

Nonlinear thermodynamic computing out of equilibrium

Corresponding Author: Dr Stephen Whitelam

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)

In this manuscript, the authors describe how a programmable thermodynamic computer could be constructed, using a network of thermodynamic neurons operating out of equilibrium. As with the neural networks implemented on ordinary computers, the neurons described here operate by performing nonlinear operations that convert inputs into outputs. However, while in standard NNs information flows in one direction, here it necessarily flows both forward and backward, as determined by the coupling between the neurons. The authors provide two proofs of concept: first, they train a 4x8 network of thermal neurons to approximate a cosine function; then, they train a 3x32 network to classify handwritten digits, using the MNIST data set.

This is a beautiful paper that presents a fascinating and original (to the best of my knowledge) idea. The overall concept is described clearly, and the proof-of-concept simulations illustrate its potential power.

I have a few minor comments regarding the presentation.

On page 1, right column, the authors write, "no design exists for a general-purpose programmable thermodynamic computer ...". In the minds of many readers, a "general-purpose programmable computer" means one based on boolean bits, which can be programmed to perform arbitrary calculations, simulations, etc. It would be useful to explain, briefly, why the ability to implement neural networks implies general-purpose programmable computing.

I'm not sure how to interpret Fig. 2a. The upper half shows two circles (neurons) with a double arrow pointing from one to the other; the lower half shows two circles connected by a line, with a double arrow to the right. Presumably both depict the same two neurons. Why are there two depictions, and what do the double arrows signify in each?

In Eq 7, I presume the subscript "r" on the left stands for "realizations". What about the subscripts "r,s" on the right side of Eq 9? Do these stand for "realizations, samples"? The sentence containing Eq 7 mentions "M independent realizations" while the one containing Eq 9 mentions "M=10³ samples". The distinction between realizations and samples should be made precise.

The text describing Figs 3b and 3c (both in the caption and in the main text) uses the term "class prediction" in two somewhat different ways. In Fig 3b, the class prediction takes on values roughly between -10 and +10, with a different histogram for each class. In Fig 3c, the term class prediction refers to a probability value, with a different value for each class. The difference is explained clearly in the SI (S16, S17). For the reader who doesn't dig into the SI, the use of "class prediction" in the main text might be confusing. A brief clarification would be useful.

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)

This work introduces an approach to doing thermodynamic computing. Specifically they introduce an abstract building block that has a quartic potential energy function, and they perform various simulations with this building block.

Practicality issues:

This is a theory paper. The results presented in this paper are numerical simulations, not actual experiments on a real thermodynamic device. This is an important point, as it side-steps questions of the practicality of building the proposed device. Thus, all of the results of the paper assume the practicality of building the device, without actually investigating the practicality. For example, I did not see a detailed circuit diagram of the proposed neuron anywhere in the paper. Both figure 1 and figure 2 just show a cartoon picture with circles, which is quite different than a circuit diagram. This limits the impact of the paper, as there is no concrete proposal of how to actually build the neuron with a circuit. Moreover, going beyond a single neuron, one would have to consider the practicality of how to connect the neurons and lay out the chip. What would these connections look like in practice?

Main result:

Given that this is a theory paper, it is not quite clear what the main theoretical result is. My guess is that the main result is that, with a quartic potential neuron, one can build a thermodynamic neural network that is capable of performing classification tasks like MNIST. This is a useful finding, but I believe it doesn't rise to the broad impact level of this journal, so I would recommend a more specialized journal for this paper.

Title of the paper:

The current title of the paper emphasizes thermodynamic computing out of equilibrium. However, upon reading the paper, I find that most of the contribution of the paper is about constructing a thermodynamic neural network with a non-linear building block. Thus, I think the title should be changed to something like the following: "Thermodynamic neural network using quartic potentials". This emphasizes the non-linear aspect of the contribution. It's important for the title to match the actual content of the text.

To justify the emphasis on being out-of-equilibrium, I would expect a much more detailed theoretical analysis of non-equilibrium thermodynamics, such as fluctuation theorems or other key theoretical ideas in non-equilibrium thermodynamics. However, the paper did not have any mention of fluctuation theorems or other deep theoretical concepts in non-equilibrium thermodynamics. So I would prefer the title of the paper to focus on the non-linearity, i.e., the thermodynamic neural network.

Relation to other work:

The work shares some similarities with a paper by Lipka-Bartosik et al.: Thermodynamic computing via autonomous quantum thermal machines
<https://www.science.org/doi/full/10.1126/sciadv.adm8792>

In the Lipka-Bartosik paper, they introduce a thermodynamic neuron, just like in the paper under review, and they show that a network of these thermodynamic neurons can be a universal approximator. Although the formulations of the neurons may be different in the two papers, it nevertheless has similar concepts. So it is worth adding a citation to the Lipka-Bartosik paper, and also a brief discussion of how the current work is different from that paper.

In addition, there is an experimental paper by Dillavou et al. that performs machine learning tasks (similar to the current paper under review) by using a nonlinear analog circuit network: <https://www.pnas.org/doi/abs/10.1073/pnas.2319718121>

So the authors can cite that paper and discuss overlapping results.

Overall recommendation:

The paper does make a nice observation that MNIST classification is possible with simple quartic potential neurons. However, for several reasons, I believe it doesn't rise to the level of the journal. These reasons include: (1) it is a pure theory paper with no experiments on real hardware, (2) it doesn't discuss the practicality of building this hardware with circuits, (3) there is some overlap with prior work that wasn't discussed, (4) there is not much theory provided on non-equilibrium thermodynamics making the title choice questionable.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)

This paper presents an idea for implementing a (function like a) logistic function in a system undergoing thermodynamic relaxation. In a nutshell, the idea is to use a quartic potential, where the input to that function is given by the coefficient of the linear term in that potential, the parameters of the neuron are given by the coefficients of the other terms, and the output is given by the value of the resultant equilibrium state of the system. They then point out that by coupling such systems, one can implement an arbitrary feedforward neural net. They end with some toy experiments involving NIST, to show that this system can in fact implement such a neural net.

This will be of some interest to some researchers, especially those few who are working on "thermodynamic computing".

However, there are some major flaws in the paper. Most importantly, *this model of the authors is simply a slight variant of Hopfield nets, and in particular of what are called "dense associative memories"*. That body of work is *very* widely known - some of it has been done by a Nobel laureate! Moreover, dense associative memories in particular have been related to both real neurons in the brain, and to the transformers used in LLMs, which makes the connection to the model of the authors even more pronounced.

This fact that their work is only a slight variant of a well-established body of work, one that the authors are unaware of, means that this paper is only of minor significance. It should not appear in Nature Communications. Someplace like PRE would be more appropriate..

Detailed concerns are below:

- 1) Given how close the connection is between the model of the authors and dense associative memories, it is imperative that the authors discuss it at length. Yet they don't even mention Hopfield nets.
- 2) The justification of "thermodynamic computing" (like the systems the authors use), if there is any, is that they reduce energy cost. But there is no analysis of this issue in the paper.
- 3) Toy examples involving an 8 x 4 neural net won't convince anyone that their system can be scaled up.
- 4) There are many other literatures that the authors are unaware of. Most obviously, there is no mention whatsoever of stochastic thermodynamics, only a few references that are never discussed. This is despite the fact that there is no way to investigate (2) without using stochastic thermodynamics. In particular, there is no mention of the stochastic thermodynamics of computation, besides an old reference to a paper by Wolpert. (The authors must (!) read and discuss <https://www.pnas.org/doi/10.1073/pnas.2321112121>)
- 5) In that regard, please note that the work by Todd Hynton, Mike Frank, etc., is not well-regarded by many in the community. See in particular <https://iopscience.iop.org/article/10.1088/1742-5468/2014/03/P03025> , for a simple proof of the irrelevance of reversible computation.
- 6) The meaning of what's being shown in Fig. 1, described just below Eq. 1, should be in the caption of that figure.
- 7) There is a cryptic mention of "thermodynamically stable", which is not a standard term in the literature. (Do the authors mean to say a meta-stable state?)

(Remarks on code availability)

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

The author has carefully addressed the issues that I raised in my report. In my view the manuscript is acceptable for publication in Nature Communications.

(Remarks on code availability)

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/>

Reviewer #1 (Remarks to the Author):

In this manuscript, the authors describe how a programmable thermodynamic computer could be constructed, using a network of thermodynamic neurons operating out of equilibrium. As with the neural networks implemented on ordinary computers, the neurons described here operate by performing nonlinear operations that convert inputs into outputs. However, while in standard NNs information flows in one direction, here it necessarily flows both forward and backward, as determined by the coupling between the neurons. The authors provide two proofs of concept: first, they train a 4x8 network of thermal neurons to approximate a cosine function; then, they train a 3x32 network to classify handwritten digits, using the MNIST data set.

This is a beautiful paper that presents a fascinating and original (to the best of my knowledge) idea. The overall concept is described clearly, and the proof-of-concept simulations illustrate its potential power.

We thank the referee for their comments.

I have a few minor comments regarding the presentation.

On page 1, right column, the authors write, "no design exists for a general-purpose programmable thermodynamic computer ...". In the minds of many readers, a "general-purpose programmable computer" means one based on boolean bits, which can be programmed to perform arbitrary calculations, simulations, etc. It would be useful to explain, briefly, why the ability to implement neural networks implies general-purpose programmable computing.

To explain this idea we now write "However, no design exists for \new{a thermodynamic computer that operates at specified observation times and is programmable in a general sense, meaning that it is capable of approximating arbitrary continuous functions. Here we provide such a design by introducing a thermodynamic computer that is analogous to a neural network.} A thermodynamic computer of this nature is a nonlinear model that can serve as a universal function approximator, and can be programmed to perform arbitrary nonlinear computations at specified observation times. This is true whether or not the computer has attained thermodynamic equilibrium at those observation times."

I'm not sure how to interpret Fig. 2a. The upper half shows two circles (neurons) with a double arrow pointing from one to the other; the lower half shows two circles connected by a line, with a double arrow to the right. Presumably both depict the same two neurons. Why are there two depictions, and what do the double arrows signify in each?

In the caption of that figure we now write "{a,b) Elements of a thermodynamic computer analogous to a neural network. (a) The thermodynamic neurons described in~\f{fig1} are connected by bilinear couplings. \new{Top: a single circle implies a neuron x_i of the type described in \f{fig1}, with an input (bias) $I=b_i$ and a nonlinear potential parameterized by $\bm{J}=(J_1,J_2,J_3)$. Bottom:

lines between circles imply a bilinear coupling $J_{ij} x_i x_j$. (b) With the visual shorthand described in panel (a), we consider layered networks of such neurons...

In Eq 7, I presume the subscript "r" on the left stands for "realizations". What about the subscripts "r,s" on the right side of Eq 9? Do these stand for "realizations, samples"?

We now write "where the subscript 'r' indicates 'reset-sampling', and the sum runs over..."

The sentence containing Eq 7 mentions "M independent realizations" while the one containing Eq 9 mentions "M=10³ samples". The distinction between realizations and samples should be made precise.

We changed "M independent realizations" to "M independent trajectories", and now write "The subscript 'r,s' indicates that averages are taken either in reset-sampling or in serial-sampling mode. If the former, each of the M samples is obtained from an independent trajectory. If the latter, all M samples are obtained from a single trajectory."

The text describing Figs 3b and 3c (both in the caption and in the main text) uses the term "class prediction" in two somewhat different ways. In Fig 3b, the class prediction takes on values roughly between -10 and +10, with a different histogram for each class. In Fig 3c, the term class prediction refers to a probability value, with a different value for each class. The difference is explained clearly in the SI (S16, S17). For the reader who doesn't dig into the SI, the use of "class prediction" in the main text might be confusing. A brief clarification would be useful.

Thank you for noting this inconsistency. We now write: In Fig 3(b) we show the output of the trained computer when presented with a single digit, an 8, which it correctly classifies. We distinguish between the class score $Y^{(C)}$, the raw output of the thermodynamic computer prior to normalization, and the class probability $p^{(C)}$, which results from applying a softmax transformation to averaged score. We use the term class prediction to refer to the class with the highest score, i.e., the computer's predicted label for the digit. We plot the probability distribution, taken over 10^5 samples, of the computer's class score $Y^{(C)}$ for the digit. The mean value of each distribution (the value used for classification) is indicated at the top of the panel. The correct distribution is shown in green, with the others in shades of blue. In panel (c) we plot the value of the trained computer's 10 class probabilities upon being shown the indicated digit. The computer is trained to classify the digit at an observation time $t_f=1$. In this case the computer has attained a steady-state dynamics at the specified observation time, but this is not a general phenomenon: when presented with other digits, the computer's neurons are still evolving at $t_f=1$, and so in general the computer operates out of equilibrium.

Reviewer #2 (Remarks to the Author):

This work introduces an approach to doing thermodynamic computing. Specifically they introduce an abstract building block that has a quartic potential energy function, and they perform various simulations with this building block.

Practicality issues:

This is a theory paper. The results presented in this paper are numerical simulations, not actual experiments on a real thermodynamic device. This is an important point, as it side-steps questions of the practicality of building the proposed device. Thus, all of the results of the paper assume the practicality of building the device, without actually investigating the practicality. For example, I did not see a detailed circuit diagram of the proposed neuron anywhere in the paper. Both figure 1 and figure 2 just show a cartoon picture with circles, which is quite different than a circuit diagram. This limits the impact of the paper, as there is no concrete proposal of how to actually build the neuron with a circuit. Moreover, going beyond a single neuron, one would have to consider the practicality of how to connect the neurons and lay out the chip. What would these connections look like in practice?

We thank the referee for their review. This paper is indeed computational, intended to show that it is possible in principle to operate a thermodynamic computer out of equilibrium. Showing this capability in simulation is an essential first step, as it motivates future experimental exploration, guides hardware implementation efforts, and provides a controlled environment for systematically testing different architectures and optimization strategies.

However, our design is directly motivated by a circuit-level equilibrium thermodynamic computer, that was built in Ref [12], summarized Fig. 1 of Ref. [12]. In our model, the dots stand for the dots (the small electrical circuits) that figure, with the inductor replaced by a nonlinear inductor. Connections between the dots could be formed using the same capacitive coupling scheme as Fig. 1 of [12]. To make this point clearer, we now write: “Our proposed design, sketched in Fig. 2, could be implemented using existing hardware elements: the neurons could be realized by RLC [12] circuits with nonlinear components, or by superconducting circuits with Josephson junctions [24].” and “We use the bilinear interaction of Refs. [12,13].”

We still use a schematic representation, however, as there are multiple ways of implementing these neurons as described in the main text.

Main result:

Given that this is a theory paper, it is not quite clear what the main theoretical result is. My guess is that the main result is that, with a quartic potential neuron, one can build a thermodynamic neural network that is capable of performing classification tasks like MNIST. This is a useful finding, but I believe it doesn't rise to the broad impact level of this journal, so I would recommend a more specialized journal for this paper.

The main results of this paper are: 1) that it is possible to do programmable nonlinear algebra with a thermodynamic computer, whereas existing thermodynamic computers [11,12] perform only linear algebra; and 2) that these computations can be done “on a clock”—that is, at a specified observation time. In equilibrium models, the system must be allowed to relax fully, but as noted in Ref. [17], even small thermodynamic systems can exhibit widely varying equilibration times, complicating practical computation. Our approach avoids this issue by working far from equilibrium: we train the system to return the correct result at a fixed time. This is nontrivial, since away from equilibrium, there is no simple guiding principle for how to shape the system’s dynamics. We address this using genetic algorithms, and we believe this general strategy represents a significant advance for thermodynamic computing.

To clarify this contribution we now write, in the introduction, “The equilibrium thermodynamic computers of Refs. [11, 12], which motivate this work, can be viewed as continuous-spin analogs of Hopfield networks [19] or Boltzmann machines [20, 21], which are statistical mechanical models that represent probability distributions over binary variables. Since a Boltzmann machine encodes information in its equilibrium or Boltzmann distribution, one could refer to the models of Refs. [11, 12] as *Boltzmann computers*: they are designed to sample from the Boltzmann equilibrium corresponding to the system’s potential energy function. In such models, the specific form of the dynamics, whether overdamped or underdamped, is less important than the fact that it is microscopically reversible and converges to the desired equilibrium.

Our contribution is to explore the computational capabilities of such systems when extended to include nonlinear neuron potentials and operated out of equilibrium. Specifically, we encode the outcome of a computation in the transient dynamical trajectories of the system, without appealing to the Boltzmann distribution. A computer operating in this mode, using Langevin trajectories to perform a computation, could be termed a *Langevin computer*.”

Title of the paper:

The current title of the paper emphasizes thermodynamic computing out of equilibrium. However, upon reading the paper, I find that most of the contribution of the paper is about constructing a thermodynamic neural network with a non-linear building block. Thus, I think the title should be changed to something like the following: “Thermodynamic neural network using quartic potentials”. This emphasizes the non-linear aspect of the contribution. It’s important for the title to match the actual content of the text.

To justify the emphasis on being out-of-equilibrium, I would expect a much more detailed theoretical analysis of non-equilibrium thermodynamics, such as fluctuation theorems or other key theoretical ideas in non-equilibrium thermodynamics. However, the paper did not have any mention of fluctuation theorems or other deep theoretical concepts in non-equilibrium thermodynamics. So I would prefer the title of the paper to focus on the non-linearity, i.e., the thermodynamic neural network.

We have modified the title to “Nonlinear thermodynamic computing out of equilibrium,” which reflects both core contributions of the work. Our computations are performed out of equilibrium: we make no appeal to the Boltzmann distribution, but instead fix an observation time and train the system to match a target result at that time. As shown in Fig. 2(g), the computer produces the desired output

only at the designated time. The loss function shown in Fig. S8(e) further illustrates this point. Because the system is not stationary at that time, it is by definition out of equilibrium (one-time quantities are stationary in thermal equilibrium).

We agree that fluctuation theorems are a rich direction for future exploration: for instance, we recently showed that it is possible to train a generative version of a thermodynamic computer by gradient descent, with the learning rule equivalent (as shown by a fluctuation relation) to minimizing the heat emission of the trained computer [S. Whitelam, Generative thermodynamic computing, arXiv:2506.15121 (2025)]. In addition, existing thermodynamic computers run with added noise, not just thermal noise, and it is possible to analyze the energy requirements of this addition within the framework of stochastic thermodynamics [S Whitelam, Increasing the clock speed of a thermodynamic computer by adding noise, arXiv:2410.12211 (v3, 2025)]. However, the focus of the present paper is to enact a particular type of calculation, and since this calculation happens in general out of equilibrium, we believe the title to be appropriate.

The work shares some similarities with a paper by Lipka-Bartosik et al.: *Thermodynamic computing via autonomous quantum thermal machines*

<https://www.science.org/doi/full/10.1126/sciadv.adm8792>

In the Lipka-Bartosik paper, they introduce a thermodynamic neuron, just like in the paper under review, and they show that a network of these thermodynamic neurons can be a universal approximator. Although the formulations of the neurons may be different in the two papers, it nevertheless has similar concepts. So it is worth adding a citation to the Lipka-Bartosik paper, and also a brief discussion of how the current work is different from that paper.

Agreed, and we did reference the paper by Lipka-Bartosik et al. in our original submission (Ref. [11] in the Supplement). We have now moved the citation to the main text, where we write:

“A recent paper¹ presented the design for a thermodynamic neuron realized by qubits coupled to multiple thermal baths. The design presented here is based on a fully classical model of a nonlinear thermodynamic neuron, and requires only a single thermal bath.” That is, the design of Lipka et al. requires quantum elements in contact with three separate thermal baths to realize. Our design can be realized at room temperature by existing classical hardware. Lipka et al. also present only a design for a single neuron, whereas we show how to program a coupled network of nonlinear neurons to do nontrivial nonlinear calculations at specified observation times (i.e. out of equilibrium in general).

In addition, there is an experimental paper by Dillavou et al. that performs machine learning tasks (similar to the current paper under review) by using a nonlinear analog circuit network:

<https://www.pnas.org/doi/abs/10.1073/pnas.2319718121>

So the authors can cite that paper and discuss overlapping results.

Thank you for noting this connection. This paper describes a physical neural network trained to do a nonlinear task, and so in that respect there is a similarity. However, the physical neural network described in that paper does classical analog computation, dissipating of order 10 pJ (or about 10^9

kT) per fundamental unit. This is very far from the fluctuation dominated, 1-100 kT range of thermodynamic computing, and so in that respect this work is different. We have added to the paper: “In a related vein, nonlinear physical neural networks that operate on the energy scales of classical computing have recently been trained to do nonlinear computations [25].”

Finally, while the nonlinear tasks demonstrated in the Lipka and Dillavou papers (e.g., XOR classification) are useful benchmarks, they are intentionally minimal. By contrast, MNIST classification is a standard and more challenging ML task, involving 784 inputs and 10 output classes. Our demonstration that a thermodynamic computer can be trained to classify MNIST with accuracy comparable to classical neural networks represents, in our view, a meaningful advance in scope and ambition.

Overall recommendation:

The paper does make a nice observation that MNIST classification is possible with simple quartic potential neurons. However, for several reasons, I believe it doesn't rise to the level of the journal. These reasons include: (1) it is a pure theory paper with no experiments on real hardware, (2) it doesn't discuss the practicality of building this hardware with circuits, (3) there is some overlap with prior work that wasn't discussed, (4) there is not much theory provided on non-equilibrium thermodynamics making the title choice questionable.

We appreciate the reviewer's concerns and their summary of key limitations. We agree that this is a theory paper, and that it does not present experiments on hardware. We wish to emphasize, however, that this serves as a valuable demonstration of this concept, prompting further experimental investigation and allowing for testing of different architectures and optimization methods. Moreover, the proposed design is directly motivated by the design of Fig. 1 of Ref. [12], which has been built in the laboratory, and could be implemented by replacing the inductor with a nonlinear circuit element. We have clarified this connection in the manuscript. Regarding the Lipka paper, which we now cite in the main text, it involves quantum elements connected to three thermal baths to realize a single neuron. The Dillavou paper describes a classical analog neural network, not a thermodynamic computer, and operates at energy scales many orders of magnitude above the thermal regime. By contrast, our design is classical, fluctuation-driven, and to our knowledge represents the first proposal for a nonequilibrium thermodynamic computer that could be realized using standard circuit elements. We have also changed the title of the paper to emphasize the nonlinearity of our computing, but would remark that, throughout we have shown (e.g. Fig. 2(g), Fig. S8) that we are clearly operating in a non-equilibrium regime, which is crucial for performing computations on a clock.

Reviewer #3 (Remarks to the Author):

This paper presents an idea for implementing a (function like a) logistic function in a system undergoing thermodynamic relaxation. In a nutshell, the idea is to use a quartic potential, where the input to that function is given by the coefficient of the linear term in that potential, the parameters of the neuron are given by the coefficients of the other terms, and the output is given by the value of the resultant equilibrium state of the system.

They then point out that by coupling such systems, one can implement an arbitrary feedforward neural net. They end with some toy experiments involving NIST, to show that this system can in fact implement such a neural net.

We are grateful to the reviewer for their thoughtful and constructive feedback.

The reviewer accurately summarizes one of our core ideas: we propose a method for implementing nonlinear response functions—analogue to logistic activation functions—within thermodynamic systems governed by quartic potentials. In our framework, the input to each thermodynamic neuron corresponds to the linear coefficient of the potential, while the other coefficients define the neuron's intrinsic properties. Crucially, however, the neuron's output is not determined by its equilibrium state, but by its state at a user-specified observation time (these can be the same, if the observation time is large, but are in general different). This design enables computation to occur “on a clock,” without requiring the system to reach thermal equilibrium—a process that can be prohibitively slow, even for physically small devices.

By coupling such neurons, we construct networks that are functionally equivalent to feedforward neural nets. The sine-wave task serves as a minimal example to demonstrate nonlinearity; it is conceptually similar to the XOR benchmark often used to test nonlinear models, including those discussed by Reviewer #2. However, we also demonstrate that our system can solve the MNIST classification problem, an established benchmark in machine learning. While MNIST is not the most demanding dataset by current standards, it represents a significant step beyond XOR in both scale and complexity. Successfully learning MNIST confirms that thermodynamic computers, even when operating far from equilibrium, can perform meaningful and nontrivial machine-learning tasks.

This will be of some interest to some researchers, especially those few who are working on "thermodynamic computing". However, there are some major flaws in the paper. Most importantly, *this model of the authors is simply a slight variant of Hopfield nets, and in particular of what are called "dense associative memories"*. That body of work is *very* widely known - some of it has been done by a Nobel laureate! Moreover, dense associative memories in particular have been related to both real neurons in the brain, and to the transformers used in LLMs, which makes the connection to the model of the authors even more pronounced.

This fact that their work is only a slight variant of a well-established body of work, one that the authors are unaware of, means that this paper is only of minor significance. It should not appear in Nature Communications. Someplace like PRE would be more appropriate..

We appreciate the connection to Hopfield networks and, in particular, dense associative memories (DAMs), and we acknowledge the architectural similarity between these models and ours, notably the use of coupled scalar degrees of freedom. However, our model departs from these classical neural-network frameworks in several important ways, both in its physical underpinnings and in its computational role.

First, our system is a thermodynamic computer governed by stochastic Langevin dynamics, designed to operate out of equilibrium. Thermal fluctuations are not treated as noise to be suppressed or averaged out, but as an active computational resource. In contrast, Hopfield networks and DAMs typically operate within deterministic or quasi-equilibrium frameworks, where dynamics are used to locate attractor states in an energy landscape defined by pairwise couplings. These attractors encode stored patterns, and computation consists of recalling or retrieving them. Our model, by contrast, is not designed for memory retrieval but for real-time nonlinear function evaluation at specified observation times.

Second, the nonlinearity in our model is implemented at the level of each neuron via non-quadratic (e.g., quartic) confining potentials, resulting in a nonlinear input-output relationship at finite times. This feature allows networks of such neurons to act as universal function approximators, even while operating far from equilibrium. In DAMs and conventional Hopfield networks, the energy functions are typically quadratic, and any nonlinearity must be introduced post hoc (e.g., through thresholding or activation functions) rather than arising naturally from the system's thermodynamics.

Third, our networks are explicitly trained using a genetic algorithm to perform general-purpose computation, such as function regression and classification, rather than content-addressable memory. For example, we demonstrate that our system can learn to approximate smooth nonlinear functions and to classify handwritten digits from the MNIST dataset. These are tasks that require a degree of programmability, generalization, and input sensitivity not addressed by traditional associative memory models.

Finally, our design explicitly embraces time-dependent, nonequilibrium dynamics: computations are defined not by fixed-point convergence but by the system's state at a pre-specified observation time. This "on a clock" behavior is essential for practical applications, particularly when equilibration times are long or ill-defined, as is often the case in thermodynamic systems. In Hopfield-type models, computation generally requires convergence to equilibrium, which may not be feasible or desirable in real physical systems.

Taken together, these differences in physical dynamics, neuron design, computational purpose, and training paradigm mark a substantive departure from associative memory models. We build upon ideas from statistical mechanics and neural computation to describe a nonequilibrium computing paradigm that uses fluctuations and dynamics as tools for general nonlinear computation.

To clarify these differences, we now write

"The equilibrium thermodynamic computers of Refs. [11, 12], which motivate this work, can be viewed as continuous-spin analogs of Hopfield networks [19] or Boltzmann machines [20, 21], which are statistical mechanical models that represent probability distributions over binary variables. Since a Boltzmann machine encodes information in its equilibrium or Boltzmann distribution, one could

refer to the models of Refs. [11, 12] as *Boltzmann computers*: they are designed to sample from the Boltzmann equilibrium corresponding to the system's potential energy function. In such models, the specific form of the dynamics, whether overdamped or underdamped, is less important than the fact that it is microscopically reversible and converges to the desired equilibrium.

Our contribution is to explore the computational capabilities of such systems when extended to include nonlinear neuron potentials and operated out of equilibrium. Specifically, we encode the outcome of a computation in the transient dynamical trajectories of the system, without appealing to the Boltzmann distribution. A computer operating in this mode, using Langevin trajectories to perform a computation, could be termed a *Langevin computer*."

In our view, these differences justify publication in a journal of broad interest, given the growing interdisciplinary interest in physically realizable, energy-efficient computing paradigms. Our approach is a nonequilibrium variant of the equilibrium thermodynamic computers of Refs. [11,12], which do appear in Nature Communications (e.g. Melanson, D., Abu Khater, M., Aifer, M. et al.

Thermodynamic computing system for AI applications. Nature Communications 16, 3757 (2025).) and journals of comparable impact factor.

Detailed concerns are below:

1) Given how close the connection is between the model of the authors and dense associative memories, it is imperative that the authors discuss it at length. Yet they don't even mention Hopfield nets.

We thank the reviewer for this important observation. In the revised manuscript, we now explicitly state that the equilibrium thermodynamic computers of Refs. [11, 12], which motivate this work, can be viewed as continuous-spin analogs of Hopfield networks used out of equilibrium. Our contribution is to explore the computational capabilities of such systems when extended to include nonlinear neuron potentials and operated out of equilibrium. This out-of-equilibrium regime, together with the nonlinear dynamics, allows our model to serve as a universal function approximator, a capability that goes beyond the attractor-based behavior of conventional Hopfield nets and dense associative memories.

2) The justification of "thermodynamic computing" (like the systems the authors use), if there is any, is that they reduce energy cost. But there is no analysis of this issue in the paper.

Agreed, energy efficiency is one of the key motivations for thermodynamic computing. In the paper we state this motivation, but the paper's focus is to demonstrate proof of principle for computation out of equilibrium, rather than to analyze energy cost. It is possible to analyze the energy requirements of this addition within the framework of stochastic thermodynamics [S Whitelam, Increasing the clock speed of a thermodynamic computer by adding noise, arXiv:2410.12211 (v3, 2025)]. However, the focus of the present paper is to enact a particular type of calculation, and since this calculation happens in general out of equilibrium we believe the title to be appropriate. In our answer to point 4), below we expand on the issue of energy cost.

3) Toy examples involving an 8×4 neural net won't convince anyone that their system can be scaled up.

We appreciate the reviewer's concern about scalability. The 8×4 network used to approximate a sine function was chosen for clarity, as it provides the simplest example we could think of of nonlinear function approximation. To move beyond toy problems, we have included in the same paper a classification task using the MNIST dataset, involving a three-layer network with 32 neurons per layer and 784 input pixels. While this architecture is modest in size, by the standards of the largest classical neural networks, it is comparable to the classical neural nets of the early 2000s, and achieves similar classification accuracy. More importantly, we know from the universal approximation theorems that networks of nonlinear units can represent any smooth function, given sufficient hidden nodes, and so the demonstration of an MNIST-like problem will show others that thermodynamic computing can do standard ML tasks. We also note that the optimization method we use here has previously been applied to neural networks with $> 10^7$ parameters [Stephen Whitelam, Viktor Selin, Ian Benlolo, Corneel Casert & Isaac Tamblyn, *Training neural networks using Metropolis Monte Carlo and an adaptive variant*, *Machine Learning: Science and Technology* **3**(4), 045026 (2022), doi:10.1088/2632-2153/aca6cd] and so can be used to scale up to large nonequilibrium thermodynamic computers.

4) There are many other literatures that the authors are unaware of. Most obviously, there is no mention whatsoever of stochastic thermodynamics, only a few references that are never discussed. This is despite the fact that there is no way to investigate (2) without using stochastic thermodynamics. In particular, there is no mention of the stochastic thermodynamics of computation, besides an old reference to a paper by Wolpert. (The authors must (!) read and discuss <https://www.pnas.org/doi/10.1073/pnas.2321112121>)

We thank the reviewer for this important recommendation. We now mention that paper, and the review by Siefert, and note three examples in which these tools were used to train or analyze a thermodynamic computer: "The theoretical framework describing thermodynamic computing includes nonequilibrium statistical mechanics, stochastic thermodynamics, and information theory. These fields offer a set of tools for analyzing small, fluctuating systems, such as fluctuation theorems, thermodynamic speed limits, and uncertainty relations [22–24]. For example, the Jarzynski equality, combined with nonequilibrium work measurements, can be used to compute the determinant of a matrix [11]. Thermodynamic computers often use added noise to accelerate computation, and the energetic cost of this addition can be quantified within stochastic thermodynamics [17]. Learning rules can also be linked to thermodynamic quantities: for instance, a generative thermodynamic computer trained by gradient descent minimizes heat dissipation during training [24]."

5) In that regard, please note that the work by Todd Hylton, Mike Frank, etc., is not well-regarded by many in the community. See in particular <https://iopscience.iop.org/article/10.1088/1742-5468/2014/03/P03025>, for a simple proof of the irrelevance of reversible computation.

We appreciate the reviewer's perspective, and have added a citation to the paper mentioned [J. Stat. Mech. (2014) P03025], in our reply to point 4).

6) The meaning of what's being shown in Fig. 1, described just below Eq. 1, should be in the caption of that figure.

We agree, and we have now moved the relevant explanatory material from the main text into the caption of Fig. 1 so that the figure is self-contained and easier for readers to interpret:

The circle represents a scalar degree of freedom x . The curved line represents its intrinsic energy, the terms in $J = (J_2, J_3, J_4)$, $V(x)$ in Eq. (1). The straight line represents an input signal or bias, the term in I in Eq. (1).

7) There is a cryptic mention of “thermodynamically stable”, which is not a standard term in the literature. (Do the authors mean to say a meta-stable state?)

Thank you for highlighting this ambiguity. In the revised manuscript, we now clarify that by “thermodynamically stable” we mean that the confining potential is bounded from below, which ensures that the system has a well-defined partition function.