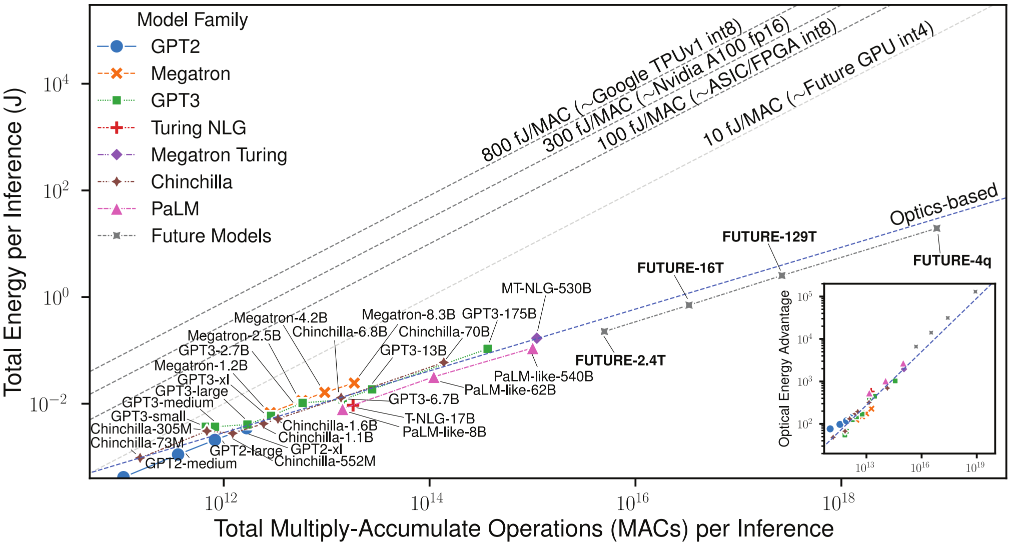

Figure 10: Energy usage estimates of forward pass for Transformers running on optical hardware, under future electronics energy cost assumptions. The energy advantages over our estimate for the current-day NVIDIA A100 GPU are larger than under our original assumptions (main text, Figure 5). $\mathrm{M} = 10^6$, $\mathrm{G} = 10^9$, $\mathrm{T} = 10^{12}$, $\mathrm{q} = 10^{15}$ parameters.

G Future ONN Energy Consumption

As optical accelerators are an emerging technology and as Transformer models continue to scale over time, it is worth considering how ONNs might improve over the next several years. For example, an interesting question to ask is how well future ONNs will do by the time it is possible to run a large model like FUTURE-4q. To investigate this, we estimated the energy costs of various Transformer models running optically again, but with the following changes and assumptions:

- $E_{\text{maintain}} = 0$—Future weights-in-place hardware will need effectively no energy to maintain weight information (for example, one might consider the usage of phase change materials [76]).

- E_{DAC} and E_{ADC} are 1/32 the size—we assume that electronics could achieve a $2\times$ improvement in fJ/c-s efficiency, while future advancements in model compression allow for 4-bit Transformer models, which are much cheaper since DAC and ADC costs scale exponentially with the number of bits [77].

- E_{read} and E_{store} are 1/5 the size—there is already a growing recognition of the fact that AI accelerators will need high efficiency and large quantities of SRAM and DRAM in the future [65, 3].

- E_{amp} $10\times$ cheaper (there are already cheaper trans-impedance amplifiers than our conservative estimate here, and receiver-less configuration without any amplifier has also been demonstrated [78]).

Under these assumptions, ONNs become far more efficient, highlighting that improvements to electronics will impact ONNs, and not just competing digital hardware. The energy scaling (Figure 10) is shifted downward for optics compared to under our previous assumptions, leading to over $1900\times$ and $130,000\times$ advantages over the current A100 GPU for MT-NLG and FUTURE-4q models respectively. Of course, by the time this is possible, GPU efficiency will have improved significantly as well, and we are comparing a 4-bit accelerator to the 16-bit performance of the A100. It is difficult to predict the future efficiency of GPUs at lower precision, but it is clear that ONNs can benefit from improvements to electronics and low-precision inference.

H Scaling ONNs: System Specifications and Communication Costs in Multi-Processor and Memory-Constrained ONN and GPU Setups

Implementation of a real ONN for large models might be difficult because the amount of hardware needed to maintain all the weights is exceedingly large. In Table 7 are the requirements for hardware to run the largest future model. To compute the number of weights/elements, we selected the largest MLP layer in the model, since that requires the most space for weights and activations. While detector and memory requirements are achievable, the number of required cores—each an optical component capable of performing 10M multiplications with weights—is enormous. There are some approaches to remedy this kind of memory issue in both GPUs and ONNs, and we are interested in their hardware-time-energy tradeoffs for ONNs.

One solution is to introduce chunking, where only a portion of the weights are loaded at a time, and the inputs are passed through. Then, the amount of time it takes to run is increased by a factor of the number of chunks. This also impacts the optical system's energy advantage over digital ones in two ways. First, the weights must be loaded, but the cost can be amortized via reuse with batched inference. This comes at the expense of latency. This is a new kind of tradeoff, since digital systems cannot reuse weight data for free. Second, for each weight chunk, all inputs must be reloaded; changing the chunk number trades energy efficiency for lower hardware requirements. These energy tradeoffs are illustrated in Figure 11; other factors dominate energy usage until the chunk number is large and chunking becomes the bottleneck.

Realizing large models with GPUs will likely also require a multi-GPU strategy, which will incur overhead over the peak performance of a single GPU. We find that with a simple model of communication costs—modelling the activation reloading in both GPU and ONN systems—that ONNs can retain some of their advantages, dependant on how much system memory (or maximum number of weight elements) is available per-processor. We created a simple model to estimate the cost of this approach in GPU systems. In GPU systems, instead of splitting a model over time, the model may instead be split over multiple GPUs. This introduces an analogous tradeoff to the activation reloading in ONNs due to communication costs: if each GPU holds some chunk of weights, then after every layer, the outputs of multiplying the inputs with each chunk must be broadcasted to every GPU in an all-to-all fashion. This is in essence an all-reduce operation—after every layer, the outputs from all GPUs must be copied onto all GPUs. In total, this means the total number of activations is loaded k times, where k is the number of GPUs. As a crude but conservative estimate of these costs, we modeled this by taking the cost of running the entire model on one GPU, and then adding the energy cost of loading the activations from DRAM, multiplied by the number of chunks (GPUs). This is likely an underestimate, as broadcasting data across GPUs in a real setup requires sending data electronically over much longer distances than required for DRAM access, which would be expensive.

To determine the number of chunks, we tested multiple assumptions about device memory. We assumed a value for the amount of memory that can be used to store weights and take the total number of weights for each model divided by this memory capacity to determine the number of chunks to be used.

With these models, we found that too much chunking is detrimental to ONN performance, but that there is still some energy advantage to be had if it is used sparingly (Figure 11). In Figure 12 (top) are the energy cost estimates assuming a fixed memory of 100M weights (ie. 100MPixel SLM, or RAM with 100MB capacity if each weight is one byte). We assumed that for GPU, the cost of communication is at least that of DRAM-level communication due to the physical distances between GPUs. The curves for GPUs bend upward as the communication costs begin to take over, as do the largest models running optically. The ONNs still maintain an advantage, but the advantage stops growing with model size. Looking at the energy advantage illustrates this idea more clearly: up to a certain model size the advantage is increasing, then as the model size reaches the memory limit it begins to level off, and then the advantage begins to shrink as the cost of chunking takes over. For a small range of model sizes near this peak, the advantage is maintained, suggesting that a small amount of chunking may be useful before it quickly diminishes the energy advantage.

The optimal configuration for ONNs, obviously, is to have enough memory (cores which have weights fixed in place) so that chunking is not necessary. When plotting the advantages for larger memories (and therefore fewer chunks), the advantage gets better, and larger models become worthwhile to

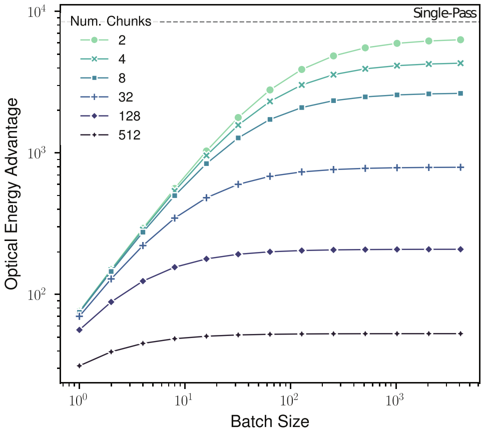

Figure 11: Optical energy advantage vs A100 (FUTURE-4q). When chunking, the cost of loading weights is amortized by increasing batch size, but the overall performance is limited by large numbers of chunks because of input data reloading.

run. In hindsight this conclusion makes sense: the benefit of ONNs is their ability to copy data ("optical fan-out") for free for parallel computation, and so reducing this in favor of repeated memory accesses removes exactly the mechanism that gives optics-based systems their advantages. This also suggests that an "optical memory" from which fixed data can be accessed for free (or significantly less than re-access through electronics) may solve this problem, allowing for more scalable ONN design without huge amounts of hardware for weights. Currently, optics still has an advantage when using multiple cores because in principle the data could be fanned out across cores, while GPUs must pay communication costs in multi-processor setups. With a fan-out/fan-in design that can collect/spread a vector across cores, the efficiency of an entirely weights-in-place system is fully that of a single, large core.

Comparison to Language Model Caching Techniques Transformers running autoregressive language modelling at inference time may utilize caching techniques (such as KV-cache in attention) to speed up and save computation for inference. However, such mechanisms also use exorbitant amounts of memory, and requires offloading to off-chip memory or farther-away memory [79], each of which is far more expensive per bit than SRAM. It is difficult to estimate the energy consumption in these scenarios, but Transformers with unrestricted attention (such as for masked language modelling [34], vision transformers [24], etc.) must perform the full computation in a single forward pass anyway.

Table 7: Requirements for optical accelerator running feed-forward layer (embedding dimension d, sequence length n) without chunking at 8-bit precision. The requirement of many cores to maintain weights for matrix-vector products (MVM) is high, and we assume the ONN system requires static RAM (SRAM) for saving and loading activations.

Model	Input Vector Elements	Detectors	MVM Cores (10^7 weights each)	SRAM (activations)
FUTURE-4.1q	2.6×10^6	2.6×10^6	170,000	5.37 GB
FUTURE-129T	6.55×10^5	6.55×10^5	11,000	1.34 GB
FUTURE-16T	3.28×10^5	3.28×10^5	2,700	671 MB
FUTURE-2.4T	1.64×10^5	1.64×10^5	671	336 MB
PaLM-like-540B	7.37×10^4	7.37×10^4	136	151 MB
MT-NLG-530B	8.19×10^4	8.19×10^4	168	168 MB
GPT3-175B	4.91×10^4	4.91×10^4	61	100 MB
General	$4d$	$4d$	$4d^2/10^7$	$4nd$

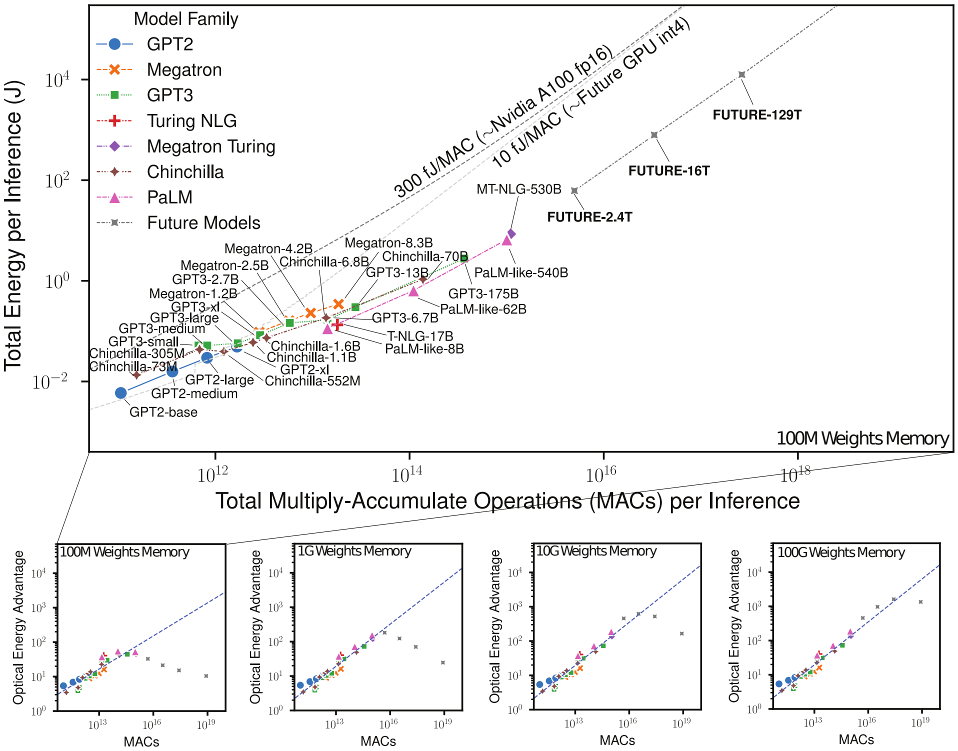

Figure 12: Energy estimates assuming a fixed processor memory size and chunking. Top: estimated energy scaling plot for Transformer models running on optical and digital hardware with 100MB of memory. As models get larger, both optical and digital systems have an upward bend in energy consumption trends, driven by communication/input-reloading-from-chunking costs. Bottom: energy advantage scaling for different memory sizes. As the memory increases, there is a maximum energy advantage for optics over NVIDIA A100 and corresponding model size before chunking costs take over. $\mathrm{M} = 10^6$, $\mathrm{G} = 10^9$, $\mathrm{T} = 10^{12}$, $\mathrm{q} = 10^{15}$ parameters.

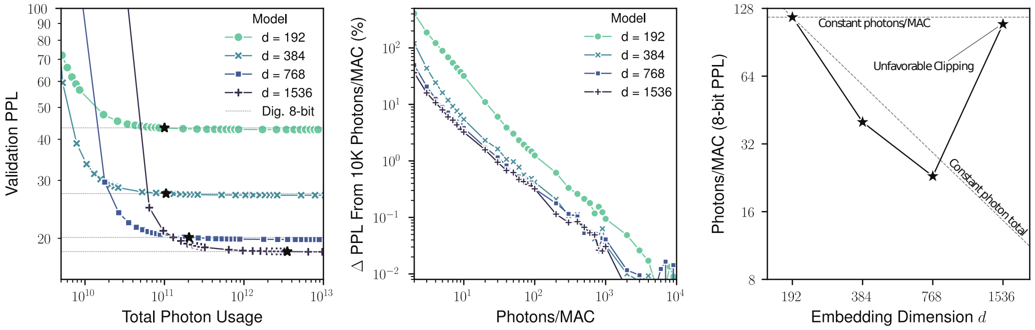

Figure 13: Behavior of optical Transformer models with varying photon usage with percentile clipping scheme. Left: Wikitext-103 validation set perplexity (PPL) versus embedding dimension d and total photons usage. 8-bit quantized digital model performance levels in dashed lines. Middle: Percent change in perplexity from ideal 10000 photon count performance still exhibits truncated power-law scaling with photons per multiply-accumulate (MAC) operation for all models. Right: Scaling of photon usage for maintaining the 8-bit digital performance versus model size. Dashed lines: constant photons per dot product (optical scaling) and constant photons/MAC analogous to digital scaling. Note that unlike for our results in the main text, smaller models beat the constant-dot-product-total scaling, but the largest model exhibits poor efficiency, as the clipping scheme used here was not well suited for it.

I Effects of Training and Quantization Scheme on Optical Scaling

Our results demonstrating favorable scaling of photon usage in Transformers show that they can be optically efficient, but in general the photon usage is affected by the training scheme and other factors like quantization. This is because approaches for optimization quantization, regularization, etc. affect the statistics of weights and activations in the network, which unlike digital systems, are tied to the resource usage. The main example of this is with weights: they are normalized before being loaded onto an ONN accelerator, and so large outliers may lead to many weights being near 0 after normalization—admitting fewer photons through to the detector. This has a direct impact on the output SNR, and so depending on weight statistics more or fewer photons may be needed in order to run at the same precision.

To discover how a different scheme might affect photon usage, we analyzed the optical scaling of our quantized optical Transformer models with percentile clipping instead of clamping based on EMA statistics. We applied the same clipping to all models (details in Table 4). These clipped models have familiar trends in their language modelling performance versus photon numbers, but we notice key differences in the photons needed to maintain 8-bit digital performance: first, the absolute number of photons needed for the smaller models (120 and 40 versus 340 and 170 of our unclipped scheme for $d = 192$, 384) is much lower—this indicates that clipping of large weight values leads to more transmission after normalization. Second, the scaling is inconsistent, with smaller models needing significantly fewer photons than the expected $1/d$ scaling, but then requiring many photons again for the largest model. The clipping scheme degraded the performance of the large model. Of course, this could be improved by designing a better scheme for the largest model such that it requires few photons; these results illustrate how differences in the training and quantization recipe could lead to a variety of outcomes, and why efficiency is achievable but not an automatic guarantee for any scheme.