# DYADIC LEARNING IN ASYMMETRIC CONVNETS

**Timothy Nest**
Montreal Institute of Learning Algorithms
timothy.nest@mila.quebec

**Rasmus Hoier**
Rain AI
rasmus@rain.ai

## ABSTRACT

Dual propagation is a local learning algorithm that treats neurons as simple two-compartment structures (dyads), encoding errors as their internal difference and predictions as their mean. Originally limited to feedforward (lower-triangular) models, a recent generalization, dyadic learning, extends to networks with arbitrary connectivity. Here we show for the first time that such models can be effectively trained on CIFAR-10, and exhibit varied benefits and drawbacks depending on the structure of the weight matrix. In particular, symmetric, skew-symmetric, feedforward and general asymmetric convolutional networks are assessed in both classification and denoising settings. We observe that the skew-symmetric and asymmetric models perform the best on the denoising task and perform competitively in the classification task .

## 1 INTRODUCTION

The widespread use of AI systems, traditionally reliant on the combination of the backpropagation (BP) and feedforward (ff) architectures, has in recent years been characterized by larger models, and a growing energy burden. As energy costs increase so has the interest in novel learning algorithms and hardware accelerators. In principle, special purpose hardware designed for the primitives of deep neural networks (vectorized multiply-add operations and nonlinearities) can yield substantial energy savings, as well as acceleration of certain critical operations, especially if a majority of the computations are carried out in the analog domain (Kendall et al., 2020; Kalinin et al., 2023). In spite of this, analog compute engines also introduce new challenges. In addition to posing practical barriers for training, device variability and non-idealities make for a kind of *mortal computing* (Hinton, 2022), as weights can not be copied from one machine to another, unlike digital *immortal* computing, where weights are easily shared between devices. Two solutions are to train on-device in the analog domain–either entirely from scratch, or to finetune a feedforward model consisting of imperfectly transferred parameters, using analog-amenable learning algorithms.

While backpropagation has powered deep learning for decades, it requires synchronization of operations and knowledge of the computational graph, which is both a theoretical concern for those interested in biological learning, and a practical one for mortal computing. Consequently, there is a strong synergy between analog on device training and local bio-inspired learning algorithms. This line of research, which includes predictive coding (Whittington & Bogacz, 2017; Song et al., 2020), contrastive Hebbian learning (Movellan, 1991; Xie & Seung, 2003) and equilibrium propagation (Scellier & Bengio, 2017), may be of major practical relevance for hardware-algorithm co-design, wherein weight updates are expressed in terms of the neural activations (e.g. voltage measurements in a resistive circuits) measured after the network has converged to an equilibrium state.

A serious limitation of such bio-inspired methods is that they often assume either a feedforward (lower triangular) or symmetric connectivity, rather than the diverse, and asymmetric connectivity found in biology. Equilibrium propagation (EP) (Scellier & Bengio, 2017) and contrastive Hebbian learning (CHL) (Xie & Seung, 2003) have traditionally focused on local learning rules for symmetrically connected networks (i.e. undirected graphs). This maps well to resistive networks (Kendall et al., 2020), where the edge from node $i$ to $j$ is the same physical object as the edge from node $j$ to $i$. However, for other substrates this symmetry is infeasible, e.g. in optical computing where each edge represents a distinct physical area of a spatial light modulator, or in biological neural networks, where most edges are directed. Attempts to generalize EP and CHL to asymmetrically connected

networks have resulted in ad hoc modifications, that lack the strong theoretical guarantees of EP, and have so far failed to perform well on challenging tasks (Scellier et al., 2018; Detorakis et al., 2019) (see Section 4.1).

Our work builds on dyadic learning (Høier et al., 2024), a generalization of dual propagation(Høier et al., 2023; Høier & Zach, 2024), which presents a principled way to train networks of arbitrary connectivity. Under the assumption of symmetric connectivity, dyadic learning recovers the inference and training dynamics of equilibrium propagation in symmetric Hopfield networks (Ernoult et al., 2019). Under the assumption of lower triangular (feedforward) connectivity dyadic learning recovers dual propagation. Though the theory applies to networks with arbitrary topology, dyadic learning has thus far only been applied to MLP-type deep Hopfield networks with symmetric, feedforward and skew-symmetric connectivity. Here, we show that dyadic learning can be successfully applied to train convolutional Hopfield networks, including those with fully asymmetric connectivity, in the classification setting. We also apply this method to a new biologically inspired denoising network, in which the network leverages feedback from deeper layers to denoise reconstructions adjacent to the inputs.

### 1.1 Contributions

- We apply dyadic learning to train symmetric, skew-symmetric, lower triangular (feedforward), and general asymmetric convolutional Hopfield models on CIFAR-10.
- We design a convolutional feedback denoising network, in which prediction units are adjacent to the input and yet benefit from model depth via feedback from deeper layers.
- We train symmetric, skew-symmetric and asymmetric variants of this network to denoise noisy Fashion-MNIST images, demonstrating diverse representational and stability advantages of negative feedback which are robust to limited training data.

## 2 Dyadic Learning

We employ the following formulation of the dyadic objective , first proposed in (Høier et al., 2024; Høier, 2025).

$$\min_{W} \min_{s^+} \max_{s^-} \beta \bar{C}(s^+, s^-, y) + G(s^+) - G(s^-) - (s^+ - s^-)^\top (W \tfrac{1}{2}(s^+ + s^-) + Ux) \tag{1}$$

The loss function $\bar{C}(s^+, s^-, y) = \frac{1}{2}(C(s^+, y) + C(s^-, y))$ is the average of costs (e.g. square error loss) associated with $s^+$ and $s^-$ (and the target $y$). In this framework individual neurons are dyads possessing two internal states $s^+ \in \mathbb{R}^N$ and $s^- \in \mathbb{R}^N$. The weight matrix $W \in \mathbb{R}^{N \times N}$ connects the neurons to each other and the input layer weight matrix $U \in \mathbb{R}^{N \times M}$ connects inputs $x$ to the neurons. For simplicity we do not write out layers explicitly or specialize to convolutional setting, as both can be seen as special cases of the dense setting with appropriate sparsity and weight sharing. By defining $E(u, v, W) = G(u) - u^\top W v$, with G a strongly convex function chosen to induce a desired activation function[1], Equation 1 can also be expressed as

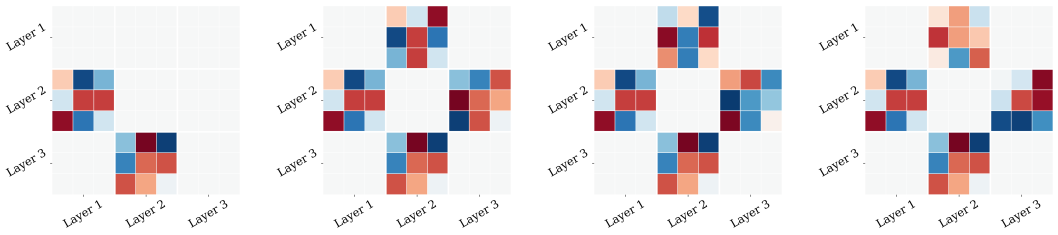
$$\begin{aligned} \min_{W} \min_{s^+} \max_{s^-} \; & \beta \bar{C}(s^+, s^-, y) - (s^+ - s^-)^\top U x \\ & + \tfrac{1}{2} \left( E(s^+, s^+, W) - E(s^-, s^-, W) + E(s^+, s^-, W) - E(s^-, s^+, W) \right) \end{aligned} \tag{2}$$

This formulation looks very similar to the contrastive objectives employed in equilibrium propagation based training of symmetric Hopfield networks. The key difference here is that we are allowing coupling between the $s^+$ and $s^-$ states. The exact structure of $W$ defines the model and leads to very different inference and training dynamics.

Figure 1 depicts the archetypal settings for $W$ considered in this work. If $W$ is symmetric, then the two last terms, $E(s^+, s^-)$ and $E(s^-, s^+)$, sum to zero. If $W$ is instead skew-symmetric then the non mixing terms, $E(s^+, s^+)$ and $E(s^-, s^-)$, each evaluate to zero. For lower triangular (feedforward) $W$ as well as asymmetric $W$ none of the terms vanish. As noted in Høier et al. (2024) symmetric models and specifically layered skew-symmetric models allow the decoupling of inference into two

[1] Choose $G$ such that $\arg\min_u G(u) - u^\top v = f(v)$, where f is desired activation function.

sequential phases in which neurons only transmit their own activity. In the case of a lower triangular $W$ the mean $\frac{1}{2}(s^+ + s^-)$ is propagated forwards through the network via $W$, whilst the difference $(s^+ - s^-)$ is propagated backwards via $W^\top$. This is the dual prop setting. General asymmetric connectivity requires the network to propagate the mean $\frac{1}{2}(s^+ + s^-)$ in both directions via $W$ and the differences $(s^+ - s^-)$ via $W^\top$. Of course at evaluation time (when $\beta = 0$) we have $s^+ = s^-$ so there is no finite difference error signal. Consequently, while the inference dynamics of the asymmetric model appear more biologically plausible, during training one still encounters the weight transport problem (Grossberg, 1987), and neurons are required to do a great deal of multiplexing. It is nevertheless, possible to address this via strategies such as the ones employed in the feedback alignment litterature (Lillicrap et al., 2014; Akrout et al., 2019).




(a) Feedforward (b) Symmetric (c) Skew-symmetric (d) Asymmetric

Figure 1: Four different structures for the adjacency matrix $W$ in a small Hopfield network with 3 layers and 9 neurons.

### 2.1 Inference dynamics

We employ the mirror descent (Nemirovskij & Yudin, 1983) inference scheme proposed in (Høier et al., 2024). It is convenient to use $F$ as a shorthand for the sum of the terms in Equation 1. Thus Equation 1 can be restated as

$$\min_{W} \min_{s^+} \max_{s^-} F(s^+, s^-, \theta, \beta), \tag{3}$$

where $\theta = \{W, U\}$. We omit the dependence on input data x and target y for brevity. At each timestep $t$ the gradient of $F$ with respect to $s^+$ and $s^-$ is computed in the space of the activations. This gradient is then used to update the pre-activations using a step size $\alpha \in (0, 1]$.

$$a^+(t+1) \leftarrow a^+(t) - \alpha \nabla_{s^+} F(s^+(t), s^-(t), \theta, \beta) \tag{4}$$

$$a^-(t+1) \leftarrow a^-(t) + \alpha \nabla_{s^-} F(s^+(t), s^-(t), \theta, \beta) \tag{5}$$

Subsequently, the activations are computed by applying the activation function corresponding to the function $G$.

$$s^+(t+1) \leftarrow f(a^+(t+1)), \qquad s^-(t+1) \leftarrow f(a^-(t+1)) \tag{6}$$

We observe that feedforward models and symmetric models benefit from a large step size $\alpha = 1$. Incidentally, this recovers the exact block coordinate descent updates used by Ernoult et al. (2019) when $W$ is symmetric and recovers the exact block coordinate descent updates used by Høier et al. (2023) when $W$ is lower triangular. In order to train the skew-symmetric and asymmetric models stably, a small step size $\alpha < 1$was found to work best.

### 2.2 Gradient estimation

The inference dynamics run until the network reaches equilibrium. We denote the equilibrium state (a saddlepoint) as $(s^+_\beta, s^-_\beta)$, and denote their concatenation $s_\beta$, to highlight that $\beta > 0$ is what forces $s^+$ and $s^-$ to differ at their steady state. Once at equilibrium the weights are simply updated via a gradient descent step with respect to $\theta$.

$$\Delta\theta \propto -\nabla F(s^+, s^-, \theta, \beta)/\beta \tag{7}$$

One way to justify this is through the lens of the equilibrium propagation theorem (Scellier & Bengio, 2017), which is a highly general result, not only applicable to learning in energy based model, but applicable to bi-level optimization problems in general (Zucchet & Sacramento, 2022). From this perspective the equilibrating variable is the concatenation of $s^+$ and $s^-$.

# 3 EXPERIMENTS

We explore two problem settings that showcase some of the novel advantages and challenges of asymmetric and skew-symmetric models as compared to symmetric and feedforward models. We employ square error loss for all experiments. All models and training details are presented in the appendix.

## 3.1 IMAGE CLASSIFICATION ON CIFAR-10

First, we compare performance in image classification on Cifar-10 of a 5-layer convolutional Hopfield network. This task has been thoroughly explored in previous work on energy-based learning and serves as a common baseline in the equilibrium prop literature. The architecture we employed is similar to that employed in Scellier et al. (2023) and Elayedam & Srinivasan (2025), with the exception that we replace max-pooling with squeezing operations (see Appendix A). We compare performance across 4 settings: symmetric, skew-symmetric, asymmetric, and lower-triangular, which serves as a feedforward baseline. All models are trained for 100-epochs, with identical hyperparameters, except for the asymmetric model which suffers from training instability and thus requires substantially lower initial learning rates.

As seen in Table 1, all models, with the exception of the skew-symmetric model, train successfully, achieving comparable performance to that of previous works on similarly sized networks Scellier et al. (2023)(without special activations or advanced data augmentation as employed in (Elayedam & Srinivasan, 2025). Despite a non-negligible drop in performance compared with other settings, negative feedback aids stability in the skew-symmetric setting, yielding accelerated inference dynamics, with unconstrained inference iterations to convergence never exceeding 20. Since the negative feedback of the skew-symmetric setting enables stable training without positive clamping, we also demonstrate training with ReLU activations, rather than the hard-sigmoid clamping typically required in energy-based training. Notably, the asymmetric model, while highly unstable, outperforms EP, and achieves validation accuracy within margins of error of our feed-forward baseline while relying on lower initial learning rate (by necessity/to avoid collapse).

| Method | Val Acc (%) |
|---|---|
| Skew-Symmetric | $84.9 \pm 0.1$ |
| Skew-symmetric w/ ReLU | $85.3 \pm 0.2$ |
| Symmetric (EP) | $89.0 \pm 0.2$ |
| Asymmetric | $89.5 \pm 0.4$ |
| Feedforward (DP) | $89.8 \pm 0.1$ |

Table 1: Validation accuracy on CIFAR-10.

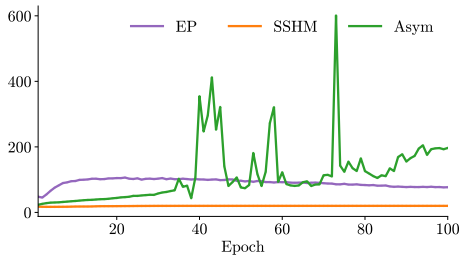



Figure 2: Avg. inference steps per epoch.

## 3.2 DENOISING ON FASHION-MNIST

Next we design a novel convolutional Hopfield network to perform image denoising. In this setting the output layer is placed immediately after the input layer and is followed by 3 more layers. The subsequent layers provide feedback which the recurrent models (symmetric, skew-symmetric and asymmetric) can learn to exploit. The feedforward network is included as a baseline, demonstrating how much of the work is done by the first layer. Further details of this architecture are provided in the appendix.

We train the model on 0.5%, 5%, 25% and 95% of the training data, while reserving 5% of the training data for validation and model selection. The asymmetric model learns very rapidly, but becomes unstable during longer runs, crashing in all but the lowest data setting, however in the minimal data setting it outperforms all models. The skew-symmetric model is stable throughout and yields the best reconstructions (see Figure 4. The symmetric model struggles in the low data settings (it is even worse than the feed-forward baseline), but eventually performs adequately when given access to sufficient data.

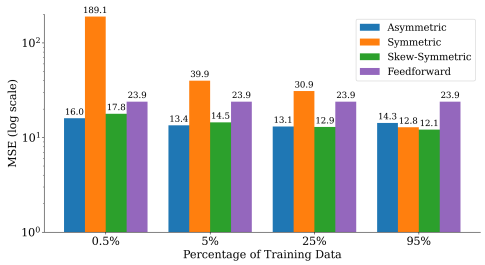



Figure 3: Test error obtained with different denoising networks.

(a) (b)

Figure 4: (a) Test images with increasing noise (b) reconstructions (skew-symmetric model).

## 4 Discussion

### 4.1 Related Works

Though typically applied in the context of feedforward models, Predictive coding (Whittington & Bogacz, 2017; Song et al., 2020) was recently applied to symmetric (Oliviers et al., 2025) and general asymmetric networks (Salvatori et al., 2022). Directed equilibrium propagation (Farinha et al., 2020; Costa & Santos, 2025) builds on (Scellier et al., 2018), exploring necessary conditions for stable inference in asymmetric Hopfield networks, as well as the alignment between the estimated gradient and the true gradient.

Inference in skew-symmetric Hopfield networks was explored in Goles (1986), where it was shown that parallel updates in a binary skew-symmetric Hopfield network leads to limit cycles. Xu et al. (1996) showed that binary Hopfield networks with general asymmetric connectivity converge to steady states, when certain conditions on the the diagonal elements are satisfied and serial updates are employed. More recently skew-symmetric RNNs have been applied to tasks requiring capturing long term dependencies in sequential data (Chang et al., 2019).

### 4.2 Limitations and Future Work

While the work presented here has implications for research in local learning, and novel hardware, it remains to be shown how these models can be scaled to more challenging tasks. In particular, the instability of the asymmetric model undermines some of its apparent advantages. Further theoretical work toward identifying stability criteria along the lines of Xu et al. (1996) and Chang et al. (2019) will allow for a more fair comparison with the other models. Likewise, the comparatively poor performance of the skew-symmetric model in the classification setting versus its superior performance in the denoising setting warrants further investigation. Finally, application of local learning algorithms to models of greater complexity with non-trivial activation functions (Krotov & Hopfield, 2016) is an extremely exciting direction.

### 4.3 Concluding Remarks and Broader Impact

We show that the Dyadic learning framework can be applied successfully to convolutional asymmetric and skew-symmetric Hopfield models on problems of moderate complexity, and demonstrate how these novel models benefit performance compared with the already well-characterized symmetric and feedforward models explored in prior works on equilibrium propagation and dual propagation. Though these results warrant further research, they remain highly relevant to the broader literature on physical and bio-inspired learning and suggest possible application in novel and less resource exhaustive approaches to machine learning, which remains an area of dire social and ecological importance.

## REFERENCES

Mohamed Akrout, Collin Wilson, Peter Humphreys, Timothy Lillicrap, and Douglas B Tweed. Deep learning without weight transport. *Advances in neural information processing systems*, 32, 2019.

Bo Chang, Minmin Chen, Eldad Haber, and Ed H Chi. Antisymmetricrnn: A dynamical system view on recurrent neural networks. *arXiv preprint arXiv:1902.09689*, 2019.

Pedro Costa and Pedro A Santos. Directed equilibrium propagation revisited. *Mathematics*, 13(11): 1866, 2025.

Georgios Detorakis, Travis Bartley, and Emre Neftci. Contrastive hebbian learning with random feedback weights. *Neural Networks*, 114:1–14, 2019.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Sankar Vinayak Elayedam and Gopalakrishnan Srinivasan. Scaling equilibrium propagation to deeper neural network architectures. *arXiv preprint arXiv:2509.26003*, 2025.

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.

Matilde Tristany Farinha, Sérgio Pequito, Pedro A Santos, and Mário AT Figueiredo. Equilibrium propagation for complete directed neural networks. *arXiv preprint arXiv:2006.08798*, 2020.

Eric Goles. Antisymmetrical neural networks. *Discrete Applied Mathematics*, 13(1):97–100, 1986.

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

Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2(3):5, 2022.

Rasmus Høier and Christopher Zach. Two tales of single-phase contrastive hebbian learning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 18470–18488. PMLR, 2024.

Rasmus Høier, D Staudt, and Christopher Zach. Dual propagation: Accelerating contrastive hebbian learning with dyadic neurons. In *International Conference on Machine Learning*, pp. 13141–13156. PMLR, 2023.

Rasmus Høier, Kirill Kalinin, Maxence Ernoult, and Christopher Zach. Dyadic learning in recurrent and feedforward models. In *NeurIPS 2024 Workshop Machine Learning with new Compute Paradigms*, 2024. URL `https://openreview.net/pdf?id=LNfWowAErI`.

Rasmus Kjær Høier. *Local Learning Rules for Deep Neural Networks with Two-State Neurons*. PhD thesis, Chalmers Tekniska Hogskola (Sweden), 2025.

Kirill P Kalinin, George Mourgias-Alexandris, Hitesh Ballani, Natalia G Berloff, James H Clegg, Daniel Cletheroe, Christos Gkantsidis, Istvan Haller, Vassily Lyutsarev, Francesca Parmigiani, et al. Analog iterative machine (aim): using light to solve quadratic optimization problems with mixed variables. *arXiv preprint arXiv:2304.12594*, 2023.

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.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. *Advances in neural information processing systems*, 29, 2016.

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.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random feedback weights support learning in deep neural networks. *arXiv preprint arXiv:1411.0247*, 2014.

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

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Gaspard Oliviers, Mufeng Tang, and Rafal Bogacz. Bidirectional predictive coding. *arXiv preprint arXiv:2505.23415*, 2025.

Tommaso Salvatori, Luca Pinchetti, Beren Millidge, Yuhang Song, Tianyi Bao, Rafal Bogacz, and Thomas Lukasiewicz. Learning on arbitrary graph topologies via predictive coding. *Advances in neural information processing systems*, 35:38232–38244, 2022.

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

Benjamin Scellier, Anirudh Goyal, Jonathan Binas, Thomas Mesnard, and Yoshua Bengio. Generalization of equilibrium propagation to vector field dynamics. *arXiv preprint arXiv:1808.04873*, 2018.

Benjamin Scellier, Maxence Ernoult, Jack Kendall, and Suhas Kumar. Energy-based learning algorithms for analog computing: a comparative study. *arXiv preprint arXiv:2312.15103*, 2023.

Benjamin Scellier, Maxence Ernoult, Jack Kendall, and Suhas Kumar. Energy-based learning algorithms for analog computing: a comparative study. *Advances in Neural Information Processing Systems*, 36, 2024.

Yuhang Song, Thomas Lukasiewicz, Zhenghua Xu, and Rafal Bogacz. Can the brain do backpropagation?—exact implementation of backpropagation in predictive coding networks. *Advances in neural information processing systems*, 33:22566–22579, 2020.

James CR Whittington and Rafal Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, 29(5): 1229–1262, 2017.

Xiaohui Xie and H Sebastian Seung. Equivalence of backpropagation and contrastive hebbian learning in a layered network. *Neural computation*, 15(2):441–454, 2003.

Zong-Ben Xu, Guo-Qing Hu, and Chung-Ping Kwong. Asymmetric hopfield-type networks: theory and applications. *Neural Networks*, 9(3):483–501, 1996.

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

# A Experimental Details

## A.1 Datasets

All simulations were run on CIFAR-10 (Krizhevsky, 2009) includes 60,000 color images of objects and animals. Images are split into 10 classes, with 6,000 images per class. Training data and test data include 50,000 images, and 10,000 images respectively.

### A.2 DATA PREPROCESSING

CIFAR-10 images were normalized using the statistics listed in 2, and augmented with random horizontal flips and random cropping at training time (but not at evaluation time).

Fashion MNIST images were normalized to the range [-1,1] and augmented with random horizontal flips during training. The Fashion-MNIST images were obscured with Gaussian noise during both validation and training. The standard deviation of the noise was drawn uniformly from the interval $[0, 1]$ for each image.

Table 2: Data Normalization. Input images were normalized by conventional mean ($\mu$) and standard deviation ($\sigma$) values for each dataset. All images used are color (three channels).

| Dataset | Mean ($\mu$) | Standard deviation ($\sigma$) |
|---|---|---|
| CIFAR-10 | (0.4914, 0.4822, 0.4465) | (0.2470, 0.2435, 0.2616) |

### A.3 SIMULATION DETAILS

**Weight initialization.** Initialization of weights follows the weighted Kaiming approach employed in (Scellier et al., 2023)

**State initialization.** All layers are initialized as zero matrices.

**Activation functions.** In the literature, activations (i.e. "clamping") is conventionally applied at each layer, *with the exception of the final layer*, where it is sometimes included e.g. Scellier et al. (2024), and sometimes omitted Laborieux et al. (2021), depending on the loss function at use. For these experiments we used both the standard hard activation employed by Ernoult et al. (2019) and Scellier et al. (2024), and, when specified , Rectified Linear Units. In the denoising experiments in particular images were normalized to [-1,1], so the reconstruction layers clamped to this interval rather than [0,1] which was used in all other layers.

**Architecture.**

**Classification setting** All convolutional layers used in classification experiments are of kernel size 3 and stride and padding 1. Channels per layer are as follows: [128,256,512,512,10]. Hard-sigmoid (0,1 clamp) was used on all but the output layer. We employed an initial learning rate of 0.0125 for all but the asymmetric model, which used an initial learning rate of 0.0022. Final learning rate was 2e-6. Batch size was set to 32, weight decay, 2.5e-4, decreased via cosine annealing. Feedforward models and symmetric models use an inference step size $\alpha = 1$, where as the skew-symmetric model uses 0.35 and the asymmetric model uses 0.7. For all layers except the final the "squeezing" method of Dinh et al. (2016) was used as a pooling surrogate, ensuring consistency across models (max-pool, for example, fails in the skew-symmetric setting). This operations reshapes a channels $\times$ height $\times$ width tensor into a $(4 \cdot \text{channels}) \times (\text{height}/2) \times (\text{width}/2)$.

**Denoising setting** The denoising network is composed of 3 convolutional hopfield layers (with 1, 32 and 64 filters respectively) followed by a Dense Hopfield layer with 512 units. The reason the first layer only has 1 filter is that it is used as the output layer and consequently it must have the same width height and depth as the input layer. All convolutional layers use 5x5 filters and stride 1 and zero padding 2. Feedforward models and symmetric models use an inference step size $\alpha = 1$, where as the skew-symmetric model uses 0.1 and the asymmetric model uses 0.5. All models used an initial learning rate of 0.01 and a final learning rate of $1e - 5$. The batch size was 64 and the weight decay was set to $3e - 3$.

**Other details.** All experiments were run using SGD with 0.9 momentum and Cosine Annealing. Code was implemented in Jax and simulations were run on diverse devices.