Scaling Laws Beyond Backpropagation

Matthew J. Filipovich[1,2] **Alessandro Cappelli**[1] **Daniel Hesslow**[1] **Julien Launay**[1,3]

[1]LightOn [2]Queen's University [3]LPENS, École Normale Supérieure

{firstname}@lighton.ai

Abstract

Alternatives to backpropagation have long been studied to better understand how biological brains may learn. Recently, they have also garnered interest as a way to train neural networks more efficiently. By relaxing constraints inherent to backpropagation (e.g., symmetric feedforward and feedback weights, sequential updates), these methods enable promising prospects, such as local learning. However, the tradeoffs between different methods in terms of final task performance, convergence speed, and ultimately compute and data requirements are rarely outlined. In this work, we use scaling laws to study the ability of Direct Feedback Alignment (DFA) to train causal decoder-only Transformers efficiently. Scaling laws provide an overview of the tradeoffs implied by a modeling decision, up to extrapolating how it might transfer to increasingly large models. We find that DFA fails to offer more efficient scaling than backpropagation: there is never a regime for which the degradation in loss incurred by using DFA is worth the potential reduction in compute budget. Our finding comes at variance with previous beliefs in the alternative training methods community, and highlights the need for holistic empirical approaches to better understand modeling decisions.

1 Introduction

Backpropagation (BP) [1, 2] is just one of many ways to solve the credit assignment problem–and hence to train neural networks. BP estimates the individual contribution of each parameter to the error, but approximate methods can be employed: either through fundamentally different approaches (e.g. Hebbian learning) [3, 4], or by relaxing constraints of BP [5–7]. Beyond backpropagation methods have a history of being studied to understand how biological brains may learn [8, 9]. Indeed, key features of backpropagation are not possible to implement under biological constraints. For instance, using the transpose of the weights $\mathbf{W}^T$ in the feedback path is not possible, as the feedforward and feedback pathways are physically distinct–this is known as the weight transport problem [10].

Once relegated to toy problems [11], alternatives to BP have now been demonstrated to be able to achieve competitive performance on challenging tasks across a variety of architectures [12]. This could result in more efficient training: for instance, local learning may enable easier parallelization of computations at scale [7, 13]. Alternative methods may even be co-designed with hardware [14, 15]: either for novel systems such as photonic co-processors [16] and memristors [17], or to circumvent distributed communication bottlenecks in the large clusters used to train state-of-the-art models [18].

However, the tradeoffs between BP and alternatives are not always clear. If a method enables 25% faster training, is it worth a 5% decrease in end-task performance? Or would a model trained with BP using a 25% smaller compute budget still be better? As most works usually only offer a few cherry-picked datapoints, this is a difficult question to answer. Instead, deriving scaling laws [19] may provide a more complete picture: by obtaining the full power-law relationship between compute spent and task performance, one may easily identify regimes in which the alternative method is competitive with BP, and even extrapolate results to larger scales.

Preprint. Under review.

Contributions. We use scaling laws to study an alternative to BP, with the following contributions:

- Drawing inspiration from work using scaling laws to evaluate modeling decisions [20–22], we are the first to **use scaling laws to study an alternative to backpropagation**.
- At variance with previous beliefs [12], **we find that the gains in compute efficiency from using the alternative method studied are never worth the degradation in performance**. This holds even if we consider the use of exotic hardware, such as optical co-processors [23], which would offload some computations and effectively make the alternative method "free".

2 Framing

Can alternative training methods accelerate neural network training? Surveying the current state-of-the-art, one may find numerous claims of alternative training methods achieving *competitive* performance with BP across a variety of settings and tasks (e.g., [12, 18, 24]).

We seek to study this claim, with three restrictions in scope:

1. We focus on Direct Feedback Alignment [25], due its simplicity and wide applicability [12], as well as its broad hardware prospects [14, 26, 27], and theoretical background [28].
2. We study compute-efficiency specifically (i.e, best performance achievable for a given compute budget), as this usually a significant bottleneck for scaling-up models.
3. We conduct our study on "GPT-like" [29] causal decoder-only Transformers trained on English data. These models are known to possess smooth scaling laws [19, 30]. Because of their unique abilities [31], they also command some of the largest training budgets in machine learning [32], making them a prime target for more compute-efficient training.

These restrictions lead us to test the following hypothesis:

Hypothesis. Direct Feedback Alignment can train causal decoder-only models more efficiently than backpropagation, achieving better performance for a given compute budget.

Scaling laws as a holistic empirical tool. Scaling laws have been proposed as an empirical approach to connect hyperparameters of neural networks (e.g., parameter count, training dataset size) to their performance. They have been derived both on specific downstream tasks [33, 34] and on upstream modeling loss [19]. Scaling laws can characterize the influence of data & modeling decisions [21, 22], or even unveil new, more optimal training practices [35, 36].

As illustrated in Figure 1, it is possible to derive a so-called *compute optimal frontier* for a class of models: this defines $\mathcal{L}(C)$, the best performance $\mathcal{L}$ achievable for a compute budget C. We fit a power-law $\mathcal{L}(C) = (C_c C)^{\alpha_C}$ over the Pareto front of multiple runs, as proposed in [19]. C_c is a constant offsetting the frontier, while α_C controls the slope. Improvements in α_C are rare [20, 22], but valuable as they would point to modeling decisions leading to increased gains at scale.

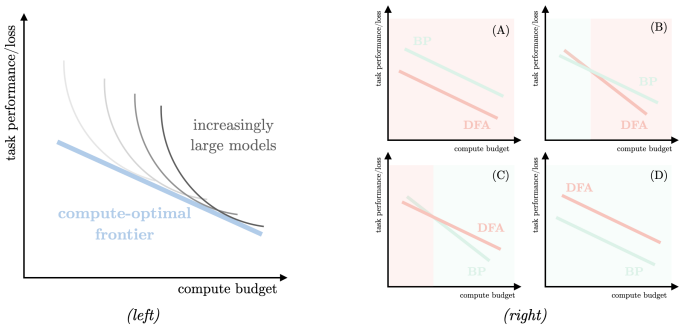

Figure 1: **Scaling laws provide optimal compute frontiers**. Left: compute-optimal frontier. Right: scenarios for DFA (red) & BP (green) scaling laws, shading is best method at a compute budget.

Potential outcomes. We identify four potential outcomes for our study, illustrated in Figure 1:

(A) **DFA always outperform BP.** Thanks to a better offset or improvements in scaling, the increased compute-efficiency from DFA makes it always favorable to BP.

(B) **DFA outperforms BP at scale.** Thanks to improvements in scaling (e.g., increased efficiency compared to BP at scale), DFA eventually makes the training of large models more efficient.

(C) **BP outperforms at scale.** DFA may exhibit poor scaling behavior [11], and may not scale to larger models, leading to BP eventually outperforming DFA.

(D) **BP always outperform DFA.** The degradation in performance observed with DFA may never be worth the potential gains in compute-efficiency.

Both (A) and (B) may be viewed as validating our hypothesis, as they both potentially motivate the use of DFA over BP. (C) and (D) would however be negative outcomes, either restraining the efficient applicability of DFA to small models, or indicating that DFA is never competitive with BP.

3 Methods

Direct Feedback Alignment. Direct Feedback Alignment (DFA) [25] is an extension of Feedback Alignment (FA) [5] which uses a random projection of the global error to directly train each layer.

We introduce at layer i: $\mathbf{W}_i$ the weights, $\mathbf{a}_i$ the pre-activations, f the non-linearity and its derivative f', $\mathbf{h}_i$ the activations, δx the derivative of the loss against x, and $\odot$ the Hadamard product. DFA replaces the backpropagated signal from the $(i+1)$-th layer $\mathbf{W}_{i+1}^T \delta \mathbf{a}_{i+1}$ by a random projection of the global error $\mathbf{Be}$. For most common losses, this error $\mathbf{e}$ is simply the difference between targets and predictions. Accordingly, the update at layer i is now $\delta \mathbf{W}_i = -[(\mathbf{Be}) \odot f'(\mathbf{a}_i)]\mathbf{h}_{i-1}^T$. $\mathbf{B}$, the fixed random Gaussian matrix, can be shared across all layers [37], reducing memory and compute costs significantly–as a single $\mathbf{Be}$ is now calculated and used for all layers . With DFA, the update now does not depend on the backward of other layers; thus, once the forward pass is complete, all of the layers can be updated concurrently, achieving so-called backward-unlocking [7].

Learning with DFA is made possible through a process called alignment. During the training, the forward weights will eventually align with the fixed backward weights, enabling updates which approximate backpropagation [28]. This is best illustrated in the simpler case of FA [5]. For FA, the learning signal still comes from the (i+1)-th layer: $\mathbf{B}_{i+1}\delta \mathbf{a}_{i+1}$. For this to approximate BP, we only need $\mathbf{W}_{i+1}^T \sim \mathbf{B}_{i+1}$. Altough we don't report on it in this work, this is a valuable diagnostic tool when experimenting: at any step, it is possible to measure the angle (cosine similarity) between the gradients predicated by backpropagation and the ones approximated by DFA. Higher alignment values are usually correlated with networks which achieve better end-task performance [12, 37].

Scaling laws for compute-efficiency. We are interested in scaling according to compute budget, $\mathcal{L}(C)$. We split $C = C_{\text{F}} + C_{\text{B}} + C_{\text{U}}$, for computing the forward pass, backpropagation of the error, and weight updates respectively. For causal decoder-only models, each phase costs roughly $2ND$ (in FLOP) with N the number of model parameters and D the dataset size in tokens [19]–the factor 2 coming from the multiply-accumulate. Hence, $C^{\text{BP}} = 6ND$. When using DFA, the backpropagation of the error is not necessary, and instead a single random projection $\mathbf{Be}$ is shared across all layers. Accordingly, $C_{\text{B}}^{\text{DFA}} = 2d_{\text{model}}^2 D$, as $\mathbf{B}$ is of shape $(d_{\text{model}}, d_{\text{model}})$. Because $N \simeq 12 n_{\text{layer}} d_{\text{model}}^2$, $C_{\text{B}}^{\text{DFA}} \ll C_{\text{B}}^{\text{BP}}$. We will neglect it and consider $C^{\text{DFA}} = 4ND$, a $\sim 30\%$ improvement.

Finally, note that this only takes into account improvements in the FLOP compute budget. However, practitioners are usually constrained by the actual compute budget in dollars, which is best represented by the number of GPU-hours required. C_{FLOP} and $C_{\text{GPU-hours}}$ can be linked through the throughput T achieved per GPU, in TFLOPS. Alternative training methods like DFA may improve this throughput, by enabling increased parallelization and reducing communication bottlenecks. Nevertheless, state-of-the-art methods already achieve hardware utilization of $\sim 50\%$ at scale [38]: at best, a 2x improvement can be expected. We thus also introduce $\tilde{C}^{\text{DFA}} = 2ND$, a (very) optimistic estimation which supposes DFA would enable a doubling in effective throughput. In practice, current implementations of DFA are not optimized, and it is unrealistic for DFA to be able to lift all bottlenecks currently encountered in distributed training–we use this estimate as an absolute lower bound of what is possible.

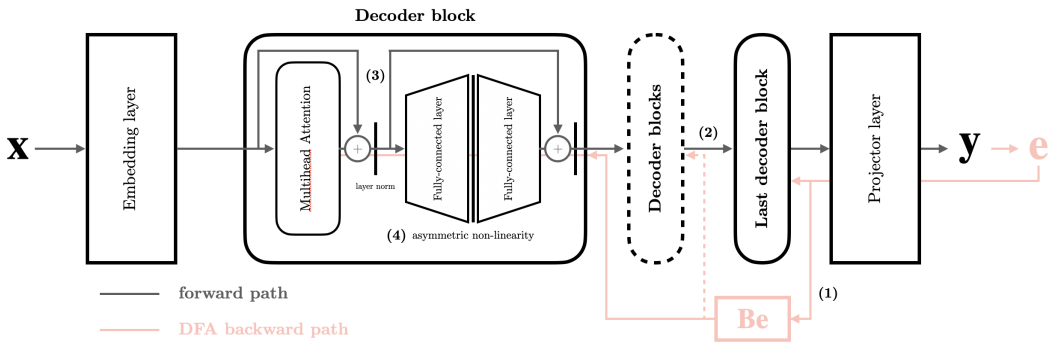

Figure 2: **Our implementation of DFA in decoder-only models**. (1): we take the error before the projector to reduce its dimensionality; (2) we apply DFA per block and backpropagate the DFA signal within a decoder block; (3) residuals are only used during the forward and ignored in the backward; (4) we use ReLU as a non-linearity during the forward, but use the derivative of $\tanh$ in the backward instead. Note that the last decoder block is trained in the same way it would with BP

Training Transformers with DFA. We train decoder-only Transformer models with both DFA and BP on a causal language modeling objective [39, 40], scaling from 60M to 500M parameters. The models are trained on 30B tokens of English CommonCrawl data filtered with CCNet [41], with hyper-parameters adapted from GPT-3 [29], but with a context size of 512 for increased training speed. We chose different optimization hyper-parameters for DFA and BP, performing a sweep for each run and using the best set found for each method. We also train a so-called shallow baseline, as recommended by [37]. This shallow approach only trains the topmost layer, and provides a baseline if DFA was not training deeper layers at all. Since the scaling law for the shallow baseline is easily predictable even at small scale we do not train the largest model in order to save compute.

To improve performance, we diverge from the canonical implementation of DFA (see Figure 2, or Table 2 in appendix for a comparison between all training methods considered):

(1) **Preprojector error.** We take the error $\mathbf{e}$ immediately before the final projector layer instead of after. This reduces the dimension of $\mathbf{e}$ from the size of the vocabulary (51,200) to d_{model} (576-1,408 for our runs). We found that not only does this decrease memory needs and compute costs, but this also results in a small improvement in autoregressive loss.

(2) **Block-wise DFA.** We only apply DFA per decoder-block, and then backpropagate the DFA signal within the block, significantly improving autoregressive loss [12]. This maintains the ability to update blocks independently from one another, simplifying classic parallelization schemes such as pipeline parallelism. However, this makes $C_{\text{B}}^{\text{DFA}}$ non-negligible. In our plots, we still neglect $C_{\text{B}}^{\text{DFA}}$, making our comparisons strongly biased toward DFA. As BP can leverage decades of research and methods finetuned for its idiosyncrasies, we believe this is a fair way to offer a best case scenario for DFA.

(3) **Asymmetric residuals.** We make the residual paths asymmetric: although they remain untouched in the feedforward, they are disabled and ignored in the backward. Within blocks, this prevent the DFA feedback from directly flowing through the attention–[12] found that attention layers often struggled to align when a DFA feedback was applied directly. Letting the DFA feedback first flow through the fully-connected layers improves alignment.

(4) **Asymmetric non-linearities.** Previous work identified that DFA performs best with activation functions which are continuous and bounded [37]; the classic activation functions used in Transformers, ReLU and GeLU, do not fit this criteria. Switching to $\tanh$ reduced the gap between DFA and BP, but also worsened autoregressive loss overall. Instead, we keep ReLU in the forward, but replace its derivative with the derivative of $\tanh$ in the backward. This asymmetric approach improves both alignment and loss.

Finally, note our experiments use 16 to 32 A100 80GB GPUs, using data parallelism only.

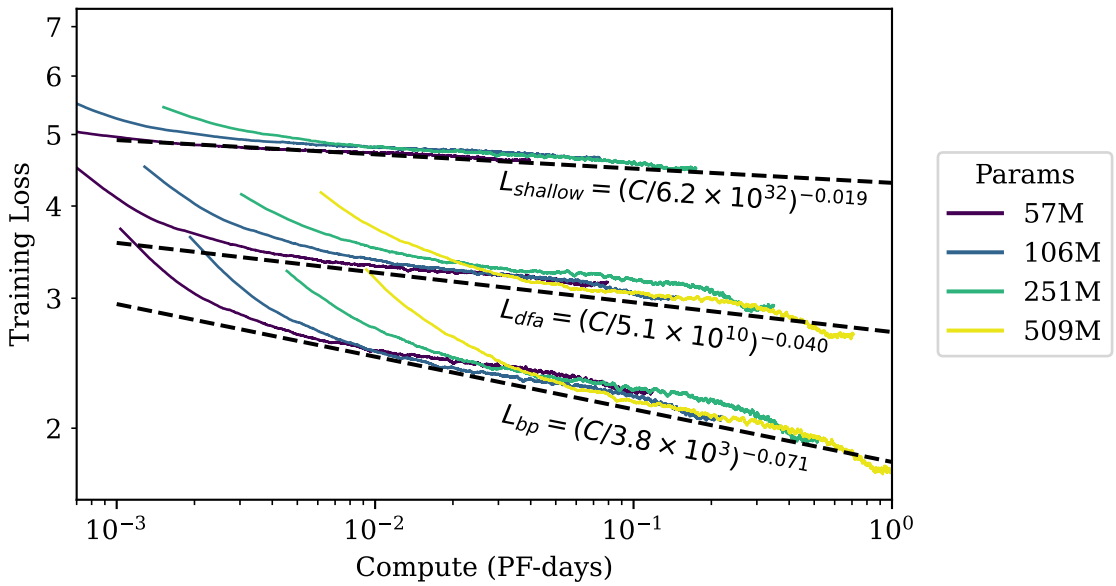

Figure 3: **The degradation in loss incurred by DFA is never worth the improvement in compute.** We plot the training loss against the compute expended–which is $\sim 30\%$ smaller for DFA thanks to improved efficiency. The compute-optimal frontier of DFA is significantly worse than that of BP, and no practical model would benefit from using DFA instead of BP, falsifying our original hypothesis.

4 Experiments

We train models over an order of magnitude in scale: from 57 million to 509 million parameters. These are the largest models ever trained with DFA, representing a significant departure from usual small-scale experiments in the alternative training methods community.

Results. Our results are showcased in Figure 3, with best fit parameters for the compute-optimal frontier in Table 1. We report the training loss directly, as we perform a single epoch over the 30B tokens of data. For the shallow baseline, $C = 2ND$ as we neglect the cost of updating the topmost layer only. Compute budgets are in PF-days, one PFLOPS of compute power sustained for a day.

First, we note that DFA performs significantly above the shallow baseline. This confirms that layers trained by DFA are actually learning meaningful representations, rather than remaining random throughout training. The shallow baseline exhibit little to no scaling, limited to the increase in width of the decoder-block as we increase overall parameter count. Random parameters from additional untrained layers are unlikely to contribute to scaling, as reported by [21].

Importantly, there is never a region for which using DFA is preferrable to using backpropagation from a compute-effiency perspective. DFA not only presents a worse offset than BP, it scales more poorly: the gap with BP widens as compute budget and model scale increases. This is scenario (D): BP always outperform DFA, thus falsifying our original hypothesis.

Table 1: **Best power-law fit of the compute-optimal frontier**. We fit $\mathcal{L}(C) = (C_c C)^{\alpha_C}$ to the curves of Figure 3, lower values of α_C denote better scaling and lower values of C_c an improved offset. Although DFA updates do train layers (the performance observed is better than the shallow baseline), scaling is significantly degraded compared to backpropagation.

	α_C	C_c [PF-days^{-1}]
BP	-0.071	3.8×10^3
DFA	-0.040	5.1×10^{10}
Shallow	-0.019	6.2×10^{32}

5 Discussion

Limitations. As for all negative results, absence of evidence is not evidence of absence. Novel modeling practices may enable DFA to better train decoder-only models, bridging the gap with BP. We did however explore and implement many of such ideas (e.g., preprojector error, asymmetric residuals/non-linearities), failing to bridge said gap on our own. Our study is also focused on the specific setting of causal decoder-only models, for which scaling laws are easy to derive. Such large language models are well-known to be difficult to train, even with BP [42]. DFA may find more success with simpler architectures. Furthermore, our study did not consider other alternatives to backpropagation: like for DFA, promising results exist also for greedy layerwise methods [24] or local gradients approaches [7]. Finally, we did not conduct a full exploration of learning rate schedules and architectural hyperparameters, as performed in recent scaling laws studies [35] to find the compute-optimal frontier. However, given the large gap between BP and DFA, it is unlikely such an exploration would result in a significantly different finding.

Conclusion. Alternative training methods exhibit compelling properties: they may enable local learning [13], leading to easier and better parallelization of distributed computations [18], and may even open new avenues for exotic hardware [14, 43]. However, care must be taken to make fair comparisons. Notably, reporting individual datapoints can be misleading, and more holistic approaches, such as scaling laws, should be favored to paint the full picture of tradeoffs.

In this work, we used scaling laws to characterize the compute-efficient frontier of DFA when training causal decoder-only models, and compared with backpropagation. We originally hypothesized, following previous literature, that the increase in compute efficiency from DFA may lead to a favorable compromise against backpropagation. However, our experiments falsified this hypothesis:

> **Finding.** At variance with previous beliefs, using Direct Feedback Alignment to train causal decoder-only models is never more compute-efficient than using backpropagation.

This finding holds despite assumptions heavily in favor of DFA: even if we were to assume a compute budget of $\tilde{C}^{\text{DFA}} = 2ND$ (removing entirely the cost of the backward and updates, for instance by offloading them to a coprocessor [6, 16]), DFA is never more compute-efficient than backpropagation.

Although this is a significantly negative finding, we would like to end on two outlooks:

- **Alternative methods beyond compute-efficiency.** Our work studied DFA under the practical light of compute-efficiency, but alternative training methods can also be motivated by more than simply reducing compute budgets. For instance, they have been invaluable in providing models for how the brain may learn [9]. Specifically in the case of DFA, they can also be leveraged for increased adversarial robustness [44–46], or for novel implementations of differential privacy [47–49]. These prospects naturally encourage further work in this direction, beyond simple compute-efficiency.
- **Scaling laws as a key modeling tool.** The pitfalls outlined by our study also apply to other modeling works. In the specific case of Transformer models, identifying best practices has been the subject of much contention in the literature [50, 51]. Scaling laws have been invaluable in going beyond cherry-picked datapoints: for neural machine translation [20], mixture-of-experts models [52], or even broad architectural choices [53].

References

[1] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing*, volume 1, pages 318–362. MIT Press, 1986.

[3] Javier R Movellan. Contrastive hebbian learning in the continuous hopfield model. In *Connectionist models*, pages 10–17. Elsevier, 1991.

[4] Randall C O'Reilly. Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural computation*, 8(5):895–938, 1996.

[5] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):1–10, 2016.

[6] Charlotte Frenkel, Martin Lefebvre, and David Bol. Learning without feedback: Fixed random learning signals allow for feedforward training of deep neural networks. *Frontiers in neuroscience*, 15:629892, 2021.

[7] Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1627–1635, 2017.

[8] Francis Crick. The recent excitement about neural networks. *Nature*, 337(6203):129–132, 1989.

[9] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, pages 1–12, 2020.

[10] Stephen Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1):23–63, 1987.

[11] Sergey Bartunov, Adam Santoro, Blake Richards, Luke Marris, Geoffrey E Hinton, and Timothy Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. In *Advances in Neural Information Processing Systems*, pages 9368–9378, 2018.

[12] Julien Launay, Iacopo Poli, François Boniface, and Florent Krzakala. Direct feedback alignment scales to modern deep learning tasks and architectures. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9346–9360. Curran Associates, Inc., 2020.

[13] Arild Nøkland and Lars Hiller Eidnes. Training neural networks with local error signals. In *International Conference on Machine Learning*, pages 4839–4850, 2019.

[14] Julien Launay, Iacopo Poli, Kilian Müller, Gustave Pariente, Igor Carron, Laurent Daudet, Florent Krzakala, and Sylvain Gigan. Hardware beyond backpropagation: a photonic co-processor for direct feedback alignment. *NeurIPS Beyond Backpropagation Workshop*, 2020.

[15] Charlotte Frenkel. Bottom-up and top-down neuromorphic processor design: Unveiling roads to embedded cognition. 2020.

[16] Julien Launay, Iacopo Poli, Kilian Müller, Igor Carron, Laurent Daudet, Florent Krzakala, and Sylvain Gigan. Light-in-the-loop: using a photonics co-processor for scalable training of neural networks. In *2020 IEEE Hot Chips 32 Symposium (HCS)*. IEEE, 2020.

[17] Maxence Ernoult, Julie Grollier, and Damien Querlioz. Using memristors for robust local learning of hardware restricted boltzmann machines. *Scientific reports*, 9(1):1–15, 2019.

[18] Michael Laskin, Luke Metz, Seth Nabarro, Mark Saroufim, Badreddine Noune, Carlo Luschi, Jascha Sohl-Dickstein, and Pieter Abbeel. Parallel training of deep networks with local updates. *arXiv preprint arXiv:2012.03837*, 2020.

[19] 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. *arXiv preprint arXiv:2001.08361*, 2020.

[20] Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation. In *International Conference on Learning Representations*, 2021.

[21] Amélie Chatelain, Amine Djeghri, Daniel Hesslow, and Julien Launay. Is the number of trainable parameters all that actually matters? In *I (Still) Can't Believe It's Not Better! Workshop at NeurIPS 2021*, pages 27–32. PMLR, 2022.

[22] Yamini Bansal, Behrooz Ghorbani, Ankush Garg, Biao Zhang, Colin Cherry, Behnam Neyshabur, and Orhan Firat. Data scaling laws in nmt: The effect of noise and architecture. In *International Conference on Machine Learning*, pages 1466–1482. PMLR, 2022.

[23] Charles BROSSOLLET, Alessandro Cappelli, Igor Carron, Charidimos Chaintoutis, Amélie Chatelain, Laurent Daudet, Sylvain Gigan, Daniel Hesslow, Florent Krzakala, Julien Launay, et al. Lighton optical processing unit: Scaling-up ai and hpc with a non von neumann co-processor. In *2021 IEEE Hot Chips 33 Symposium (HCS)*, pages 1–11. IEEE, 2021.

[24] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. In *International Conference on Machine Learning*, pages 583–593, 2019.

[25] Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. In *Advances in neural information processing systems*, pages 1037–1045, 2016.

[26] Charlotte Frenkel, Martin Lefebvre, Jean-Didier Legat, and David Bol. A 0.086-mm^2 12.7-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos. *IEEE transactions on biomedical circuits and systems*, 13(1):145–158, 2018.

[27] Matthew J Filipovich, Zhimu Guo, Mohammed Al-Qadasi, Bicky A Marquez, Hugh D Morison, Volker J Sorger, Paul R Prucnal, Sudip Shekhar, and Bhavin J Shastri. Monolithic silicon photonic architecture for training deep neural networks with direct feedback alignment. *arXiv preprint arXiv:2111.06862*, 2021.

[28] Maria Refinetti, Stéphane d'Ascoli, Ruben Ohana, and Sebastian Goldt. Align, then memorise: the dynamics of learning with feedback alignment. In *International Conference on Machine Learning*, pages 8925–8935. PMLR, 2021.

[29] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[30] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

[31] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective work best for zero-shot generalization? In *ICML*, 2022.

[32] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[33] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

[34] Ibrahim Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. *arXiv preprint arXiv:2209.06640*, 2022.

[35] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[36] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *arXiv preprint arXiv:2206.14486*, 2022.

[37] Julien Launay, Iacopo Poli, and Florent Krzakala. Principled training of neural networks with direct feedback alignment. *arXiv preprint arXiv:1906.04554*, 2019.

[38] Vijay Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *arXiv preprint arXiv:2205.05198*, 2022.

[39] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[40] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

[41] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, 2020.

[42] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[43] George Dabos, Dimitris V Bellas, Ripalta Stabile, Miltiadis Moralis-Pegios, George Giamougiannis, Apostolos Tsakyridis, Angelina Totovic, Elefterios Lidorikis, and Nikos Pleros. Neuromorphic photonic technologies and architectures: scaling opportunities and performance frontiers. *Optical Materials Express*, 12(6):2343–2367, 2022.

[44] Alessandro Cappelli, Julien Launay, Laurent Meunier, Ruben Ohana, and Iacopo Poli. Ropust: Improving robustness through fine-tuning with photonic processors and synthetic gradients. *arXiv preprint arXiv:2108.04217*, 2021.

[45] Albert Jiménez Sanfiz and Mohamed Akrout. Benchmarking the accuracy and robustness of feedback alignment algorithms. *arXiv preprint arXiv:2108.13446*, 2021.

[46] Alessandro Cappelli, Ruben Ohana, Julien Launay, Laurent Meunier, Iacopo Poli, and Florent Krzakala. Adversarial robustness by design through analog computing and synthetic gradients. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3493–3497. IEEE, 2022.

[47] Ruben Ohana, Hamlet Medina, Julien Launay, Alessandro Cappelli, Iacopo Poli, Liva Ralaivola, and Alain Rakotomamonjy. Photonic differential privacy with direct feedback alignment. *Advances in Neural Information Processing Systems*, 34:22010–22020, 2021.

[48] Jaewoo Lee and Daniel Kifer. Differentially private deep learning with direct feedback alignment. *arXiv preprint arXiv:2010.03701*, 2020.

[49] Arash Asadian, Evan Weidner, and Lei Jiang. Self-supervised pretraining for differentially private learning. *arXiv preprint arXiv:2206.07125*, 2022.

[50] Sharan Narang, Hyung Won Chung, Yi Tay, Liam Fedus, Thibault Févry, Michael Matena, Karishma Malkan, Noah Fiedel, Noam Shazeer, Zhenzhong Lan, et al. Do transformer modifications transfer across implementations and applications? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5758–5773, 2021.

[51] Teven Le Scao, Thomas Wang, Daniel Hesslow, Lucile Saulnier, Stas Bekman, M Saiful Bari, Stella Biderman, Hady Elsahar, Jason Phang, Ofir Press, et al. What language model to train if you have one million gpu hours? In *Challenges & Perspectives in Creating Large Language Models, BigScience ACL Workshop*, 2022.

[52] Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International Conference on Machine Learning*, pages 4057–4086. PMLR, 2022.

[53] Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Won Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Q Tran, Dani Yogatama, and Donald Metzler. Scaling laws vs model architectures: How does inductive bias influence scaling? *arXiv preprint arXiv:2207.10551*, 2022.

A Overview of differences between training methods

Table 2: **Overview of the training methods considered.**. We make a number of changes to the canonical implementation of DFA to enhance its ability to train Transformer models.

	BP	**DFA**		**Shallow**
		Canonical	Ours	
Training cost	$6ND$	$4ND$	$\sim 6ND$	$\sim 2ND$
Update rule	$-[(\mathbf{W}_{i+1}^T \delta \mathbf{a}_{i+1}) \odot f'(\mathbf{a}_i)]\mathbf{h}_{i-1}^T$	$-[(\mathbf{Be}) \odot f'(\mathbf{a}_i)]\mathbf{h}_{i-1}^T$		no update
Error	N/A	Directly from loss	From gradient of loss before projector	N/A
Strategy	Full backpropagation	DFA feedback at every layer, no backpropagation	Block-wise: DFA feedback at the top of decoder blocks, backpropagation within the blocks	All decoder blocks but the last one frozen
Last block		Trained with backpropagation		
Residuals	Vanilla	Vanilla	Asymmetric	Vanilla
Non-linearity	Vanilla	Vanilla	Asymmetric	Vanilla