TRAINING A CONVERGENT ENERGY TRANSFORMER WITH EQUILIBRIUM PROPAGATION

Rasmus Høier
Rain AI

Tugdual Kerjan
Rain AI

Benjamin Scellier
Rain AI

ABSTRACT

Equilibrium Propagation (EP) is a learning framework for energy-based models, i.e. models whose dynamics evolve toward minima (or more generally critical points) of an energy functional. Because it relies on equilibration dynamics and local learning rules, EP is well suited to computing platforms based on analog physics, which may offer substantial energy-efficiency gains. Although standard Transformers are not usually viewed or framed as energy-based models, the recently introduced the Energy Transformer (ET) (Hoover et al., 2024) implements transformer-like computations through dynamics minimizing a global energy function. In its original form, however, the ET is not directly compatible with EP, because it is designed to perform only a small number of energy-minimization steps rather than to converge to equilibrium. We therefore develop a convergent variant, the Convergent Energy Transformer (CET), and train it with EP on a masked image completion task. This work takes a step toward physically inspired, hardware-friendly training methods for transformer-like models.

1 INTRODUCTION

Recent progress in machine learning has been driven in large part by the Transformer architecture (Vaswani et al., 2017), whose attention mechanism enables to model long-range interactions in sequences, images, and other structured data. Transformers now underpin many of the most influential AI systems, with applications to language (Achiam et al., 2023), vision (Kirillov et al., 2023), speech (Baevski et al., 2020), video (Kondratyuk et al., 2023), as well as biological structure prediction systems (Jumper et al., 2021). As these models have grown in scale, their empirical performance has improved, but so too have the computational and energy costs of training and deployment (Cottier et al., 2024). State-of-the-art transformer systems therefore require increasingly large accelerator clusters and data-center infrastructure, which raises both economic and environmental concerns and motivates the search for more energy-efficient computing and learning paradigms.

Analog neuromorphic hardware accelerators promise to reduce power consumption of computing by exploiting the physical dynamics and massive parallelism naturally present in many neural-network-like physical systems - such as optical or analog electrical devices (Marković et al., 2020). Proof-of-concept analog optical accelerators have been proposed to carry out the bulk of matrix computations of Transformers in the optical domain during inference (Anderson et al., 2023). To realize more energy gains, however, training also needs to happen in the analog domain. Yet, traditional back-propagation relies on precise digital arithmetic, which is not well suited to noisy analog substrates.

In recent years, a number of "physical learning" algorithms have been developed for training such physical neural-network-like systems, by harnessing physical dynamics, local measurements and local update rules (Momeni et al., 2024). One such approach is Equilibrium Propagation (EP) (Scellier & Bengio, 2017), which applies in energy-based systems, i.e. systems driven by energy or Lyapunov dynamics and converging to equilibrium. EP computes the gradients of a user-chosen cost function by comparing two equilibrium states of the same dynamical system under different boundary conditions, providing local parameter updates. EP has been successfully applied in a variety of systems, including continuous Hopfield networks (Scellier & Bengio, 2017), nonlinear resistive networks (Kendall et al., 2020) and oscillatory neural networks (Zoppo et al., 2022; Wang et al., 2024; Rageau & Grollier, 2025). Experimental demonstrations have been conducted in resistor networks (Dillavou et al., 2022), memristor crossbars (Yi et al., 2023; Oh et al., 2023), and D-wave's Ising

machine (Laydevant et al., 2024). Beyond energy-based models, EP has been extended in a number of directions, including Lagrangian systems (Kendall, 2021; Scellier, 2021; Massar, 2025; Pourcel et al., 2025), quantum systems (Massar & Mognetti, 2024; Wanjura & Marquardt, 2024; Scellier, 2024), holomorphic models (Laborieux & Zenke, 2022) and bilevel optimization problems (Zucchet & Sacramento, 2022).

Although standard Transformers are not usually formulated as energy-based models, recent work on modern Hopfield networks has shown how attention-like updates can be derived from an underlying energy function (Ramsauer et al., 2020). Building on this, the Energy Transformer (ET) (Hoover et al., 2024) reformulates transformer-like computation - including attention, multi-layer perceptrons (MLPs), layer normalization, and residual connections - as iterative minimization of a global energy. Because the same energy function is used at every step, the resulting dynamics can be viewed as a depth-unrolled architecture with shared parameters across steps. In numerical experiments, Hoover et al. (2024) report performance competitive with conventional transformers, despite having substantially fewer parameters. Originally used for image reconstruction tasks Hoover et al. (2024), the ET has also been employed in hydrodynamic problems (Zhang et al., 2025) and has inspired GPT-like energy-based language modeling (Dehmamy et al., 2025). An implementation of the ET (and other dense associative memories) in analog electrical circuits has also been proposed (Bacvanski et al., 2025).

In its original formulation, however, the ET is not directly compatible with EP. To obtain accurate parameter gradients, EP requires the network state to relax to an equilibrium of the energy. By contrast, the original ET was designed to operate in a transient regime: in practice, performance is best when the energy minimization is stopped after only a small number of steps, well before convergence. Notably, in their image reconstruction experiments, Hoover et al. (2023) report that the highest-quality images were obtained using 12 steps of gradient descent on the ET's energy with a step-size of 0.1, while Figure 8 and the accompanying video found in their supplemental material[1] show that the energy continues to decrease beyond this point. Importantly, further minimization significantly degrades the quality of images. To resolve this incompatibility, we modify the ET's architecture by treating the partially masked input image as a fixed boundary condition throughout the relaxation dynamics. This modification allows the network to converge to an equilibrium (minimum or critical point of the energy), and thereby makes the model amenable to training with EP. We then train our model, which we call the the Convergent Energy Transformer (CET) using EP to perform masked image completion on the CELEBA-faces dataset Liu et al. (2015), and demonstrate comparable performance to the baseline obtained using backpropagation through the equilibration trajectory.

Our work relates to recent efforts to train modern Hopfield networks with EP (Bal & Sengupta, 2023; Koulischer, 2023) and demonstrate a more hardware-friendly transformer-like model that could support both analog inference and training. We note that Gladstone et al. (2025) have recently introduced another type of energy-based transformer which, however, is not readily compatible with physical dynamics and local learning rules, which is the focus of our work.

2 The Convergent Energy Transformer

This section presents a convergent version of the ET (Hoover et al., 2024) that we call the CET. We consider the masked image completion task, where a partially masked image is given and the goal is to fill the missing regions with plausible content. We use the CELEBA dataset, treating every image as a 120×120 normalized RGB image, using standard data augmentation techniques - See appendix C for details. Each image is partitioned into 36 non-overlapping patches of size 20x20, 18 of which (50%) are chosen at random to be zeroed out. The objective is to reconstruct the masked patches from the visible ones.

2.1 CET Architecture

The model's architecture is shown in Figure 1. Denoting x the clean image, and $\bar{x}$ the partially masked one, $\bar{x}$ is viewed as a set of 121 overlapping patches of size 20x20 (with a stride of 10), which interacts with the tokens (z). Using overlapping patches helps prevent visible seams between

[1]The figure and video can be found in Hoover et al. (2023), not in Hoover et al. (2024).

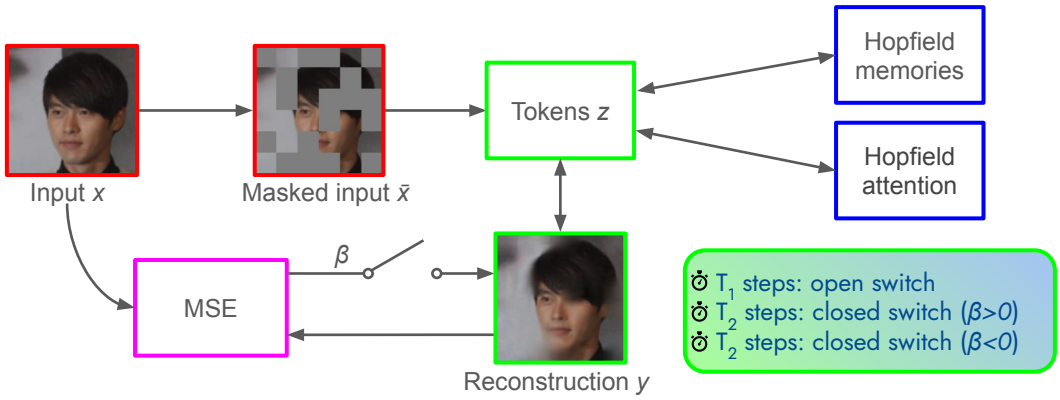


Figure 1: The CET takes a partially masked image as input (red outline) and is trained to reconstruct it. During inference, we run T_1 steps of energy minimization with open switch (meaning $\beta = 0$). During EP training, we close the switch, permitting β-scaled feedback from the cost function: we run T_2 steps with negative β to obtain a negatively-nudges state, and another T_2 steps with positive β to obtain a positively-nudges state. The two states are used to extract the parameter gradients.

patches in the reconstructed images – see Appendix B.2.1 for details. The tokens also interact with a memory module and an attention module, as well as the output layer (y) representing the reconstructed image. Dynamics are determined by an energy function defined in section 2.2. During inference, the tokens and output layer (green outline) are updated iteratively to minimize the energy function. The memory and attention modules (blue outline) use a modern Hopfield formulation, which avoids explicitly materializing these layers. As described by Hoover et al. (2024), each step of inference can be interpreted as a layer, with weights shared across layers. Thus, although the network consists of only a few modules, its "depth" comes from the temporal dimension.

2.2 The CET's Energy Function

As an energy-based model, the CET's dynamics minimize a scalar energy function E, which is the sum of individual energy terms associated with interactions between the network components. The energy function has the form

$$E(\bar{x}, z, y, \theta) = E^{\text{enc}}(\bar{x}, z, \theta^{\text{enc}}) + E^{\text{dec}}(z, y, \theta^{\text{dec}}) + E^{\text{pos}}(z, \theta^{\text{pos}}) + E^{\text{mem}}(z, \theta^{\text{mem}}) + E^{\text{att}}(z, \theta^{\text{att}}), \tag{1}$$

where E^{enc} and E^{dec} represents the encoder and decoder interactions, E^{pos} represents positional information, E^{mem} is a Modern Hopfield interaction serving as a memory bank, and E^{att} is a LogSumExp-based modern Hopfield attention energy allowing the network to learn relationships between tokens associated with distinct image patches. As previously mentioned, $\bar{x}$, z, y are the masked input image, the tokens and the reconstructed image, respectively, while $\theta = \{\theta^{\text{enc}}, \theta^{\text{dec}}, \theta^{\text{pos}}, \theta^{\text{mem}}, \theta^{\text{att}}\}$ denotes the set of all the learnable parameters. These energy terms and the parameters of each term are defined in detail in appendix B.

To use the CET for inference, we first clamp $\bar{x}$, then let z and y minimize the energy function, and finally we read the equilibrium value of y. The tokens are subject to static layer normalization constraints (zero mean and unit standard deviation) while the reconstructed image is unconstrained. The CET's equilibrium thus reads

$$(z_\star, y_\star) = \underset{z \in \mathcal{C},\ y}{\text{argmin}}\ E(\bar{x}, z, y, \theta), \tag{2}$$

where $\mathcal{C}$ denotes the set of tokens with mean 0 and std 1. To reach equilibrium, z and y perform projected gradient descent (PGD) on the energy function:

$$z \leftarrow \text{Proj}_{\mathcal{C}}(z - \epsilon \nabla_z E(\bar{x}, z, y, \theta)), \qquad y \leftarrow y - \epsilon \nabla_y E(\bar{x}, z, y, \theta), \tag{3}$$

where ϵ is the step size, until convergence. In our experiments we use $\epsilon = 1$.

In practice, in our experiments the CET does not seem to always settle to a minimum of E but sometimes settles to a critical point (saddle point) of E – see Appendix A.

2.3 Differences Between the ET and CET

In the original ET (Hoover et al., 2024), the masked input $\bar{x}$ serves as an initial condition for energy minimization, whereas in our CET it acts as a fixed boundary condition. We find that our fixed boundary condition is necessary to ensure full equilibration (to a minimum or critical point) without hampering model expressivity.

Compared to the ET, another difference in the CET is that the encoder and decoder have their own energy terms (Hopfield energy interactions), making them a part of the energy-based model. This modification significantly simplifies the EP learning rules for the encoder and decoder weights.

3 Equilibrium Propagation

The objective to optimize is $\mathcal{L} = C(y^\star, x)$ where C is a cost function representing the mismatch between input x and reconstruction (output) y. As in Hoover et al. (2024), we use the MSE.

To extract the parameter gradients $\nabla_\theta \mathcal{L}$, Equilibrium Propagation (EP) (Scellier & Bengio, 2017) introduces a parameter $\beta \in \mathbb{R}$ called the *nudging* strength, and augments the network's energy function by βC, an energy term proportional to the cost function. The network's total energy function is thus

$$F(x, z, y, \theta, \beta) = E(\bar{x}, z, y, \theta) + \beta C(x, y). \tag{4}$$

The training process involves comparing two *nudge* equilibrium states (minima of F) associated with two different values of β. Here we employ the centered variant of EP (Laborieux et al., 2021) which uses the two equilibrium states

$$(z_\star^{\pm\beta}, y_\star^{\pm\beta}) = \underset{z \in \mathcal{C},\ y}{\operatorname{argmin}}\ F(x, z, y, \theta, \pm\beta). \tag{5}$$

The main insight of EP is that the gradients of $\mathcal{L}$ can be approximated using the following formula (Scellier & Bengio, 2017; Laborieux et al., 2021):

$$\nabla_\theta \mathcal{L} = \left.\frac{d}{d\beta}\right|_{\beta=0} \frac{\partial F(x, z_\beta^*, y_\beta^*, \theta)}{\partial \theta} \approx \frac{1}{2\beta}\left(\frac{\partial F(x, z_{+\beta}^*, y_{+\beta}^*, \theta, +\beta)}{\partial \theta} - \frac{\partial F(x, z_{-\beta}^*, y_{-\beta}^*, \theta, -\beta)}{\partial \theta}\right). \tag{6}$$

Similar to the free equilibrium of equation 2, the $\pm\beta$ states are obtained using projected gradient descent (PGD) on F with respect to (z, y). In our experiments, for stability and computational efficiency, both the $\pm\beta$ states are obtained using the free equilibrium state of equation 2 as an initial state for energy minimization. Thus, to train the CET with EP, we first set $\beta = 0$ and run T_1 steps of PGD until the tokens and outputs settle to $(z_\star^0, y_\star^0)$. Next, we set $\beta > 0$ and run T_2 steps of PGD until the model reaches $(z_{+\beta}^*, y_{+\beta}^*)$. And next, starting from the free state, we set $\beta < 0$ and run another T_2 steps of PGD until the model reaches $(z_{-\beta}^*, y_{-\beta}^*)$. Finally, the two $\pm\beta$ states are used to express the gradient with respect to the weights, based on the EP formula of equation 6.

An important feature of EP is that the nudge states need not be global minima of F, but can be local minima or even just critical points (saddle points) – see for example Chapter 2 of Scellier (2021) for a discussion of this point. It seems that invoking this property is necessary to explain why EP successfully trains the CET in our experiments (Appendix A).

4 Masked Image Completion Experiments

We train the CET to do masked image reconstruction on the CELEBA-faces dataset, with a 50% masking ratio. The image is divided into $N_{\mathbf{P}} = 121$ overlapping patches, resulting in 121 tokens as well. The token dimension is $D_T = 768$, the attention module has $N_{\mathrm{H}} = 12$ attention heads of dimension $D_{\mathrm{H}} = 64$, and the Hopfield memory module has $N_{\mathrm{M}} = 3072$ memories of size $D_{\mathbf{T}} \times N_{\mathrm{M}}$.

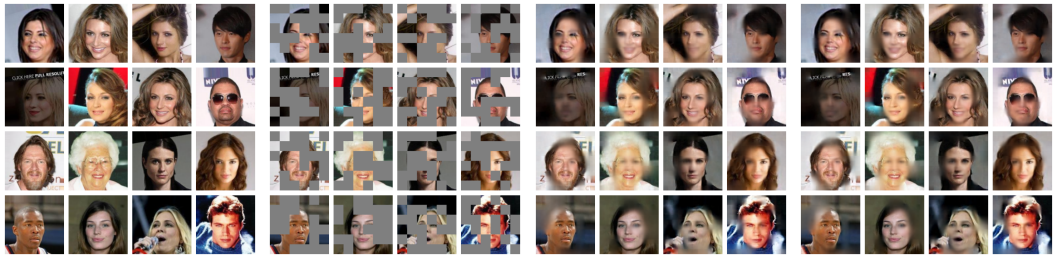
We use EP as a training method, and compare it with the baseline obtained using truncated backpropagation through the equilibration process (TBPTE). For EP, we run the free phase for $T_1 = 150$ steps and the two nudged phases for $T_2 = 5$ steps. For TBPTE, we run energy minimization for 155 steps and backpropagate through the last 5 steps. For the full list of hyperparameters used in our experiments, see appendix C.

Table 1: Pixel-wise mean squared error of the trained CET. EP achieves comparable performance to the baseline obtained using truncated backpropagation through equilibration (TBPTE). Note that images are normalized to the range [-1,1].

	EP	TBPTE
Training dataset	0.01413	0.01371
Testing dataset	0.01422	0.01376

The mean squared pixel errors (Square error normalized by $C \times h_{\text{I}} \times w_{\text{I}}$) for the trained models are reported in Table 1. We observe that the EP-trained model is competitive with TBPTE-trained model. Figure 2 shows the reconstructions after 150 gradient steps. While the CET is able to grasp the global structure of the data and fill in the missing parts we notice that the reconstructions often appear blurry. This is a common issue with vision transformers in general, which can be remedied in various ways. The simplest solution is to use smaller patches, but this will significantly increase the number of patches (and hence the number of tokens), leading to much longer training times.

In Appendix A, we show that for a part of the images from the test set, after $T_1 = 150$ steps of energy minimization, the CET hasn't converged to a minimum but rather a saddle point of the energy function. For these images, performing more steps of energy minimization eventually brings the CET in a new equilibrium state with significantly lower energy, and the reconstructed image associated to it is of significantly poorer quality. In practice, these spurious energy minima do not seem to pose any problem as long as the model is used in the same conditions during inference and during training, i.e. using $T_1 = 150$ steps of energy minimization during the free phase during both inference and training.



(a) Clean test images (b) Masked test images (c) EP-CET output (d) TBPTE-CET output

Figure 2: (a) 16 example images from the test datasets. (b) The same mask is applied before the EP-trained model (c) and the TBPTE-trained model (d) are used to reconstruct the image.

5 Conclusion

We have developed a convergent version of the Energy Transformer compatible with EP, and trained it to do image reconstruction on the CELABA dataset. The CET treats the partially masked input image as a fixed boundary condition (rather than an initial condition for energy minimization as is done in the ET), and includes explicit energy terms for the encoder and decoder. These modifications allow to perform full equilibration (a requirement of EP) and simplify the analytical expressions of the EP learning rule for the encoder and decoder weights, making them more amenable to hardware implementation. In this respect, our work is complementary of Bacvanski et al. (2025), showing how the ET can be implemented in analog electrical circuits not just for inference, but also for learning. Our CELEBA experiments demonstrate that EP achieves comparable performance with the baseline obtained via truncated backpropagation through the equilibration process (TBPTE). An interesting aspect of the CET is that, in the regime where we trained it (using 150 steps of energy minimization in the free phase), it seems to occasionally converge to saddle points of the energy function, rather than minima. This does not pose any problem for EP training, and does not pose any issue during inference either as long as the CET is used in the same conditions where it was trained (using 150 steps of energy minimization).

ACKOWLEDGEMENTS

We thank Benjamin Hoover, Axel Laborieux, Félix Koulischer, Cédric Goemaere and Malyaban Bal for discussions on Modern Hopfield Networks, the Energy Transformer and their connection with EP. We thank Rain AI and the Advanced Research + Invention Agency's (ARIA) Scaling Compute programme for funding this work.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Maxwell Anderson, Shi-Yuan Ma, Tianyu Wang, Logan Wright, and Peter McMahon. Optical transformers. *Transactions on machine learning research*, 2023.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, April 2024. doi: 10.1145/3620665.3640366. URL `https://docs.pytorch.org/assets/pytorch2-2.pdf`.

Marc Gong Bacvanski, Xincheng You, John Hopfield, and Dmitry Krotov. Dense associative memories with analog circuits. *arXiv preprint arXiv:2512.15002*, 2025.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Malyaban Bal and Abhronil Sengupta. Sequence learning using equilibrium propagation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 2949–2957, 2023.

Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen. The rising costs of training frontier ai models. *arXiv preprint arXiv:2405.21015*, 2024.

Nima Dehmamy, Benjamin Hoover, Bishwajit Saha, Leo Kozachkov, Jean-Jacques Slotine, and Dmitry Krotov. Nrgpt: An energy-based alternative for gpt. *arXiv preprint arXiv:2512.16762*, 2025.

Sam Dillavou, Menachem Stern, Andrea J Liu, and Douglas J Durian. Demonstration of decentralized physics-driven learning. *Physical Review Applied*, 18(1):014040, 2022.

Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

Maxence Ernoult, Julie Grollier, Damien Querlioz, Yoshua Bengio, and Benjamin Scellier. Updates of equilibrium prop match gradients of backprop through time in an rnn with static input. *Advances in neural information processing systems*, 32, 2019.

Alexi Gladstone, Ganesh Nanduru, Md Mofijul Islam, Peixuan Han, Hyeonjeong Ha, Aman Chadha, Yilun Du, Heng Ji, Jundong Li, and Tariq Iqbal. Energy-based transformers are scalable learners and thinkers. *arXiv preprint arXiv:2507.02092*, 2025.

Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed J Zaki, and Dmitry Krotov. Energy transformer, 2023. URL https://openreview.net/forum?id=4nrZXPFN1c4.

Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed Zaki, and Dmitry Krotov. Energy transformer. *Advances in Neural Information Processing Systems*, 36, 2024.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Jack Kendall. A gradient estimator for time-varying electrical networks with non-linear dissipation. *arXiv preprint arXiv:2103.05636*, 2021.

Jack Kendall, Ross Pantone, Kalpana Manickavasagam, Yoshua Bengio, and Benjamin Scellier. Training end-to-end analog neural networks with equilibrium propagation. *arXiv preprint arXiv:2006.01981*, 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

Felix Koulischer. Exploration of the link between modern hopfield networks and transformers through equilibrium propagation. 2023. URL https://libstore.ugent.be/fulltxt/RUG01/003/150/039/RUG01-003150039_2023_0001_AC.pdf. Accessed: 2025-05-12.

Axel Laborieux and Friedemann Zenke. Holomorphic equilibrium propagation computes exact gradients through finite size oscillations. *Advances in Neural Information Processing Systems*, 35: 12950–12963, 2022.

Axel Laborieux, Maxence Ernoult, Benjamin Scellier, Yoshua Bengio, Julie Grollier, and Damien Querlioz. Scaling equilibrium propagation to deep convnets by drastically reducing its gradient estimator bias. *Frontiers in neuroscience*, 15:129, 2021.

Jérémie Laydevant, Danijela Marković, and Julie Grollier. Training an ising machine with equilibrium propagation. *Nature Communications*, 15(1):3671, 2024.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Danijela Marković, Alice Mizrahi, Damien Querlioz, and Julie Grollier. Physics for neuromorphic computing. *Nature Reviews Physics*, 2(9):499–510, 2020.

Serge Massar. Equilibrium propagation for learning in lagrangian dynamical systems. *Physical Review E*, 112(3):035304, 2025.

Serge Massar and Bortolo Matteo Mognetti. Equilibrium propagation: the quantum and the thermal cases. *arXiv preprint arXiv:2405.08467*, 2024.

Ali Momeni, Babak Rahmani, Benjamin Scellier, Logan G Wright, Peter L McMahon, Clara C Wanjura, Yuhang Li, Anas Skalli, Natalia G Berloff, Tatsuhiro Onodera, et al. Training of physical neural networks. *arXiv preprint arXiv:2406.03372*, 2024.

Seokjin Oh, Jiyong An, Seungmyeong Cho, Rina Yoon, and Kyeong-Sik Min. Memristor crossbar circuits implementing equilibrium propagation for on-device learning. *Micromachines*, 14(7): 1367, 2023.

Guillaume Pourcel, Debabrota Basu, Maxence Ernoult, and Aditya Gilra. Lagrangian-based equilibrium propagation: generalisation to arbitrary boundary conditions & equivalence with hamiltonian echo learning. *arXiv preprint arXiv:2506.06248*, 2025.

Théophile Rageau and Julie Grollier. Training and synchronizing oscillator networks with equilibrium propagation. *Neuromorphic Computing and Engineering*, 5(3):034008, 2025.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Benjamin Scellier. *A deep learning theory for neural networks grounded in physics*. PhD thesis, Université de Montréal, 2021.

Benjamin Scellier. Quantum equilibrium propagation: Gradient-descent training of quantum systems. *arXiv preprint arXiv:2406.00879*, 2024.

Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Qingshan Wang, Clara Wanjura, and Florian Marquardt. Training coupled phase oscillators as a neuromorphic platform using equilibrium propagation. *Neuromorphic Computing and Engineering*, 2024.

Clara C Wanjura and Florian Marquardt. Quantum equilibrium propagation for efficient training of quantum systems based on onsager reciprocity. *arXiv e-prints*, pp. arXiv–2406, 2024.

Su-in Yi, Jack D Kendall, R Stanley Williams, and Suhas Kumar. Activity-difference training of deep neural networks using memristor crossbars. *Nature Electronics*, 6(1):45–51, 2023.

Qian Zhang, Dmitry Krotov, and George Em Karniadakis. Operator learning for reconstructing flow fields from sparse measurements: an energy transformer approach. *Journal of Computational Physics*, 538:114148, 2025.

Gianluca Zoppo, Francesco Marrone, Michele Bonnin, and Fernando Corinto. Equilibrium propagation and (memristor-based) oscillatory neural networks. In *2022 IEEE international symposium on circuits and systems (ISCAS)*, pp. 639–643. IEEE, 2022.

Nicolas Zucchet and João Sacramento. Beyond backpropagation: bilevel optimization through implicit differentiation and equilibrium propagation. *Neural Computation*, 34(12):2309–2346, 2022.

A Equilibrium States of the CET: Energy Minima or Critical Points?

In this appendix, we take a closer look at the CET's energy minimization dynamics and the equilibrium states that it reaches, showing that these equilibria are more likely critical points than minima of the energy function.

Taking a trained CET (with EP) and images from the test dataset, we plot the energy minimization curves, that is, the energy as a function of the number of minimization steps (T) - see Figure 3. For most of the images, the energy seems to have converged after 150 steps, but for 5 of the 16 images - those with indices 1, 4, 8, 12 and 16 - a significant drop in energy occurs hundreds of steps later. image 5 also undergoes a small but visible drop in energy. This hints that, for part of the masked input images, the equilibrium state reached after $T = 150$ steps of energy minimization is not a minimum of the energy function, but more likely a critical point. This does not pose any problem for EP training as EP only requires critical points, not necessarily energy minima.

We then take a look at the CET's reconstructed images after T steps of energy minimization, for different values of T ranging from 2 to 100,000 - see Figure 4. From the image indices, we see that the images experiencing a drop in energy are also accompanied by a significant degradation of the quality of reconstructed images. This shows the existence of spurious minima in the energy landscape. In practice these spurious minima do not seem to pose any issue as long as the number of energy minimization steps used at test time matches the one used during training ($T = 150$).

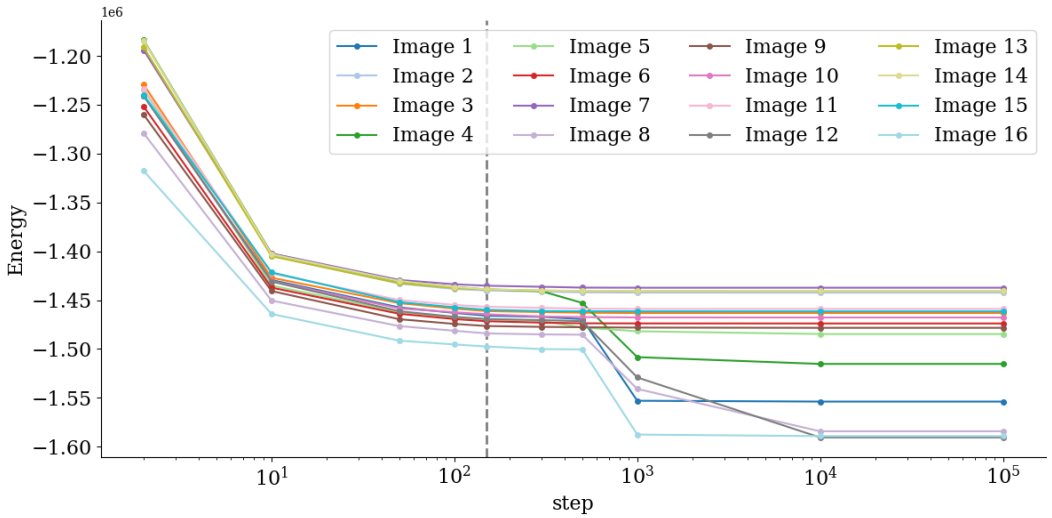


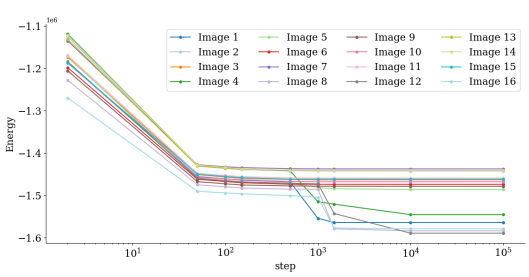
Figure 3: Energy as a function of the number of gradient steps (T) for 16 different images from the test sets, using step size $\epsilon = 1.0$. The image numbers correspond to the image numbers in Figure 4. The vertical dashed line denotes step 150 (the number of gradient steps used during training).

Figure 4: Reconstructed images produced by the EP-trained CET, shown for different number of gradient steps, using step size $\epsilon = 1.0$.

A.1 Changing the step size of projected gradient descent

In our experiments, we minimize the CET's energy using projected gradient descent with a step size of 1. While we empirically observe that the energy steadily goes down after each PGD step, it is important to emphasize that we do not have a proof for this property: GD or PGD with fixed (non-infinitesimal) step size is not guaranteed to decrease the energy function. The continuous-time version of the PGD dynamics (gradient flow), however, steadily goes down the energy. Here we speculate that the (discrete-time) PGD dynamics approximates well the continuous-time gradient flow.

To test this hypothesis, we repeat the inference experiments of Figure 3 and Figure 4 with a 10x smaller step size ($\epsilon = 0.1$). The result, plotted in Figure 5), is very similar (converging to roughly the same energies and with the same images collapsing) as the experiments with step size 1.0, albeit at an increased computational cost.



(a) Energy as a function of gradient steps (T) for 16 images.

(b) Reconstructed images produced by the EP-trained CET for different gradient steps.

Figure 5: Energy curves and reconstructions when running with a reduced step size $\epsilon = 0.1$.

B Definition of the CET's Energy Function

In this appendix, we define the terms of the CET's energy function, as well as the dimensions of the tensors involved in these energy terms.

B.1 CET Hyperparameters and tensor dimensions

Table 2 defines the CET's hyperparameters, including the symbols used to represent them and the values used in the CELEBA experiments. Table 3 defines the dimensionality of the state and weight tensors in terms of the CET hyperparameters.

The number of tokens is equal to the number of patches (N_P), which, letting $\lfloor\cdot\rfloor$ denote the floor function, is calculated as

$$N_{\mathrm{P}} = \lfloor \tfrac{h_{\mathrm{I}}-h_{\mathrm{P}}}{s_{\mathrm{P}}} + 1 \rfloor \times \lfloor \tfrac{w_{\mathrm{I}}-w_{\mathrm{P}}}{s_{\mathrm{P}}} + 1 \rfloor = (\tfrac{120-20}{10} + 1) \times (\tfrac{120-20}{10} + 1) = 11 \times 11 = 121. \quad (7)$$

In the second step, we plugged in the values from Table 2. This expression is just the product of the width and height of the output of a convolutional layer with no padding and dilation 1 (see for example section 2.3 in Dumoulin & Visin (2016)).

B.2 Energy Function Definition

The energy function is of the form

$$E(\bar{x}, z, y, \theta) = E^{\mathrm{enc}}(\bar{x}, z, W^{\mathrm{enc}}, b^{\mathrm{enc}}) + E^{\mathrm{dec}}(z, y, W^{\mathrm{dec}}, b^{\mathrm{dec}}) + E^{\mathrm{pos}}(z, b^{\mathrm{pos}}) \quad (8)$$

$$+ E^{\mathrm{mem}}(z, W^{\mathrm{mem}}) + E^{\mathrm{att}}(z, W^K, W^Q), \quad (9)$$

where θ denotes the set of all learnable parameters: $\theta = \{W^{\mathrm{enc}}, b^{\mathrm{enc}}, W^{\mathrm{dec}} \dots\}$. Next, we define each of these energy terms.

B.2.1 Encoder

The encoder is modeled as a Hopfield interaction between patches and tokens. Because the same encoder weights are used for all patches of the image, it is convenient to model it as a convolutional Hopfield interaction between masked images ($\bar{x}$) and tokens (z), where the patchification and linear projection from patches to tokens is performed using a single convolutional operation. The encoder's energy is

$$E^{\mathrm{enc}}(\bar{x}, z, W^{\mathrm{enc}}, b^{\mathrm{enc}}) = \tfrac{1}{2} \sum_{ij} z_{ij}^2 - \sum_{ij} [F(\bar{x}, W^{\mathrm{enc}})]_{ij} z_{ij} - \sum_{i,j} z_{ij} b_i^{\mathrm{enc}}, \quad (10)$$

where $F(u, V)$ denotes the convolution between an input tensor u and a kernel V, composed with flattening over the spatial dimensions:

$$F(u, V) = \mathrm{flatten}(\mathrm{conv2d}(V, u)). \quad (11)$$

The padding and stride, omitted in equation 11, determine the number of patches (N_{P}) via equation 7. The convolutional operation makes it straightforward to use overlapping patches, by changing the stride.

In equation 10, $F(\bar{x}, W^{\mathrm{enc}})$ is a matrix of shape (D_T, N_P), the same shape as the tokens (z), and $[F(\bar{x}, W^{\mathrm{enc}})]_{ij}$ denotes an element of this matrix.

B.2.2 Positional Bias

A learnable 'positional bias' is responsible for encoding the patches' spatial location. To this end, the tokens interact with these biases via a positional interaction whose energy is

$$E^{\mathrm{pos}}(z, b^{\mathrm{pos}}) = -\sum_{ij} z_{ij} b_{ij}^{\mathrm{pos}}. \quad (12)$$

Unlike the bias from the encoder, the positional bias is not shared between tokens, allowing the network to learn to encode spatial information.

Table 2: CET Hyperparameters. The number of patches N_{P} is calculated as in equation 7. By construction, the number of tokens is equal to the number of patches, and the memory dimension is equal to the token dimension.

Category	**Hyperparameter**	**Symbol**	**Value** (CELEBA Experiment)
Image	Channels	C	3
	Height	h_{I}	120
	Width	w_{I}	120
Patch	Height	h_{P}	20
	Width	w_{P}	20
	Stride	s_{P}	10
	Number of patches	N_{P}	121
Tokens	Number of tokens	N_P	(121)
	Token dimension	D_{T}	768
Attention	Number of heads	N_{H}	12
	Head dimension	D_{H}	64
	Inverse temperature	γ	$1/4$
Memory	Number of memories	N_M	3072
	Memory dimension	D_T	(768)

Table 3: Dimensions of state and weight tensors.

Description	**Symbol**	**Dimensionality**
State tensors (non-learnable)		
Image	x	$C \times h_{\mathrm{I}} \times w_{\mathrm{I}}$
Masked image	$\bar{x}$	$C \times h_{\mathrm{I}} \times w_{\mathrm{I}}$
Tokens	z	$D_{\mathrm{T}} \times N_{\mathrm{P}}$
Reconstructed image	y	$C \times h_{\mathrm{I}} \times w_{\mathrm{I}}$
Weight tensors (learnable parameters)		
Encoder weights	W^{enc}	$D_{\mathrm{T}} \times C \times h_{\mathrm{P}} \times w_{\mathrm{p}}$
Encoder bias	b^{enc}	D_{T}
Positional bias	b^{pos}	$D_{\mathrm{T}} \times N_{\mathrm{P}}$
Decoder weights	W^{dec}	$D_{\mathrm{T}} \times C \times h_{\mathrm{P}} \times w_{\mathrm{p}}$
Decoder bias	b^{dec}	C
Key weights	W^K	$N_{\mathrm{H}} \times D_{\mathrm{H}} \times D_{\mathrm{T}}$
Query weights	W^Q	$N_{\mathrm{H}} \times D_{\mathrm{H}} \times D_{\mathrm{T}}$
Memory weights	W^{mem}	$D_{\mathrm{T}} \times N_{\mathrm{M}}$

B.2.3 Attention Module

The attention module is responsible for exchanging information between the tokens. Similar to the attention mechanism of a standard transformer, the attention module comprises key and query tensors of shape $N_{\mathrm{H}} \times D_{\mathrm{H}} \times D_{\mathrm{T}}$ (number of attention heads $\times$ head dimension $\times$ token dimension). Each token is mapped N_H times into an internal space, generating the key and query tensors Q and K. The Attention tensor A is computed from Q and K as follows:

$$Q_{ijl} = \sum_k W^Q_{ijk} z_{kl}, \qquad K_{ijl} = \sum_k W^K_{ijk} z_{kl}, \qquad A_{lmn} = \sum_j Q_{mjl} K_{njl}. \tag{13}$$

The energy associated with the attention module is

$$E^{\mathrm{att}}(z, W^Q, W^K) = -\tfrac{1}{\gamma} \sum_{lm} \log \left(\sum_n \exp(\gamma A_{lmn}) \right), \tag{14}$$

where $\gamma > 0$ is a scalar, referred to as the inverse temperature - usually denoted as β, but here we use γ to avoid confusion with the nudging parameter β of EP.

As in the original ET architecture (Hoover et al., 2024) there is no separate value tensor.

B.2.4 MEMORY MODULE

The memory module plays a role analogous to the MLP in a standard transformer. It is responsible for learning general information about images from the dataset, ensuring that the tokens are consistent with what one expects to see in realistic images. Its energy function is

$$E^{\text{mem}}(z, W^{\text{mem}}) = -\sum_{jk}(\text{ReLU}(\sum_{i} W^{\text{mem}}_{ik} z_{ij}))^2. \tag{15}$$

Importantly, each token is subject to the same linear transformation. In the memory module, tokens only interact with stored memories, not with each other ; inter-token communication takes place in the attention module.

The columns of the memory matrix W^{mem} can be interpreted as a set of memories learned from the training data. During energy minimization at test time, the tokens tend to align with these memories to find similar patterns as in the training data.

B.2.5 DECODER

Similar as the encoder, the decoder's energy between tokens (z) and reconstructed image (y) is

$$E^{\text{dec}}(z, y, W^{\text{dec}}, b^{\text{dec}}) = \tfrac{1}{2}\sum_{ij} y_{ij}^2 - \sum_{ij}[F(y, W^{\text{dec}})]_{ij} z_{ij} - \sum_{ijk} y_{ijk} b_i^{\text{dec}}. \tag{16}$$

B.2.6 TOKEN NORMALIZATION

To ensure that the tokens are normalized, a constraint is imposed on the mean and std of the free- and nudge-equilibrium tokens, represented by the feasible set $\mathcal{C}$ in equation 2 and equation 5. Energy minimization then amounts to projected gradient descent, with projection onto the set of tokens with zero mean and unit standard deviation.

We note that normalization is formulated differently in Hoover et al. (2024), using an energy-based layer-norm operation with learnable mean and std.

C CELEBA EXPERIMENT DETAILS

Hyperparameters related to the model architecture are listed in Table 2. Hyperparameters related to inference and training are listed in Table 4. Our simulations were implemented in PyTorch (Ansel et al., 2024). We used the AdamW optimizer (Loshchilov & Hutter, 2017) with a cosine decay learning rate schedule. The hyperparameter T_2 denotes both the number of equilibration steps in the nudged phase of EP, and the number of equilibration steps through which we backpropagate when employing TBPTE.

We note that what we call 'backpropagation through equilibration' (BPTE) has been referred to as backpropagation through time (BPTT) in previous works (Ernoult et al., 2019). We choose this terminology to avoid confusion, because BPTT is traditionally employed in models with time-varying inputs such as LSTMs, whereas in the CET and other energy-based models, the input provided to the model is static throughout equilibration.

All images were normalized to the $[-1, 1]$ range. The 178x218 CELEBA images were center cropped to square images of size 178x178, before getting resized to size 120x120. During training a stochastic resizing scheme was implemented via PyTorch's RandomResizedCrop augmentation, using a scale parameter of $(0.9, 1.0)$.

Table 4: Hyperparameters

Parameter	**Value**
Weight decay	3×10^{-5}
Initial learning rate	4×10^{-4}
Final learning rate	1×10^{-6}
Free gradient steps (T_1)	150
Nudged gradient steps (T_2)	5
Nudging strength (β)	0.01
Step size (ϵ)	0.1