Perturbative Gradient Training: A novel training paradigm for bridging the gap between deep neural networks and physical reservoir computing

Cliff B. Abbott, Mark Elo, and Dmytro A. Bozhko

***Abstract*—We introduce Perturbative Gradient Training (PGT), a novel training paradigm that overcomes a critical limitation of physical reservoir computing: the inability to perform backpropagation due to the black-box nature of physical reservoirs. Drawing inspiration from perturbation theory in physics, PGT uses random perturbations in the network's parameter space to approximate gradient updates using only forward passes. We demonstrate the feasibility of this approach on both simulated neural network architectures, including a dense network and a transformer model with a reservoir layer, and on experimental hardware using a magnonic auto-oscillation ring as the physical reservoir. Our results show that PGT can achieve performance comparable to that of standard backpropagation methods in cases where backpropagation is impractical or impossible. PGT represents a promising step toward integrating physical reservoirs into deeper neural network architectures and achieving significant energy efficiency gains in AI training.**

***Index Terms*—Reservoir computing, spin waves, neural networks, energy-efficient AI, perturbation methods, magnonics.**

I. Introduction

Machine Learning has seen explosive growth in recent years. With that growth comes an immense increase in the power consumed when training and operating AI/ML models. Training the ML model Meena had a carbon footprint roughly equal to driving 242,231 miles in an average passenger vehicle [1] and ChatGPT-4 cost more than $100 million to train [2] with the majority of the cost assumed to be energy usage. Some estimates even put the annual deployment energy consumption for popular models at 25x the energy cost of training [3], [4], which would put an energy price tag on ChatGPT-4 of around $2.5 billion. On the lower end, estimates still indicate that the energy cost of training makes up only 10-40% of the lifetime cost [1], [5]. The growth in the energy demands for AI has been exponential [5] and with current demands for a single model like ChatGPT being equivalent to that of 33,000 households in the US [6], more energy efficient forms of machine learning will be necessary to continue development in the field.

Reservoir computing (RC) is a promising concept for reducing the energy costs of machine learning. An RC takes a lower-dimensional input and maps it to a higher-dimensional output via a complex, nonlinear, but dynamically consistent process. Unlike other machine learning architectures, the training process is regulated to the output or readout layer(s) of the RC, allowing significant reductions in training time [7]. The idea of RCs has been previously applied to the transformer architecture, which currently dominates the field of Large Language Models, and showed that the introduction of "frozen layers" as reservoirs had a significant improvement in performance and training time [8]. Physical reservoir computing takes this concept and uses other natural systems, such as water waves [9], to perform the non-linear process. Magnonics systems are one such promising example of a physical reservoir computer with significantly lower energy costs, with potential energy savings of up to 90% compared to traditional electronic systems [10]–[20]. The ability to replace conventional electronic systems with magnonic reservoirs could potentially save $100's of millions of dollars in lifetime energy costs.

Despite their promise, physical RCs face a critical limitation: the reservoir acts as a black box, making internal parameter access and backpropagation impossible. This severely limits the applicability of physical RCs to only the input layer of a network. As cutting-edge models are getting deeper and more complex, this means the impact of a physical RC to reduce energy costs is diminishing. To address this, we propose Perturbative Gradient Training (PGT), a novel method that eliminates the need for backpropagation, enabling physical reservoir computers to be seamlessly integrated into neural networks, regardless of depth and location of the RC.

This paper is structured as follows: Section II introduces the theoretical framework and applications of PGT. Section III presents results, comparing PGT to standard backpropagation in a simulated reservoir system. Section IV applies PGT to an actual physical reservoir system and examines its performance in experimental settings. Finally, Section V discusses the results and highlights the challenges and opportunities for further development of Perturbative Gradient Training.

II. Theory

The current state-of-the-art training methods, while differing in how they update model parameters, all rely on backpropagation. This is where the model is run in a forward pass on a set of training data and a loss is calculated as a comparison of the model output to the target data.

$$L = f(y, y'; \theta) \tag{1}$$

Here $f()$ is the chosen function, such as a mean squared difference, y and y' are the model output and target respectively,

Cliff B. Abbott and Dmytro A. Bozhko are with the Department of Physics and Energy Science, University of Colorado Colorado Springs, Colorado Springs, CO 80918, USA

Mark Elo is with Tabor Electronics, Inc., Hatasia 9, Nesher 3660301, Israel

and θ is the set of model parameters. The loss is then passed backwards through the model, layer by layer, calculating the gradient of the loss for each individual parameter in the network and then applying the chosen update function (standard Stochastic Gradient Descent given here):

$$\theta_{Grad} = \frac{\partial L}{\partial \theta} \tag{2}$$

$$\theta_n = \theta_{n-1} - l_r \cdot \theta_{Grad} \tag{3}$$

Where l_r is the learning rate, a hyperparameter set by the user. Visually, this method searches the n-D parameter space, looking along a single dimension at a time, and determines whether or not the loss improves with an advance in that direction. This is already a significant approximation for the ideal path to a minimum [21], [22]. For instance, in a three-dimensional landscape, the local gradient might indicate that moving north and east individually reduces the loss, while a combined north-east direction could lead uphill. The update to the model would then have a negative impact. However, the likelihood that such a negative update would happen repeatedly is low and this method offers a favorable trade in simplicity over precision. This simplicity allows computers to perform these steps thousands of times, which has historically led to good results.

The concept for Perturbative Gradient Training comes from a technique often used in physics called Perturbation Theory. Perturbation Theory approximates solutions to complex problems by starting with a simpler, exactly solvable version of the problem and iteratively introducing small corrections. This approach allows for incremental refinement, enabling solutions to problems that would otherwise be too difficult to solve directly. Here we start with our simple solution as the initial pre-generated parameters. A small change to some of the parameters is made and we analyze how that change effected the loss. If the change was favorable, we keep a portion of the change determined by the optimization step. If the change was unfavorable, we subtract a portion of that change from the original parameters during the optimization step.

In comparison to traditional backpropagation, which looks at each dimension individually in parameter space, PGT picks a single random direction in the overall space and evaluates the gradient of the loss function along that random direction. This is done by taking loss measurements by a full forward pass of the model just as with equation (1). However, this time, instead of calculating individual gradients for each dimension, we generate a random perturbation matrix which is a set of integers $[-r, -r+1, ..., 0, ..., r-1, r]$ that is the same size as the parameter space. This matrix represents the direction in parameter space that we are looking. Here r is the "range" of the perturbations and is a hyper-parameter set by the user. Increasing the range increases the number of possible directions that PGT can look in parameter space (ex. Fig 1).

It is not always beneficial to include all the parameters all the time. Thus at this step, we introduce the next hyperparameter, the dropout scale. The dropout scale is a probability factor used to determine whether a specific parameter perturbation is kept or set to zero. When dropout is set to 0, all perturbations are kept. At 1, all perturbations are removed.

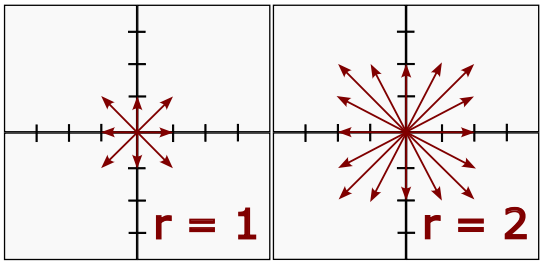

Fig. 1. In a 2-D space, $r = 1$ corresponds to 8 directions and $r = 2$ corresponds to 16 possible directions, demonstrating how changing the range effects the possible directions that PGT can search during training.

Next, we add (subtract) the perturbations to the parameters and determine the new loss.

$$\theta_{p+} = \theta + [PM] \cdot \delta \tag{4}$$

$$\theta_{p-} = \theta - [PM] \cdot \delta \tag{5}$$

Where [PM] is the perturbation matrix and δ is the scaling factor. A new loss is then calculated and an overall gradient for that perturbation (i.e. direction in parameter space) is calculated.

$$L_{p+} = f(y, y'; \theta_{p+}) \tag{6}$$

$$L_{p-} = f(y, y'; \theta_{p-}) \tag{7}$$

$$Grad = \frac{L_{p+} - L_{p-}}{2 \cdot \delta} \tag{8}$$

A "gradient matrix" is then formed by multiplying the gradient by the perturbation matrix.

$$[Update] = \frac{Grad \cdot [PM]}{[Counts]} \tag{9}$$

where [Counts] is a matrix keeping track of the absolute value of the perturbations for scaling. The update can then be applied using any of the standard optimization step techniques. This is similar to proposed methods for training Quantum Neural Networks through parameter shifting [23]. However, unlike PGT, parameter shifting requires a baseline forward pass and an additional forward pass for every parameter in the network, resulting in potentially millions of forward passes per training sample. With PGT, there are only two forward passes per sample. During the training cycle, we found that generating a new perturbation matrix for every sample resulted in the best performance.

In this fashion, backpropagation has been replaced by a second forward pass. Because this method only requires two forward passes, a physical reservoir deployed at any position within the network will not prevent training of the parameters before it.

III. Simulated Results

For initial testing, we created a small dense network that fed into a simulated physical reservoir that was a second complex network whose parameters were not updated (Fig 2) but gradients could be passed back through the reservoir. PGT is not intended to be used in a system where backpropagation is possible. However, this comparison is helpful in understanding

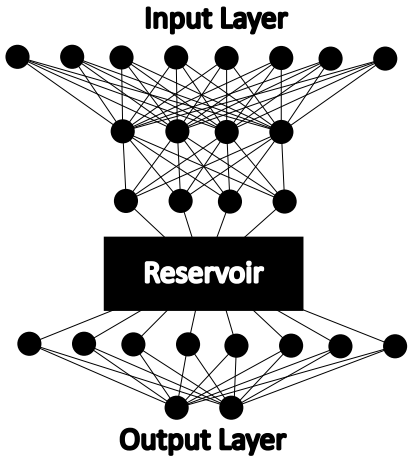

Fig. 2. Initial simple network design. Actual sizes: Input 30, 1st Hidden 200, 2nd Hidden 200, Reservoir In 5, Reservoir Out 100, Output 2.

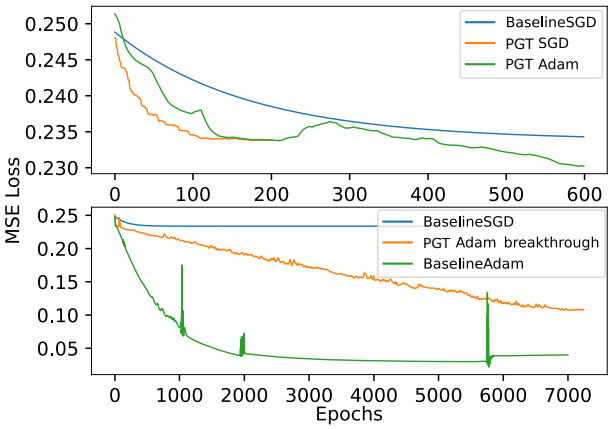

Fig. 3. Initial PGT training performance with Stochastic Gradient Descent (SGD) or Adam Optimization compared to backpropagation "Baseline" training. Top: PGT on the small network outperformed standard SGD significantly both in speed and final loss. Bottom: Adam backpropagation significantly outperformed SGD and PGT. Occasionally PGT training with Adam updates would break through the SGD limit and steadily improve towards the best loss achieved by Adam backpropagation.

the trade-off between the benefits of introducing a physical reservoir into the network and the current limitations of this training method. The simulated reservoir consisted of 4 layers of size 100 or 200. The forward pass of the reservoir passed through the layers multiple times creating recurrent loops.

For this training, we used the Wisconsin Breast Cancer Database [24], which consists of 30 input features resulting in 2 classifications, malignant or benign. A simple Mean Squared Error Loss was used.

$$MSELoss = \sum_{n}(y_n - y'_n)^2 \tag{10}$$

This dataset and network design resulted in an interesting outcome in that Stochastic Gradient Descent [25] (SGD) backpropagation always stopped at a MSE Loss around 0.2338 whereas Adam [26] backpropagation was regularly able to achieve a minimum loss of 0.0221. When PGT was applied to the problem, both PGT with SGD and Adam optimization achieved results faster than the SGD baseline, but also got

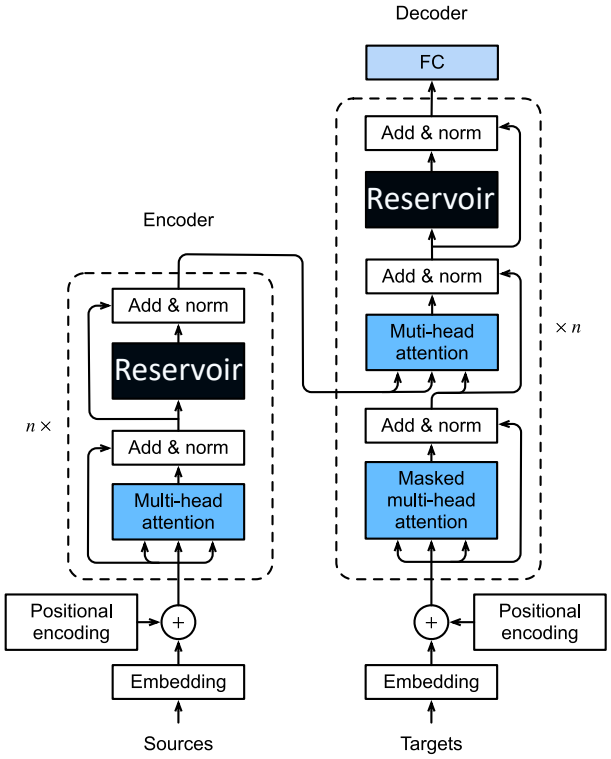

Fig. 4. Transformer Architecture with Feed Forward networks replaced by Reservoir Computers.

stuck at the SGD limit (Fig 3 top). Occasionally, the PGT with Adam optimization would break through that limit and continue steadily towards the min loss achieved by Adam backpropagation. However, that improvement was very slow comparatively (Fig 3 bottom). Using a low dropout rate, we were able to reach the SGD limit more quickly, but only a high dropout rate (0.999) tended to break through the SGD limit.

Next, we looked at applying PGT to the transformer architecture as suggested by Shen et al. [8] where the feed-forward layers of the transformer are treated as reservoirs (Fig 4). We used a single-layer encoder and decoder with an embedding size of 512. The same simulated reservoir was used as in the previous section. For this training, we used a small segment of the Multi30k English to German translation database [27] (250 samples to train and 250 samples to test). The outputs were left as embeddings and a MSE Loss was used again. With the transformer, both SGD and Adam backpropagation regularly outperformed PGT. Only a very high dropout rate of 0.9999, which corresponded to about 330 parameters per perturbation out of 3,363,652 total parameters, was able to consistently train for PGT. Still, PGT achieved a minimum loss of 1.097 after 745 epochs, taking 2.86x longer to reach maximum performance, while Adam achieved a minimum loss of 0.86 after about 260 epochs (Fig 5).

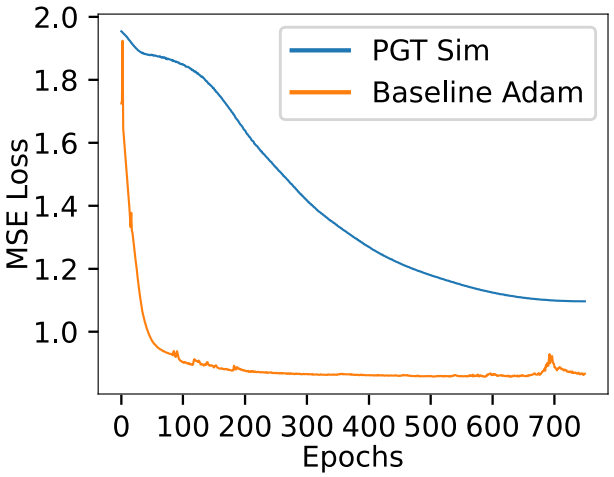

Fig. 5. Performance of PGT on a simulated reservoir transformer. While backpropagation improved significantly faster, PGT was able to get a reasonable result taking only 2.86x longer.

IV. Experimental Results

A. Physical Reservoir Dynamics

To test our approach experimentally, we used a Magnonic Auto-Oscillation Ring as the physical reservoir computer. A basic diagram of the setup is provided in Fig. 6. A thin $10.5\,\mu$m-thick film of Yttrium Iron Garnet (YIG, $Y_3Fe_5O_{12}$) is placed into a 500 G external magnetic field applied along the direction of intended wave propagation realizing the so-called Backward Volume Magnetostatic Spin Wave (BVMSW) geometry.

The orientation of the external magnetic field allows for the excitation of different spin wave geometries and is one of the characteristics that makes magnonics interesting for reservoir computing. A field applied out of plane of the waveguide excites Forward Volume Magnetostatic Spin Waves (FVMSW) and a field in plane but perpendicular excites a Surface Spin Wave or Damon-Eshbach mode (DE). Each wave type has characteristics that may be of more interest depending on the application. For example, the propagation direction: DEs propagate in one principal axis (perpendicular to the in-plane field), but in a nonreciprocal manner (the waves localize differently on opposite surfaces for $+k$ vs. $-k$). BVMSWs propagate along the axis parallel to the magnetization (i.e., one in-plane axis) in both the $+k$ and $-k$ directions. FVMSWs propagate in any direction along the 2D plane of the waveguide. Combinations of these waves can be achieved by orienting the external field in between these primary directions.

A Tabor Proteus Arbitrary Waveform Transceiver (AWT) delivers an electric signal to a thin strip of wire laid perpendicularly across the YIG waveguide, serving as the input antenna. According to Ampère's Law,

$$\oint B \cdot dl = \mu_O I_{enc} \tag{11}$$

the current in this antenna induces an oscillating magnetic field. Here B is the Magnetic field along the closed loop path defined by dl, I_{enc} is the current in the antenna, and μ_O is the permeability of free space. This oscillating magnetic field excites a spin wave in the magnetic spins of the YIG film. The dynamics of this wave is governed by the Landau Lifshitz Gilbert (LLG) equation.

$$\frac{d\mathbf{M}}{dt} = -\gamma \mathbf{M} \times \mathbf{H}_{\text{eff}} + \frac{\alpha}{M_s} \mathbf{M} \times \frac{d\mathbf{M}}{dt} \tag{12}$$

Here M is the magnetization vector (magnetic moment per unit volume) of the material; $\mathbf{H}_{\text{eff}}$ is the effective magnetic field, which includes external fields, anisotropy fields, exchange fields, and demagnetization fields; γ is the gyromagnetic ratio, which relates the magnetic moment to its angular momentum; α is the Gilbert damping parameter, a dimensionless constant that governs how quickly the magnetization relaxes to equilibrium; and M_s is the saturation magnetization, the maximum magnetization the material can achieve. As the wave travels along the waveguide past the output antenna, the wave creates a current in it in the reverse process as the input antenna. That signal then travels to an amplifier that increases the signal by a constant amount. The signal is split back to the AWT for readout and back to the input signal of the input antenna. In this fashion, a pulse from the AWT will travel around the loop several times (determined by the amount of gain given by the amplifier and attenuator combination on the main ring loop). Interactions between the looping waves and newly generated waves introduce non-linear dynamics. This setup gets it's name from the fact that, if the amplification is sufficient, noise in the system will begin to propagate as a signal at the resonant frequency for the waveguide. The amplification gain at which this occurs is called the "auto-oscillation threshold". For the purposes of acting as a RC, the ring is operated just below this threshold. This is done so that previous inputs to the system circulate the loop several times in a decaying fashion, which gives the system it's fading memory, an important characteristic for RCs. The amount of effective memory can be tuned in the system by changing the amount of gain from the amplifier, which determines how many times a pulse in the system will loop before it decays completely.

The dynamics of the wave propagation, governed by the LLG equation, is the source of the non-linear dynamics required for the Auto-Oscillation Ring to function as the RC. $\mathbf{H}_{\text{eff}}$ is dependent on the wave amplitude leading to a nonlinear frequency shift and other nonlinear spin-wave phenomena providing a rich set of complex spin-wave interactions. For more detail into the physics of magnonic auto-oscillation rings and the state of that research, the reader is referred to the reference works by Watt et al. [28], [29] and Ustinov et al. [30].

B. Reservoir Characterization

A measure of short-term memory (STM) and parity check (PC) are typically used to characterize the ability of auto-oscillation rings to act as physical reservoir computers [31], [32]. STM is done by giving the RC a series of 1's and 0's in random ordering. The output of a trained RC is then analyzed to see how many inputs the system can determine were 1 or 0 looking back from the last input to the previous. For example, a STM of 7 would mean that the RC could reliably tell you the last 7 inputs into the system. This metric quantifies the

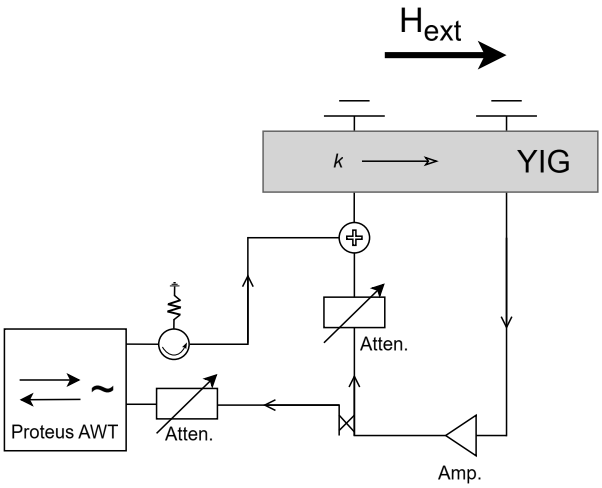

Fig. 6. Basic design of the Magnonic Auto Oscillation Ring. A signal is generated at the AWT Out Channel that travels to the input antenna (gold) of the YIG waveguide. A spin wave is excited in the waveguide and travels to the output antenna. The signal generated at the output antenna travels to an amplifier that amplifies the signal by a set amount. The signal is then split to the In Channel of the AWT and back to the waveguide input antenna via a variable signal attenuator.

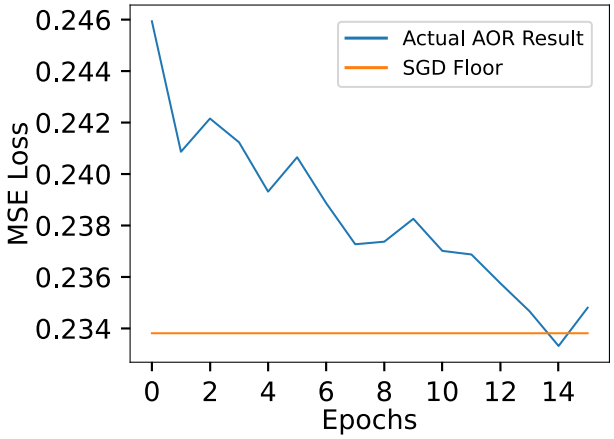

Fig. 7. Results for training the initial neural network design with a physical auto-oscillation ring as the reservoir.

"memory depth" of the system. PC is very similar, but looks at the modulus 2 (XOR) summation of the previous inputs. This means the RC must keep track of whether the number of 1's in the series of inputs was even or odd. A high PC score indicates the system's capacity to perform nonlinear transformations. Typically, scores for STM and PC are given as a Capacity defined as

$$C = \sum_{T_{\text{delay}}=1}^{T_{\text{delay, max}}} \text{Cor}(T_{\text{delay}})^2 \tag{13}$$

where

$$\text{Cor}(T_{\text{delay}})^2 = \frac{(\text{Cov}[y_{\text{train}}(T, T_{\text{delay}}), y_{\text{out}}(T)])^2}{\text{Var}[y_{\text{train}}(T, T_{\text{delay}})] \cdot \text{Var}[y_{\text{out}}(T)]} \tag{14}$$

and T_{delay} is how far back you are looking, COV is the covariance, and Var is the variance. This can range up to 10 and 4 for C_{STM} and C_{PC} respectively in numerical simulations [28], although there is usually a trade off between maximizing one or the other. The setup for our auto-oscillation ring achieved $C_{STM} = 2.91$ and $C_{PC} = 0.01$. For reference, Ustinov et al. [30] had $C_{STM} = 5.1$ and $C_{PC} = 0.1$ for one of their trials. Therefore, while it is not state-of-the-art for the field or indicative of the potential for Magnonics Reservoir Computing, our setup was sufficient to conceptually test PGT.

C. PGT Results

Initially we started with the small neural network described by Fig 2 and replaced the simulated reservoir with the magnonic auto-oscillation ring described above. This performed extremely well achieving the MSE Loss minimum of SGD backpropagation in only 14 training epochs (Fig 7). However, it was noticed during debugging that there was no change in the performance when training only the portion of the network before the reservoir. Therefore, all the training was occurring in the readout layer post reservoir. Although still technically successful in the ability of PGT to train the system, it did not prove the ability of using reservoirs deeper in neural networks and training the network in the pre-reservoir portion in a meaningful way. This is why we shifted to the transformer approach.

As with the initial neural network, we took the reservoir transformer architecture in Fig 4 and replaced the simulated reservoir with our auto-oscillation ring. Due to the design and implementation of the ring, each token for each sample had to be sent through the reservoir one at a time which significantly slowed down the training speed of the system. Even with only 250 training samples and 250 testing samples, out of the 30,000 available, it took around 24 minutes per epoch. It should be noted that this is a limitation in our realization of the RC with the given hardware and is not a limitation inherent to physical reservoirs in general, nor a limit of PGT.

The physical reservoir transformer performed as expected from the simulated results (Fig 8). Compared to the simulation, we see that the initial loss was higher (simply a result of the parameter initialization) but improves at a consistent rate. Fig. 8b shows that difference between the simulated and experimental loss is also decreasing with a negative second derivative. This indicates that the experimental loss appears to be converging slightly faster than simulation, which was also observed in the small initial neural network.

V. Discussion

In this work, we introduced the framework for Perturbative Gradient Training (PGT) and demonstrated its potential to address a major limitation in physical reservoir computing. By enabling training without requiring backpropagation, PGT offers a viable solution for systems where traditional gradient-based methods are impractical or impossible. Our results show that PGT achieves comparable performance to standard backpropagation methods in scenarios where backpropagation cannot be applied, underscoring its value as a training method uniquely suited for such cases.

We estimated that training our reservoir transformer using PGT took approximately 2.8x more training epochs than

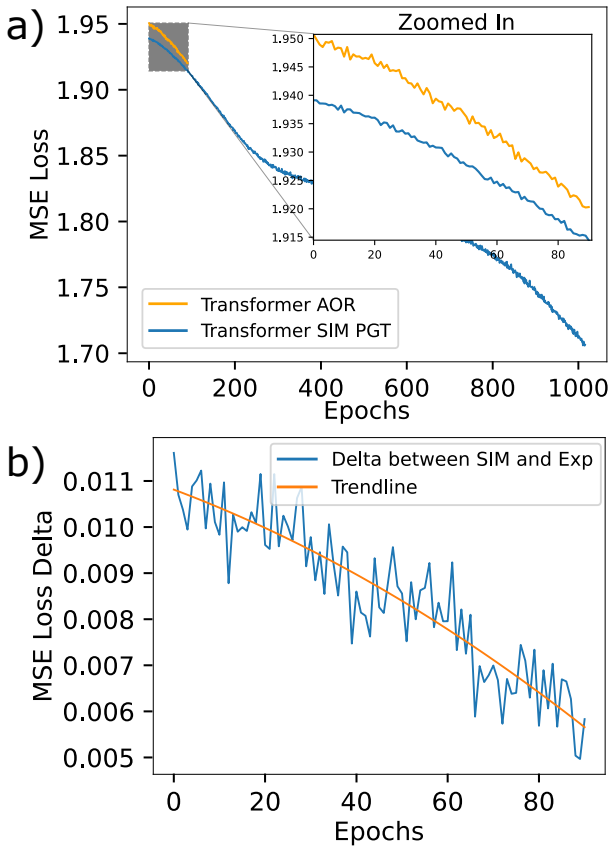

Fig. 8. Performance of the physical implementation of the reservoir transformer vs the simulation. a) shows that the physical reservoir is training in a similar fashion to the simulation. b) the simulation started at a lower loss but the delta is closing showing that the physical implementation is converging faster than the simulation.

a comparable traditional transformer. However, the energy efficiency gains achievable with physical reservoirs present a compelling case for adoption. On the higher end of estimates, substituting physical reservoirs into a transformer requires only a 7.5% improvement in overall energy efficiency to break even after the first year of model deployment. On the lower end, energy reductions of 13-35% would be necessary.

Despite its promise, several limitations remain. One notable challenge is the inability to batch training samples, which leads to significantly longer training times per epoch compared to GPU-based systems. Addressing this limitation will require advancements at the reservoir hardware level, as it is not inherently a constraint of PGT itself. Furthermore, integrating physical reservoir transformers into existing AI ecosystems will necessitate improvements in hardware-software co-design to enhance scalability and reduce latency. The other major hurdle for PGT is that it is currently not able to match the best training losses achieved through current state of the art backpropagation based methods. Further work needs to be done to address this limitation and refine the method. Even with these limitation, PGT offers a pathway to significant energy cost savings, potentially reducing AI/ML operational expenses by hundreds of millions of dollars.

On the physics side, future research should focus on optimizing reservoir designs to support parallel processing and exploring hybrid architectures that combine PGT-trained physical reservoirs with GPU-based systems. Such innovations could significantly accelerate training times while preserving the energy efficiency advantages of physical reservoirs. On the computer science side, further research on PGT should look to refine the training methodology to achieve minimum losses similar to that achieved with backpropagation.

In conclusion, the introduction of Perturbative Gradient Training marks a significant milestone in the field of physical reservoir computing. Much like the impact of stochastic gradient descent (SGD) on the development of modern AI, we hope that PGT will inspire a wave of innovation and research. By bridging the gap between physical reservoirs and state-of-the-art AI systems, this paradigm has the potential to drive sustainable and transformative advancements in artificial intelligence.

VI. Acknowledgments

Authors acknowledge the support from the National Science Foundation of the United States by Grant No. ECCS-2138236.

References

[1] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. Aga, J. Huang, C. Bai *et al.*, "Sustainable ai: Environmental implications, challenges and opportunities," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 795–813, 2022.

[2] W. Knight, "OpenAI's CEO Says the Age of Giant AI Models Is Already Over — wired.com," https://www.wired.com/story/openai-ceo-sam-altman-the-age-of-giant-ai-models-is-already-over, [Accessed 14-03-2025].

[3] G. Wilkins, S. Keshav, and R. Mortier, "Hybrid heterogeneous clusters can lower the energy consumption of llm inference workloads," in *The 15th ACM International Conference on Future and Sustainable Energy Systems*, ser. e-Energy '24. ACM, May 2024, p. 506–513. [Online]. Available: http://dx.doi.org/10.1145/3632775.3662830

[4] S. Samsi, D. Zhao, J. McDonald, B. Li, A. Michaleas, M. Jones, W. Bergeron, J. Kepner, D. Tiwari, and V. Gadepally, "From words to watts: Benchmarking the energy costs of large language model inference," in *2023 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, Sep. 2023, p. 1–9. [Online]. Available: http://dx.doi.org/10.1109/HPEC58863.2023.10363447

[5] A. George, A.S.Hovan George, and A.S.Gabrio Martin, "The environmental impact of ai: A case study of water consumption by chat gpt," 2023. [Online]. Available: https://zenodo.org/record/7855594

[6] "Q&A: UW researcher discusses just how much energy ChatGPT uses — washington.edu," https://www.washington.edu/news/2023/07/27/how-much-energy-does-chatgpt-use, [Accessed 14-03-2025].

[7] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, p. 127–149, Aug. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2009.03.005

[8] S. Shen, A. Baevski, A. S. Morcos, K. Keutzer, M. Auli, and D. Kiela, "Reservoir transformers," 2020. [Online]. Available: https://arxiv.org/abs/2012.15045

[9] I. S. Maksymov, "Analogue and physical reservoir computing using water waves: Applications in power engineering and beyond," *Energies*, vol. 16, no. 14, p. 5366, Jul. 2023. [Online]. Available: http://dx.doi.org/10.3390/en16145366

[10] P. Pirro, V. I. Vasyuchka, A. A. Serga, and B. Hillebrands, "Advances in coherent magnonics," *Nature Reviews Materials*, vol. 6, no. 12, p. 1114–1135, Jul. 2021. [Online]. Available: http://dx.doi.org/10.1038/s41578-021-00332-w

[11] D. Lachance-Quirion, Y. Tabuchi, A. Gloppe, K. Usami, and Y. Nakamura, "Hybrid quantum systems based on magnonics," *Applied Physics Express*, vol. 12, no. 7, p. 070101, Jun. 2019. [Online]. Available: http://dx.doi.org/10.7567/1882-0786/AB248D

[12] B. Rana and Y. Otani, "Towards magnonic devices based on voltage-controlled magnetic anisotropy," *Communications Physics*, vol. 2, no. 1, Aug. 2019. [Online]. Available: http://dx.doi.org/10.1038/s42005-019-0189-6

[13] A. V. Chumak, A. A. Serga, and B. Hillebrands, "Magnonic crystals for data processing," *Journal of Physics D: Applied Physics*, vol. 50, no. 24, p. 244001, May 2017. [Online]. Available: http://dx.doi.org/10.1088/1361-6463/aa6a65

[14] A. V. Chumak, *Magnon Spintronics*. CRC Press, May 2019, p. 247–302. [Online]. Available: http://dx.doi.org/10.1201/9780429423079-6

[15] A. V. Sadovnikov, E. N. Beginin, M. A. Morozova, Y. P. Sharaevskii, S. V. Grishin, S. E. Sheshukova, and S. A. Nikitov, "Nonlinear spin wave coupling in adjacent magnonic crystals," *Applied Physics Letters*, vol. 109, no. 4, Jul. 2016. [Online]. Available: http://dx.doi.org/10.1063/1.4960195

[16] E. Albisetti, D. Petti, G. Sala, R. Silvani, S. Tacchi, S. Finizio, S. Wintz, A. Calò, X. Zheng, J. Raabe, E. Riedo, and R. Bertacco, "Nanoscale spin-wave circuits based on engineered reconfigurable spin-textures," *Communications Physics*, vol. 1, no. 1, Sep. 2018. [Online]. Available: http://dx.doi.org/10.1038/s42005-018-0056-x

[17] A. N. Mahmoud, F. Vanderveken, C. Adelmann, F. Ciubotaru, S. Cotofana, and S. Hamdioui, "Spin wave normalization toward all magnonic circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 1, p. 536–549, Jan. 2021. [Online]. Available: http://dx.doi.org/10.1109/TCSI.2020.3028050

[18] F. Heussner, A. A. Serga, T. Brächer, B. Hillebrands, and P. Pirro, "A switchable spin-wave signal splitter for magnonic networks," *Applied Physics Letters*, vol. 111, no. 12, Sep. 2017. [Online]. Available: http://dx.doi.org/10.1063/1.4987007

[19] C. B. Abbott and D. A. Bozhko, "Hybrid magnonic reservoir computing," 2024. [Online]. Available: https://arxiv.org/abs/2405.09542

[20] W. Namiki, D. Nishioka, Y. Nomura, T. Tsuchiya, K. Yamamoto, and K. Terabe, "Iono–magnonic reservoir computing with chaotic spin wave interference manipulated by ion-gating," *Advanced Science*, vol. 12, no. 3, Nov. 2024. [Online]. Available: http://dx.doi.org/10.1002/advs.202411777

[21] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1139–1147.

[22] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations (ICLR)*, 2017.

[23] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, p. 625–644, Aug. 2021. [Online]. Available: http://dx.doi.org/10.1038/s42254-021-00348-9

[24] O. M. William Wolberg, "Breast cancer wisconsin (diagnostic)," 1993. [Online]. Available: https://archive.ics.uci.edu/dataset/17

[25] L. Bottou, *Large-Scale Machine Learning with Stochastic Gradient Descent*. Physica-Verlag HD, 2010, p. 177–186. [Online]. Available: http://dx.doi.org/10.1007/978-3-7908-2604-3_16

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[27] D. Elliott, S. Frank, K. Sima'an, and L. Specia, "Multi30k: Multilingual english-german image descriptions," in *Proceedings of the 5th Workshop on Vision and Language*. Association for Computational Linguistics, 2016. [Online]. Available: http://dx.doi.org/10.18653/v1/W16-3210

[28] S. Watt and M. Kostylev, "Numerical simulations of a magnonic reservoir computer," *Journal of Applied Physics*, vol. 135, no. 2, Jan. 2024. [Online]. Available: http://dx.doi.org/10.1063/5.0184848

[29] S. Watt, M. Kostylev, A. B. Ustinov, and B. A. Kalinikos, "Implementing a magnonic reservoir computer model based on time-delay multiplexing," *Physical Review Applied*, vol. 15, no. 6, Jun. 2021. [Online]. Available: http://dx.doi.org/10.1103/PhysRevApplied.15.064060

[30] A. B. Ustinov, R. V. Haponchyk, and M. Kostylev, "A current-controlled magnonic reservoir for physical reservoir computing," *Applied Physics Letters*, vol. 124, no. 4, Jan. 2024. [Online]. Available: http://dx.doi.org/10.1063/5.0189542

[31] T. Furuta, K. Fujii, K. Nakajima, S. Tsunegi, H. Kubota, Y. Suzuki, and S. Miwa, "Macromagnetic simulation for reservoir computing utilizing spin dynamics in magnetic tunnel junctions," *Physical Review Applied*, vol. 10, no. 3, Sep. 2018. [Online]. Available: http://dx.doi.org/10.1103/PhysRevApplied.10.034063

[32] N. Bertschinger, T. Natschläger, and R. Legenstein, "At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks," *Advances in neural information processing systems*, vol. 17, 2004.