

Annealing-inspired training of an optical neural network with ternary weights.

Corresponding Author: Mr Anas Skalli

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

The manuscript by Mr Skalli and co-workers titled "Annealing-inspired training of an optical neural network with ternary weights" demonstrates a novel learning algorithm on an optical reservoir computer. The optical reservoir consists of a multimode fiber and a large area VCSEL. The optical inputs are spatially profiled beams created with a DMD. The inputs are inserted into the reservoir by coupling the beam into a multimode fiber that directs it onto the VCSEL. The light emitted by the VCSEL is spatially filtered by another DMD, realizing an optical output layer, and measured on a photodetector. The main contributions of this manuscript are 1) the demonstration of ternary weights in the output layer, and 2) the conception and implementation of a novel learning algorithm to train the weights of the output layer more efficiently. The implementation of ternary weights is realized by a double-pass method that combines two sequential readings from the reservoir, one with positive weights, and a second one with negative weights. The novel learning algorithm is likened to simulated annealing, with an annealing-schedule that is given by the error, such that for a large error, many weights are changed (temperature is high) and for a small error only a few weights are changed (temperature is low). These techniques are demonstrated on a one-vs-all MNIST classification task, in which the goal is to recognize one particular digit among a dataset of all digits.

The major claims of the paper are that 1) ternary weights significantly improve the performance of the optical neural network, 2) the novel learning algorithm significantly improves convergence speed and performance compared to naïve random search, and 3) this ONN exhibits exceptional stability.

The implementation of ternary weights is creative, appears very effective and is to my knowledge novel. Similar schemes exist, e.g. pulse-width modulation, superpixels (for non-binary modulation of light with DMDs [1,2]) or balanced photodetection (for negative weights). These techniques are known in the optical neural network community, but I have not seen this particular scheme yet. It is interesting how much the ONN performance improves by using ternary vs binary weights, though I am uncertain how much of this is due to using three weight levels vs negative weights [3] (i.e., how would a binary output layer with values $-1/1$ do?). It would be helpful if the authors could clarify this point.

The novel learning algorithm is an interesting and apparently effective combination of random search and simulated annealing. This will be a welcome addition in the toolbox of algorithms with little memory and computing needs. However, there are two points that I wish the authors would comment on: 1) Training the output weights of a reservoir computer is a rather simple task (equivalent to a matrix-inversion). I understand that this requires memory to store and computation to invert the matrix, but I wonder how much more or less energy it takes to train with an iterative training algorithm for 1000 epochs. 2) I believe there are simulated-annealing algorithms that like the algorithm here make the temperature a dynamic variable dependent on the value of the cost function, e.g. thermodynamic simulated annealing or adaptive simulated annealing. It would be great if the authors could put their algorithm in perspective with regards to those (or other) related algorithms.

Lastly the authors claim that their ONN is extremely stable, thereby "addressing one of the main concerns in the field". Optical stability is indeed a major concern that hinders the development of larger ONNs. There is a brief sentence describing what the authors did to achieve this stability (PID temperature control and mechanically securing the MMF), but with this manuscript, it is difficult for the community to learn from the author's achievement. It would be helpful if the authors could elaborate on which techniques have contributed the most to the excellent stability.

Overall, the manuscript presents interesting and helpful, albeit incremental, findings for the optical neural network community. I recommend this manuscript for publication in Nature Communications Physics, if the authors revise the text and include some of the suggestions from above. The figure captions in their current form are also insufficient and need to

be extended, as well as typographical errors fixed.

[1] Goorden, Sebastianus A., Jacopo Bertolotti, and Allard P. Mosk. "Superpixel-based spatial amplitude and phase modulation using a digital micromirror device." *Optics express* 22.15 (2014): 17999-18009.

[2] Popoff, Sébastien M., et al. "A practical guide to digital micro-mirror devices (dmds) for wavefront shaping." *Journal of Physics: Photonics* 6.4 (2024): 043001.

[3] Wang, Qingyang, et al. "Why do networks have inhibitory/negative connections?." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

[4] De Vicente, Juan, Juan Lanchares, and Román Hermida. "Placement by thermodynamic simulated annealing." *Physics Letters A* 317.5-6 (2003): 415-423.

Reviewer #2

(Remarks to the Author)

The authors propose a training method for an optical neural network with ternary weights and demonstrate its effectiveness through photonic implementation. The proposed algorithm is inspired by simulated annealing, in which they determine the number of the weight to flip based on the squared error (NMSE) between the measured output y and the desired output y_t . The method is quite simple, and the results from physical experiments clearly support the effectiveness of the method. Additionally, thanks to the simplicity of the algorithm, adaptive training against the drift in the physical hardware is possible. The experiments show that performance can be maintained over 10 hours even under such physical fluctuation. These findings will likely provide interesting insights into the various researchers in the field of optical computing and physical reservoir computing. Although I am generally positive for publishing this work, I have also several significant concerns listed below before the acceptance.

Major issues:

As the physical hardware used in this report is based on the author's previous work [1], I think the primary novelty of this paper lies in the algorithm part. Therefore, comparisons with other algorithms and a discussion on the differences from previous works are necessary. Although differences from the authors own prior work are mentioned, I think that a broader comparison with other previous studies should be required. (Note: There is also a proposal regarding how to compute ternary values $(-1, 0, 1)$ with hardware that can only take binary values $(0, +1)$, but since this has already been demonstrated in existing reports (e.g., [2]), I consider this to be a trivial contribution. I think previous work such as [2] should be cited.)

Specifically, I think the following is necessary:

On reference:

1. The paper should cite existing studies on binary and ternary neural networks (e.g., [3], [4]). Also, briefly summarization of the progress in BNN and the advantages of ternary neural networks in the ML field should be described in introduction or supplementary information. In the current manuscript, there is only a brief mention of the 1.56-bit transformer.

2. One of the advantages of this method seems to be that it does not require gradient information from the physical system and can train the ONN based on the output. Recently, many backpropagation-free algorithms have been proposed (e.g., [5]-[8]) based on similar motivation. Thus, I think that brief description of the differences and the effectiveness of this method should be added for the comparison.

On performance:

3. Training of neural networks using simulated annealing has been studied in previous research (e.g., [9]). However, in general, it is reported that they take a long time to converge comparing with another method such as back propagation. Please describe the convergence behavior of your method in comparison to other algorithms. (From Fig. 4, it seems to converge in around 100 epochs, but since Algorithm 1 includes the if-statement "if MSE temp < MSE best then", I assume there are many cases where it fails (most trial seems to go else part). Please take this into account for the fair comparison.)

4. In Figure 6, the comparison is made with a digital linear layer, which does not seem appropriate as a baseline to demonstrate the effectiveness of the algorithm. I think you should add the result of standard RC (or extreme learning machine, ELM) with linear regression learning of W^{out} with floating point and/or quantized weight value. If non-linear modeling of your hardware is difficult, a comparison with tanh-based RC or ELM would be acceptable. Alternatively, the physical system in the proposed method seems to have a camera for measuring the reservoir state x , so you could also determine the W^{out} through linear regression using measured x experimentally.

5. What about the scalability of this method? For instance, is this method only effective for W^{out} , or can it be applied to all layers of a multi-layer NN, including W^{in} of (physical) RC? The training of the final layer weights W^{out} seems much easier than another layer due to the simplicity of the loss function. Whether this method can be applied to other layers is a

significant point of interest.

Minor issues:

6. Related to the comment 2, a key advantage of your method is that it allows learning based only on the low-dimensional output data y , without measurement of the high-dimensional reservoir state x . If this understanding is correct, I recommend adding to discuss it briefly because this point is not explicitly stated in the manuscript.

7. Why does the NMSE occasionally increase in Figure 4? In Algorithm 1, it seems weights are always updated to the direction for reducing NMSE.

8. When α is small in Algorithm 1, it seems to correspond to a low-temperature state in simulated annealing, making it difficult to escape local minima. Larger values of α seem to correspond to high-temperature states in simulated annealing, which may help escape local minima. If this understanding is correct, supplementing this explanation would enhance the readers understanding.

[1] <https://opg.optica.org/ome/fulltext.cfm?uri=ome-12-7-2793&id=477351>

[2] <https://www.nature.com/articles/s41586-020-03070-1>

[3] <https://arxiv.org/abs/1602.02830>

[4] <https://arxiv.org/abs/1609.00222>

[5] <https://www.nature.com/articles/s41467-022-35216-2>

[6] <https://www.science.org/doi/abs/10.1126/science.adi8474>

[7] <https://www.nature.com/articles/s41586-024-07687-4>

[8] <https://journals.aps.org/prapplied/abstract/10.1103/PhysRevApplied.19.054002>

[9] <https://www.sciencedirect.com/science/article/pii/S1877050915035759>

Reviewer #3

(Remarks to the Author)

This paper presents an advancement in optical neural networks through the introduction of ternary weights in a fully hardware-implemented system using a large-area vertical-cavity surface-emitting laser (LA-VCSEL) and digital micromirror devices (DMDs). Building on previous work that used Boolean weights, the authors propose a minimalist implementation of ternary weights without requiring any physical modifications to the existing hardware. They demonstrate improved classification accuracy on the MNIST dataset, using an annealing-inspired training algorithm to optimize the hardware. Additionally, the authors address concerns regarding the stability of semiconductor laser neural networks, showing that their system maintains highly stable performance over a period exceeding 10 hours.

The proposed approach appears innovative and promising. The manuscript is clearly organized and provides convincing evidence that the proposed method can enhance classification performance. However, I have concerns regarding the novelty and effectiveness of the approach, and I would like to raise several questions for clarification:

1. A key contribution of the paper is the use of ternary weights, but the effectiveness of this approach has not been thoroughly discussed. The authors show that classification accuracy improves compared to the Boolean weight update method. However, higher-resolution weights, such as those provided by a liquid-crystal spatial light modulator (e.g., LCOS-SLM), could potentially yield further improvements. Could the authors discuss this possibility?

2. The benefit of using ternary weights based on the DMD compared to high-resolution SLMs, such as training speed, efficiency, or long-term stability, has not been made clear. Could the authors provide a comparison between the two approaches?

3. As shown in Fig. 6, the performance of the ternary weights is lower than that of the digital linear classifier. The authors should discuss the limiting factors in their setup and suggest possible ways to improve performance.

4. In the current experimental setup, ternary weights are created sequentially, which could limit inference bandwidth and increase computational costs. The authors briefly mention the simultaneous creation of weights using optical copies of the LA-VCSEL image, but they do not provide details on how these optical copies are generated. Clarifying the experimental setup for implementing this method, or presenting preliminary results to support the scheme shown in Figure 3, would strengthen the manuscript.

5. The long-term stability is one of the promising results of this paper. However, it is unclear what factors contribute to the system's stability. The use of a multimode fiber and semiconductor laser in the setup may make the system susceptible to external environmental fluctuations, which typically cause drift. The authors should discuss the stability of their setup and whether the use of Boolean hardware (e.g., DMD) contributes to this stability.

In addition to these concerns, I would like the authors to address the following points:

- The authors refer to their system as a reservoir computing (RC) system. However, since the LA-VCSEL is operated in its steady state during the experiment, it does not act as a recurrent neural network. This suggests that it may function more as an extreme learning machine, rather than a RC system. Could the authors comment on this point?

- In Section 2.3, the authors mention that " $\alpha = 0$ corresponds to the original algorithm." However, " $\alpha = 0$ " implies $n^{\{\text{mirrors}\}}$ is always zero in Eq. (2). Additionally, the authors should provide a citation for the "original algorithm."

- In Section 2.3, could the authors compare the convergence speed of the proposed in-situ algorithm with that of conventional simulated annealing?

- In Section 3.1, the authors introduce the "one-vs-all" classification method but do not provide an explanation of what it entails. A brief explanation of the "one-vs-all" classification approach would be helpful for readers unfamiliar with the method.

- The "one-vs-all" classification approach has limitations [Bishop 2006], especially in handling a large number of classes, as it can become computationally expensive. It would be better if the authors could discuss why they chose this approach and what advantages it offers in the context of their hardware implementation. Reference: [Bishop 2006] C. M. Bishop, Pattern Recognition and Machine Learning. New York: Springer, 2006.

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

The authors have addressed my comments adequately, and for my part I believe the work should be ready for publication. My main comments were:

1) How much of the performance boost is due to using three weight levels vs negative weights [3] (i.e., how would a binary output layer with values -1/1 do?)

The authors have satisfactorily addressed this claim by a paragraph at the end of section 3.4 and an additional figure in Appendix D.

2) Two points regarding the algorithm:

a) How much more or less energy it takes to train with an iterative training algorithm for 1000 epochs.

The authors have presented a convincing case that matrix-inversion would not be significantly cheaper in energy, but would suffer from other problems.

b) It would be great if the authors could compare their algorithm against thermodynamic simulated annealing or related algorithms.

The authors compared their algorithm against thermo-adaptive SA and showed that it achieves similar performance but much higher convergence speed.

3) That the authors elaborate on which techniques have contributed the most to the excellent stability.

The authors explain that they use a short MMF, a DMD with only two possible states, and a cage-system to stability component drifts.

Reviewer #2

(Remarks to the Author)

I appreciate that the authors have addressed all our concerns. I recommend this paper for the publication.

one small comment:

I have understood your comment on R2-Q5: "The reviewer raises an interesting question regarding the ability of our algorithm to scale to a potentially trainable input layer. However, this aspect lies beyond the scope of this current manuscript and would require some significant and detailed investigation, which due to the significant interest we are currently in the process of exploring". I recommend describing that the scalability of this algorithm remains an open question in the final version of paper.

Reviewer #3

(Remarks to the Author)

The authors have adequately addressed my previous concerns regarding their manuscript. Considering the improvements made and the overall quality of the revised version, I recommend its acceptance for publication.

Open Access This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

Dear reviewers,

We would like to thank you for your positive evaluation of our work and your relevant comments and suggestions. We find that the manuscript's quality has improved thanks to your input. In the following, we present a detailed point-by-point reply to your comments, indicating where the manuscript has been modified in accordance. We also included in the attached documents:

- 1) Updated version of the manuscript : "Trinary_weights_new_version.pdf"
- 2) Marked up pdf highlighting the changes to the manuscript: "Trinary_weights_latex_diff"

With kind regards from all the authors,

Anas SKALLI.

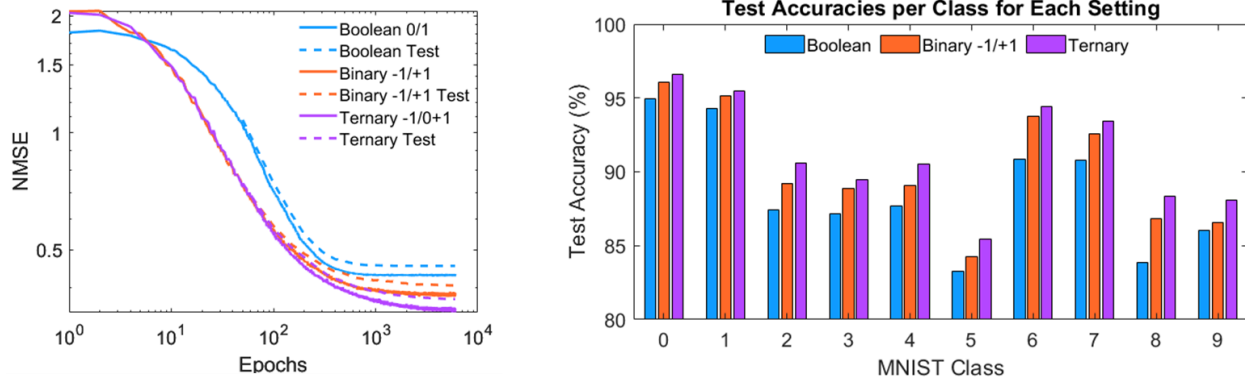
Reviewer 1 :

We thank the reviewer for the positive words and for the relevant questions and suggestions. We made several additions to the manuscript in order to address the points raised by the reviewer. We also rewrote the figure captions to be more descriptive as requested by the reviewer to aid reader understanding.

R1-Q1)The implementation of ternary weights is creative, appears very effective and is to my knowledge novel. Similar schemes exist, e.g. pulse-width modulation, superpixels (for non-binary modulation of light with DMDs [1,2]) or balanced photodetection (for negative weights). These techniques are known in the optical neural network community, but I have not seen this particular scheme yet. It is interesting how much the ONN performance improves by using ternary vs binary weights, though I am uncertain how much of this is due to using three weight levels vs negative weights [3] (i.e., how would a binary output layer with values $-1/1$ do?). It would be helpful if the authors could clarify this point.

The reviewer raises an interesting point. One could assume that some nodes of an ONN have anticorrelated responses to the desired target output, in which case the -1 weights help by allowing the network to leverage these anticorrelated responses. In contrast, if the only available options are $0/1$ then these nodes are assigned 0 , therefore reducing network dimensionality. However, it is also reasonable to assume that not all nodes are beneficial, in which case having the possibility of assigning a 0 weight is very useful to discard responses from potentially uncorrelated nodes.

To illustrate this we trained a software ELM of a representative size to our experiment, i.e.100 neurons, with our annealing inspired algorithm in 3 different output weight configurations: $0/1$; $-1/+1$; and $-1/0/+1$:



Left: ELM 100 neurons trained on on-vs-all MNIST for different weight resolutions. Right: One-vs-all test accuracy for each class for different weight resolutions.

The possibility of having 0 even when negative weights are available clearly improves performance in this case. Boolean weights achieve on average 88.6%, binary positive and negative weights i.e. -1/+1 achieves 90.2%, while ternary weights achieve 91.2%. Showing that there is a significant improvement of 1% when simply adding a “0” level. We included a sentence in section “3.4 Benefits of ternary weights”:

“””

In addition, we performed simulations comparing an ELM of 100 neurons trained with our annealing inspired algorithm in 3 different output weight configurations: Boolean 0/1; Binary -1/+1; and Ternary -1/0/+1. The possibility of having 0 even when negative weights are available clearly improves performance in certain architectures. Boolean weights achieve on average 88.6%, binary positive and negative weights i.e. -1/+1 achieves 90.2%, while ternary weights achieve 91.2%. Showing that there is a significant improvement when simply adding a “0” level.

“””

We also added the plot above to an appendix.

R1-Q2) Training the output weights of a reservoir computer is a rather simple task (equivalent to a matrix-inversion). I understand that this requires memory to store and computation to invert the matrix, but I wonder how much more or less energy it takes to train with an iterative training algorithm for 1000 epochs.

The reviewer raises an important question. In our manuscript we represent a position opposed to the idea of offline training via matrix inversion. Indeed, training in one step via matrix inversion seems rather simple at first, given that we can just take camera images of the VCSEL mode profile and use those as a response matrix to train offline weights in one step. However, apart from being very expensive, fast cameras are highly energy consuming so that their employment would by far consume more energy than the repeated optimization, meaning we would need to significantly slow down the injection rate of the DMD. Furthermore, the finally integrated weighted response on our ‘bucket detector’ will very likely deviate from the simple linear sum, and the matrix inversion would first require extensive beam propagation simulations including DMD diffraction and imaging

aberrations - not very practical all together. Advancing the same argument, in the future we target inference at much higher data rates on the order of GHz. Our VCSEL has sub-nanosecond response time, which does not allow the use of cameras and makes the process of training via matrix inversion more complex. Moreover, we find that offline weights are not inline with the spirit of full hardware implementation for which we strive in this manuscript, they can however be valuable to set an upper bound on performance.

Regarding the concern about the cost to train with our algorithm compared to matrix inversion, our algorithm can converge in about ~200 epochs which means that we measure the output 200 times. In addition, in the context of limited weight resolution such as Boolean or trinary, we need to consider that matrix inversion is not applicable. Similarly, our algorithm is specifically tailored to low resolution and not really applicable (in its current form) to continuous problem optimization. We included a paragraph discussing this in the manuscript in section “3.3 Hardware learning”

“”

Our adaptive search strategy offers significant gains both in terms of performance and convergence speed with minimal computational overhead with no changes to the existing hardware, while being compatible with both Boolean and ternary weights. Finally, in the context of RC or ELM, training of the linear output weights is often realized via a "single-shot" approach via matrix inversion as follows:

$$\begin{equation} \{W\}^{\{\text{out}\}} = \mathbf{X}^{\{\dagger\}} \{y\}^{\{\text{train}\}}, \end{equation}$$

where $\dagger$, is the Moore-Penrose inverse, $\mathbf{X}$ the so called state collect matrix containing the response of all nodes or neurons to each input example and $\{y\}^{\{\text{train}\}}$ is the training set target for each corresponding input example.

It is therefore interesting in the context of hardware learning to compare the computational training cost of matrix inversion training with iterative methods such as the one presented in this work.

While it is true that training via matrix inversion is potentially realized in a single step, the physical nature of a scheme such as the one presented here requires measuring responses of each node individually to obtain all information to construct the state collect matrix.

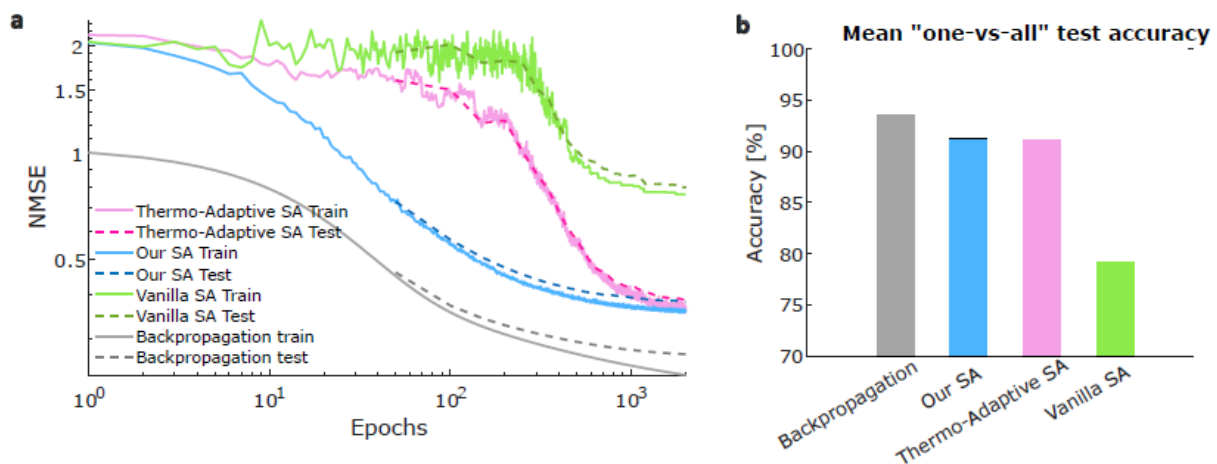
The associated measurement process scales with the number of neurons, $n_{\{\text{neurons}\}} = 450$ in our case. From results presented in Fig. \ref{fig:alphascan_MNIST} we can see that our SA-inspired algorithm converges in ~ 200 epochs, moreover convergence scales with network size as previously shown in \cite{andreoli2020boolean}. As a consequence, training with matrix inversion or via our iterative method have comparable computational overhead in principle since both scale linearly with network size. However, offline weight computation using matrix inversion is very likely not going to correspond with high accuracy to the physically detected output signal. For that, various optical effects such as diffraction at the DMD, aberrations by the imaging optics and detection nonlinearity have to be included. The impact of both will differ for each $\{W\}^{\{\text{out}\}}$, rendering matrix inversion for ‘single-shot’ training of physical hardware likely inapplicable.

R1-Q3) I believe there are simulated-annealing algorithms that, like the algorithm here, make the temperature a dynamic variable dependent on the value of the cost function, e.g. thermodynamic simulated annealing or adaptive simulated annealing. It would be great if the authors could put their algorithm in perspective with regards to those (or other) related algorithms.

We thank the reviewer for raising this valid point that will help place our algorithm in context. First of all, comparison with conventional simulated annealing (SA) is tricky since originally SA was defined for continuous optimization. Here, we are focusing on ternary level weights. Nevertheless, in order to respond to the reviewer's suggestion, we adapted conventional SA methods and compared them with our own algorithm.

We compare the following 3 variants on the original SA algorithm each with their own hyperparameters (HP):

- Vanilla SA: all mirrors are flipped at every epoch, configurations which yield improvements are always accepted while the probability to accept worse configurations decreases with temperature as the system cools down according to a fixed schedule $T_{k+1} = T_k * \text{Beta}$ – 2HP: initial temperature T_0 , cooling_rate Beta. Accepting worse configurations allows the algorithm to escape local minima.
- Thermo-Adaptive SA: the number of mirrors as well as the probability to accept worse configurations is dependent on a schedule. – 3HP: initial temperature T_0 , Beta, alpha proportionality factor $n_{\text{mirrors}} = \text{floor}(\alpha * (1 - \text{epoch} / \text{epochs})) + 1$.
- Our SA: the number of mirrors is directly proportional to the NMSE 1HP alpha: $n_{\text{mirrors}} = \text{floor}(\alpha * \text{MSE})) + 1$



Comparison between different simulated annealing (SA) algorithms. (a) Convergence of different SA algorithms on a ternary output layer of an ELM with 100 neurons. The "Vanilla", "Thermo-Adaptive", and our original SA variant are

shown and compared with a backpropagation baseline training of the ELM using full resolution weights. (b) Mean "one-vs-all" accuracy averaged across all classes of the MNIST dataset for the different algorithms shown in (a).

From our analysis, after hyperparameter tuning our novel algorithm is very efficient when it comes to optimizing trinary problems, it reaches the same performance level as the "thermo-Adaptive" variant close to an order of magnitude faster, while only requiring optimizing a single hyperparameter. To account for this interesting finding, we added a full section in the manuscript comparing our SA with other conventional SA approaches: "3.2 Comparison with conventional simulated annealing"

R1-Q4) Lastly the authors claim that their ONN is extremely stable, thereby "addressing one of the main concerns in the field". Optical stability is indeed a major concern that hinders the development of larger ONNs. There is a brief sentence describing what the authors did to achieve this stability (PID temperature control and mechanically securing the MMF), but with this manuscript, it is difficult for the community to learn from the author's achievement. It would be helpful if the authors could elaborate on which techniques have contributed the most to the excellent stability.

Again the reviewer's suggestion is very valid point.

In our case, we use a short MMF (1 m) which is inherently less susceptible to variations of its output speckle pattern than a longer MMFs. Studying the correlation of the speckle with camera images taken over a period of 4 days yields above 90% correlation. Longer fibers are more sensitive to external perturbations (like vibrations, temperature fluctuations, or mechanical stress), leading to changes in the modal distribution for smaller fluctuations in the experiment's environment.

In addition, the entire experimental setup is constructed based on a cage-system layout so that all elements which can be linked are fixed together, preventing independent drifts. The use of a DMD, providing only 2 different possible states for its pixels, is also inherently less prone to drifts compared to analog valued weights as would be implemented with a liquid crystal on silicon Spatial Light Modulator.

All the elements mentioned in our answer were incorporated in a new paragraph in section "3.5 Long term stability characterization":

“””

In addition, we use a short MMF with a length of 1 m, which is inherently less susceptible to variations in its output speckle pattern compared to longer fibers. Indeed, longer MMFs are more vulnerable to external perturbations such as vibrations, temperature fluctuations, and mechanical stress. Moreover, the use of a DMD, providing only two different possible states for its pixels in the form of two different mechanical positions is also inherently less prone to drifts compared to analog valued weights as would be implemented with a liquid crystal on silicon Spatial Light Modulator. Finally, the entire experimental apparatus is constructed based on a cage-system, preventing component- independent drifts.

Reviewer 2 :

We thank the reviewer for raising constructive criticism and interesting questions that helped us improve the quality of the manuscript. We addressed the reviewer's questions where possible and included the suggested references and some additional ones.

Major issues:

As the physical hardware used in this report is based on the author's previous work [1], I think the primary novelty of this paper lies in the algorithm part. Therefore, comparisons with other algorithms and a discussion on the differences from previous works are necessary. Although differences from the authors own prior work are mentioned, I think that a broader comparison with other previous studies should be required. (Note: There is also a proposal regarding how to compute ternary values (-1, 0, 1) with hardware that can only take binary values (0, +1), but since this has already been demonstrated in existing reports (e.g., [2]), I consider this to be a trivial contribution. I think previous work such as [2] should be cited.)

Specifically, I think the following is necessary:

On reference:

R2-Q1) The paper should cite existing studies on binary and ternary neural networks (e.g., [3], [4]). Also, briefly summarization of the progress in BNN and the advantages of ternary neural networks in the ML field should be described in introduction or supplementary information. In the current manuscript, there is only a brief mention of the 1.56-bit transformer.

To address the comment raised by the reviewer we reworked the paragraph mentioned in section 1 :

Binary neural networks leveraging Boolean weights and activation functions have garnered significant attention as they are inherently simpler. Indeed, complex operations such as multiplication can be replaced with simpler bitwise operations such as XNOR and population count, which drastically reduce computational overhead while maintaining reasonable accuracy in certain tasks \cite{courbariaux2016binarized, hubara2016binarized, rastegari2016xnor}. Despite these advantages, the simplicity of binary weights can limit the representational capacity of the network, which may impact performance on more complex tasks. For additional context, we refer to the reviews on binary neural networks in \cite{qin2020binary, yuan2023comprehensive}.

In the context of physically implemented neural networks, binary weights have additional appeal as they do not require high-complexity control. Using bistable systems such as ferro-electric devices offers a clear avenue towards non-volatile weights, with the associated substantial gains in energy efficiency and system simplicity.

Ternary weights provide a richer representation, which substantially improves the learning capability of the network while maintaining low system complexity and energy costs. This added expressiveness significantly enhances the learning ability of the network, enabling improved accuracy compared to binary approaches, without a proportional increase in computational overhead \cite{alemdar2017ternary, zhu2016trained, mishra2017wrpn}.

As a consequence, there has been growing interest in applications of ternary weights in ANNs for increased efficiency, for instance in the context of large language models \cite{zhu2024scalable}, as well as the context of electronic hardware implementations \cite{hirtzlin2020digital, laborieux2020low, laborieux2021synaptic}.

Although binary and ternary ANNs have been extensively studied, comparisons in the context of analog unconventional physical ANNs remain scarce.

“””

R2-Q2) One of the advantages of this method seems to be that it does not require gradient information from the physical system and can train the ONN based on the output. Recently, many backpropagation-free algorithms have been proposed (e.g., [5]-[8]) based on similar motivation. Thus, I think that a brief description of the differences and the effectiveness of this method should be added for the comparison.

To address the valid comment made by the reviewer we made the following changes:

Section 1 Introduction -> Reworked the paragraph introducing different hardware compatible methods including the suggested references

“””

With the objective of leveraging unconventional ANN hardware to solve a particular task, there has been a significant advancement in hardware-compatible training algorithms \cite{wright2022deep, momeni2024training, momeni2023backpropagation, nakajima2022physical, xue2024fully}. These advances encompass a variety of approaches, including model-based techniques like backpropagation using a digital twin \cite{wright2022deep}, augmented direct feedback alignment \cite{nakajima2022physical} as well as experimental methods for implementing in-situ backpropagation \cite{xue2024fully, pai2023experimentally}. Additionally, model-free or black-box strategies have gained traction \cite{skalli2022computational, andreoli2020boolean, momeni2024training, mccaughan2023multiplexed}.

Yet, fully realized analog hardware implementations of these techniques remain scarce. For unconventional neural networks to be truly competitive, implementing in-situ training techniques is essential to overcome numerous bottlenecks and to reduce dependence on external high-performance computers, whose usage generally challenges the overall benefit of unconventional ANN computing substrates in general \cite{brunner2021competitive}.\hfill\break

“””

Section 3.3 Hardware Learning -> Added the following paragraph:

“””

The merit of our algorithm is that just like other model-free methods it allows optimization based solely on measurements of the output data y and hence offers the potentially lowest implementation complexity. In addition, relying only on simple random mutations does not involve heavy computations while being specifically designed to handle discrete problems. Furthermore, we found it to be more efficient than traditional SA methods in terms of performance and convergence speed. Unlike methods such as those described in \cite{wright2022deep,nakajima2022physical}, where a digital twin or augmented direct feedback alignment are employed to estimate gradients or error signals, our approach does not require access or knowledge about internal variables of the physical system such as neuron activations, which can be prohibitive as LA-VCSELs are notoriously complex to probe and simulate \cite{mulet2002spatio,mulet2001transverse}. This highlights that in potentially numerous physical ANN implementations, model-free optimization or physical backpropagation may potentially be the only realistically available training options.

“””

We also added section 3.2 comparing our algorithm with other SA variants.

We understand the appeal of a more thorough comparison with all hardware compatible algorithms present in the literature. However, it would require significantly more work and is beyond the scope of the current algorithm and manuscript. We have some time ago started such work, research is in its final stages, and in particular our detailed algorithm-overhead analysis clearly shows that the costs of physically information training methods - besides a potential exception of feedback alignment - is prohibitive. It cannot be won back during the lower-power inference stage based only on the unconventional ANN hardware.

On performance:

R2-Q3) Training of neural networks using simulated annealing has been studied in previous research (e.g., [9]). However, in general, it is reported that they take a long time to converge compared with another method such as back propagation. Please describe the convergence behavior of your method in comparison to other algorithms. (From Fig. 4, it seems to converge in around 100 epochs, but since Algorithm 1 includes the if-statement "if MSE temp < MSE best then", I assume there are many cases where it fails (most trials seem to go else part). Please take this into account for the fair comparison.)

We thank the reviewer for this comment. We would like to point out that, importantly, our learning curves include all trials, the if-statement "if MSE temp < MSE best then" does not mean that we do not show the error when a random mutation (i.e. changing weights) leads to a reduced performance. Indeed, the learning curves show many such situations. Regarding comparison with other

algorithms, we would like to refer to our answer to the 3rd question raised by reviewer 1 R1-Q3), where we compare different variants of SA to our discrete optimization problem and provide the baseline of backpropagation on a digital ELM, and to the discussions we have included in our manuscript in response to R2-Q2, and R1-Q1.

R2-Q4) In Figure 6, the comparison is made with a digital linear layer, which does not seem appropriate as a baseline to demonstrate the effectiveness of the algorithm. I think you should add the result of standard RC (or extreme learning machine, ELM) with linear regression learning of W^{out} with floating point and/or quantized weight value. If nonlinear modeling of your hardware is difficult, a comparison with tanh-based RC or ELM would be acceptable. Alternatively, the physical system in the proposed method seems to have a camera for measuring the reservoir state x , so you could also determine the W^{out} through linear regression using measured x experimentally.

To address the concern from the reviewer we added the test accuracy for each class in Figure 6 when using a 400 neuron ELM with both pseudoinverse training and our proposed algorithm for fair comparison. Unfortunately, we cannot determine the W^{out} through linear regression using an experimentally measured x . For the detailed reason, please see the final paragraph in our response to R1-Q2.

R2-Q5) What about the scalability of this method? For instance, is this method only effective for W^{out} , or can it be applied to all layers of a multi-layer NN, including W^{in} of (physical) RC? The training of the final layer weights W^{out} seems much easier than another layer due to the simplicity of the loss function. Whether this method can be applied to other layers is a significant point of interest.

The reviewer raises an interesting question regarding the ability of our algorithm to scale to a potentially trainable input layer. However, this aspect lies beyond the scope of this current manuscript and would require some significant and detailed investigation, which due to the significant interest we are currently in the process of exploring.

Minor issues:

R2-Q6) Related to the comment 2, a key advantage of your method is that it allows learning based only on the low-dimensional output data y , without measurement of the high-dimensional reservoir state x . If this understanding is correct, I recommend adding to discuss it briefly because this point is not explicitly stated in the manuscript.

As this point is related to R2-Q2) we have already addressed it in our corresponding reply.

R2-Q7) Why does the NMSE occasionally increase in Figure 4? In Algorithm 1, it seems weights are always updated to the direction for reducing NMSE.

As mentioned above in response to comment R2-Q3), we plotted all epochs not just the ones that lead to improvements. Another possible explanation is simply hardware related drift that can occasionally happen.

R2-Q8) When α is small in Algorithm 1, it seems to correspond to a low-temperature state in simulated annealing, making it difficult to escape local minima. Larger values of α seem to correspond to high-temperature states in simulated annealing, which may help escape local minima. If this understanding is correct, supplementing this explanation would enhance the reader's understanding.

The reviewer raises a valid point, and we added a sentence that reflects this description of our algorithm in section "2.3 In-situ Learning":

“”

As such, high values of α tend to correspond to high temperature states in SA which help escape local extrema.

“”

Reviewer 3 :

We thank the reviewer for the kind words and for the relevant questions that were asked. We made several additions to the manuscript in line with the reviewer's comments.

R3-Q1) A key contribution of the paper is the use of ternary weights, but the effectiveness of this approach has not been thoroughly discussed. The authors show that classification accuracy improves compared to the Boolean weight update method. However, higher-resolution weights, such as those provided by a liquid-crystal spatial light modulator (e.g., LCOS-SLM), could potentially yield further improvements. Could the authors discuss this possibility?

The reviewer raises an interesting point which we plan to fully investigate in future work, using SLMs with higher resolution weights entails applying a new learning algorithm which is beyond the scope of this present manuscript. From our current preliminary investigations additional weight resolution can offer significant performance improvements.

R3-Q2) The benefit of using ternary weights based on the DMD compared to high-resolution SLMs, such as training speed, efficiency, or long-term stability, has not been made clear. Could the authors provide a comparison between the two approaches?

The benefit of Using SLMs is that they natively provide high-resolution weights, yet most SLMs are significantly slower than DMDs, and regardless of speed they can be often at least twice as expensive. The limited resolution offered by DMD can also be somewhat mitigated via the use of superpixels, but since in this present work we were mainly concerned with optimizing in a discretized setting we did not investigate this. Moreover, the analog nature of SLMs makes them potentially more prone to drifts than the simpler design of DMDs, which only require mechanical flipping between 2 positions.

R3-Q3) As shown in Fig. 6, the performance of the ternary weights is lower than that of the digital linear classifier. The authors should discuss the limiting factors in their setup and suggest possible ways to improve performance.

We added the following paragraph to section 3.4 to discuss the points raised by the reviewer:

“”

Interestingly, our ternary weight ONN with the LA-VCSEL switched off performs better than when the LA-VCSEL is on and using Boolean weights, which indicates that at this very low-end of resolution, weight resolution is a severely limiting factor. Conceptually, because they are positive, Boolean weights cannot exploit nodes or neurons whose responses are anti-correlated with the desired target. The respective weights for these nodes are set to 0 when using Boolean weights which effectively restricts the dimensionality of our ONN.

This comparison highlights how crucial weight resolution and weights with a sign can be. Although the performance improvements when using ternary weights are substantial, our ONN still falls short and cannot outperform a digital linear classifier. It therefore follows that our next step should be to further improve our scheme, in order to overcome the digital linear limit. A rather non-invasive way of improving our current experiment would be to increase the imaging magnification of the LA-VCSEL on DMD_{b} by changing L_5 . Doing so would effectively provide higher resolution weights since every LA-VCSEL speckle would be significantly oversampled, allowing for the creation of more fractional weights using super-pixels. Another way would be to replace DMD_{b} with a liquid crystal on silicon spatial light modulator, which usually provide at least 8-bit resolution. Furthermore, the development and exploration of more advanced hardware-compatible learning algorithms, whether in the form of model-based or model-free black-box algorithms, could unlock the potential of this enhanced weight resolution, requiring further investigation.

“”

R3-Q4) In the current experimental setup, ternary weights are created sequentially, which could limit inference bandwidth and increase computational costs. The authors briefly mention the simultaneous creation of weights using optical copies of the LA-VCSEL image, but they do not provide details on how these optical copies are generated. Clarifying the experimental setup for

implementing this method, or presenting preliminary results to support the scheme shown in Figure 3, would strengthen the manuscript.

The reviewer raises a valid point here and we added a simple schematic to Figure 3 showing the output part of such a setup to address the reviewer's comment.

R3-Q5) The long-term stability is one of the promising results of this paper. However, it is unclear what factors contribute to the system's stability. The use of a multimode fiber and semiconductor laser in the setup may make the system susceptible to external environmental fluctuations, which typically cause drift. The authors should discuss the stability of their setup and whether the use of Boolean hardware (e.g., DMD) contributes to this stability.

The reviewer raises again a valid question which needs further explanation. The same question was asked by reviewer 1 so we refer to our answer to R1-Q4).

In addition to these concerns, I would like the authors to address the following points:

- The authors refer to their system as a reservoir computing (RC) system. However, since the LA-VCSEL is operated in its steady state during the experiment, it does not act as a recurrent neural network. This suggests that it may function more as an extreme learning machine, rather than a RC system. Could the authors comment on this point?

The reviewer is correct that strictly speaking our setup is more akin to an ELM than a reservoir because it is operated in its steady state, although there are aspects such as internal coupling between nodes that would differentiate further between our experimental ONN and an ELM scheme. Furthermore, this internal coupling matrix is dependent on optical injection. We added a paragraph in section 2.1 to address this point:

“”

Each frame on $\text{\textit{DMD}}_{\text{\textit{a}}}$ is displayed for $\sim \mu\text{s}$, which is orders of magnitude slower than the intrinsic time scales of the LA-VCSEL ($\sim 1\text{ns}$), meaning that we operate the LA-VCSEL and hence our ONN in its steady state. Consequently, we do not utilize the memory capacity of the LA-VCSEL, making our ONN functionally comparable to an extreme learning machine (ELM). However, certain characteristics, such as the internal coupling between nodes that is also dependent on the injected information, distinguish further between our experimental ONN and conventional ELM and RC schemes.

“”

- In Section 2.3, the authors mention that “ $\alpha = 0$ corresponds to the original algorithm.” However, “ $\alpha = 0$ ” implies n^{mirrors} is always zero in Eq. (2). Additionally, the authors should provide a citation for the “original algorithm.”

This comes from a mistake on our part in the equation : $n_{\text{mirrors}} = \lceil \alpha \cdot \text{NMSE}_k \rceil$, $\lceil \cdot \rceil$ refers to the ceiling function, we should have written $n_{\text{mirrors}} = \lfloor \alpha \cdot \text{NMSE}_k \rfloor + 1$, where $\lfloor \cdot \rfloor$ is the floor function, ensuring that at least 1 mirror is always flipped. We corrected this mistake in the manuscript.

- In Section 2.3, could the authors compare the convergence speed of the proposed in-situ algorithm with that of conventional simulated annealing?

The reviewer raises a valid point here and similar questions were asked by the other 2 reviewers, we refer therefore to our answer to reviewer 1 in R1-Q3)

- In Section 3.1, the authors introduce the "one-vs-all" classification method but do not provide an explanation of what it entails. A brief explanation of the "one-vs-all" classification approach would be helpful for readers unfamiliar with the method.

- The "one-vs-all" classification approach has limitations [Bishop 2006], especially in handling a large number of classes, as it can become computationally expensive. It would be better if the authors could discuss why they chose this approach and what advantages it offers in the context of their hardware implementation. Reference: [Bishop 2006] C. M. Bishop, Pattern Recognition and Machine Learning. New York: Springer, 2006.

We thank the reviewer for the comment. In our optical neural network, the scalar output is measured using a single photodetector, which makes the one-vs-all classification method a straightforward and compatible choice for our setup. While it is true that this approach may not be optimal for solving certain tasks, the main objective of our work is not to achieve the best possible performance on a specific task but rather to explore the feasibility of a hardware-compatible training algorithm. Even if one-vs-all classification is not ideal, it provides sufficient grounds for comparison, helping us to evaluate the challenges and potential of training directly within hardware constraints. To address the last two points made by the reviewer we added a paragraph in section 3.1:

“””

In a "one-vs-all" setting, for n_{classes} we train n_{classes} independent binary classifiers whose purpose is to distinguish one specific class out of all the others. This makes it generally ill-suited to problems with many classes \cite{bishop2006pattern}. In contrast, each individual binary classification problem is comparatively easier to solve than a single classification problem with n_{classes} , yet "one-vs-all" still provides sufficient ground for the comparison of different algorithms and their convergence behavior. For each comparison, all alternative approaches too were trained according to one-vs-all.

“””