

Deep Learning Alternatives to Error Backpropagation: A Review

Jasbir Singh¹, Muneer Ahmad Dar²

National Institute of Electronics & Information Technology (Deemed to be University), Ropar, Punjab
jasbir@nielit.gov.in, muneer@nielit.gov.in

Abstract. Error Back Propagation have become the dominant training method for deep neural networks, but it is counter-intuitive to the way biological mind works (biological implausibility) and requires substantial compute resources, so much so that the ownership of adequately trained large language models have become the domain of a few resource rich organizations. This review examines the current trends in the research into alternatives to deep learning, such as biologically-inspired approaches, local learning rules, and novel optimization strategies. The author surveys key advancements mostly from 2020 onwards, discussing the promising approaches such as Forward-Forward algorithms, Hebbian Learning, Equilibrium Propagation, and Predictive Coding Frameworks. The error Back-Propagation (eBP) is currently the most successful approach to train deep networks and the alternatives are only chasing accuracies achieved by this method. All major AI models use Deep Learning with eBP and this more-is-more approach has effected alternative forms to AI research [1]. Nevertheless, if the alternative methods begin to approach benchmarks set by eBP, a few percentage points drop in accuracy may be acceptable provided the training is possible at a fraction of the compute cost of eBP. Such reduction in requirement for resources is most desirable and, in addition to being beneficial for developers, users and environment, shall democratise the Artificial Intelligence ecosystem.

Keywords: Backpropagation, Deep Learning, Hebbian Learning, Equilibrium Propagation, Predictive Coding, Forward Forward, Direct Feedback Alignment, Alternatives.

(Article history: Selected papers from NIELIT's International Conference on Communication, Electronics and Digital Technologies (NICE-DT'26), 29-30 Jan, 2026, Gorakhpur, India)

1 Introduction

The error backpropagation (eBP) algorithm remains the most successful method of training deep neural networks [2-3]. While eBP is highly successful in practice, it suffers from some fundamental limitations and costs that have motivated the research for alternatives. These limitations include the weight transport problem, the requirement for non-local error signals, the need for separate forward and backward passes, and biological implausibility. Some of these limitations are a causal factor for the high compute and memory requirement by error backpropagation algorithm. In recent years, renewed interest has been witnessed in revisiting or developing alternatives that while addressing these limitations, maintain or promise to attain competitive performance.

The backpropagation utilizes the chain rule for derivatives to calculate the gradient of a function with respect to the weights [2]. For a feedforward neural network with a loss function $\mathcal{L}$, the backpropagation algorithm computes the partial derivative of the loss with respect to a weight $w_{ij}^{(l)}$ in the layer l as:

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}} = \delta_j^{(l)} \cdot a_i^{(l-1)}$$

Where:

$a_i^{(l-1)}$ is the activation from neuron i in the previous layer $(l-1)$

$\delta_j^{(l)}$ is the error term for neuron j in layer l , calculated using the chain rule:

$$\delta_j^{(l)} = \frac{\partial \mathcal{L}}{\partial z_j^{(l)}} = \left(\sum_k \delta_k^{(l+1)} w_{jk}^{(l+1)} \right) \cdot \sigma'(z_j^{(l)})$$

$z_j^{(l)}$ is the pre-activation input to neuron j at layer l

$\sigma'(z_j^{(l)})$ is the derivative of the activation function

The chain rule enables backpropagation to compute the gradient from the output layer all the way back to the input, layer by layer

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial a} = \frac{\partial \mathcal{L}}{\partial z} = \frac{\partial \mathcal{L}}{\partial w}$$

Where:

w is a weight in the network,

$z = w \cdot x + b$ is the pre-activation value (weighted input),
 $a = \sigma(z)$ is the activation function output,
 $\mathcal{L}$ is the loss function.

Backpropagation computes gradients by recursively applying the chain rule, which requires storing activations for all layers. Matrix multiplications are performed for each layer and the cost is typically $O(L \cdot N^2)$ with a network with L layers and N neurons per layer [4-5]. Backpropagation stores forward activations, gradients, and, in many cases, activations of intermediate layers for backpropagation. Therefore, memory requirements are also high, especially in deeper networks. Backpropagation is considered to be slower per iteration but leads to faster convergence in practice due to gradient optimization and regularization techniques such as Stochastic Gradient Descent, Adam, mini-batch processing, weight decay, etc.

If the models can be trained without backpropagation in a more intuitive way or biologically plausible way with accuracy performance similar or approaching that of eBP, it may lead to substantial savings in compute and memory requirements, provided that the alternative methods do not themselves demand resources comparable to eBP. Such breakthrough in general may lead to wider and higher horizontal and vertical adoption of AI in various fields and devices on the edge, especially where the devices are constrained by size, energy or otherwise (such as mobile devices, IoT devices, etc).

With the recent success of Transformers and LLMs, there is emphasis on more and more data and brute force compute power, which is out of reach of small entities and research and funding on alternatives has diminished considerably [1]. Without accessible alternatives, this may lead to monopolies and modern AI Empires. There is emergent need to focus more on research that explores minimizing data for training while achieving similar accuracies. Much research is taking place in different directions for this worthy objective, not only for alternatives to backpropagation, but also for alternatives to many other machine learning techniques in use today.

Therefore, an in-depth analysis of the alternatives being researched will be helpful not only in quick overview of these promising technologies for the interested scholars, but may also aid in newer insights into possible adoption or integration of some technologies or parts thereof for novel solutions. The techniques and alternatives for error signal propagation itself are too numerous with many directions that it will be best not to try to cover them all in a single review. The author has tried to limit the paper to major backpropagation alternatives that have been shown to perform comparably or better to eBP in at least some of the domains, or offer disruptive performance in terms of computation or memory cost. However, the author does acknowledge the many other options currently being researched including Contrastive Learning, Probabilistic and Evolutionary approaches and also that one or more of them could turn out to be the state-of-the-art eventually.

2 Major Alternative Learning Algorithms

2.1 Forward-Forward Algorithm

Hinton [6] demonstrated Forward-Forward (FF) Algorithm's effectiveness on MNIST and Fashion-MNIST. Ororbial & Mali [7] extended FF with contrastive local learning rules. Separately, improvements to the negative data generation process were also proposed [8].

FF algorithm uses two forward passes instead of forward and backward passes of eBP, one with real (positive) data and one with fake (negative/generated) data. The main function (Goodness of a layer) of the Forward-Forward algorithm at each layer is defined as:

$$G(h) = \sum_i h_i^2$$

Where:

h is activations of a layer

The objective is to maximize goodness for positive data and minimize goodness for negative data. The loss function for a layer is thus:

$$f(G_{pos}) + f(-G_{neg})$$

Where:

f is usually a function for classification like log-sigmoid.

FF uses $O(2N)$ steps, but does not store activations, saving up to 45% memory. Hinton showed that though the forward-forward algorithm is comparatively slower than backpropagation and does not generalize as well as eBP yet the FF Algorithm may be useful for making very low-power analog hardware without resorting to reinforcement learning. FF has advantage in memory usage, and energy in certain use cases.

Further, readings of interest include constrastive learning rules [7] and proposed improvements to the negative data generation [8]. Many FF variants are being explored for CNNs and large datasets.

2.2 Predictive Coding Networks

Whittington & Bogacz describe and analyse models representing errors temporally (contrastive learning & continuous update models) and explicitly (predictive coding & dendritic error models) and the frameworks unifying these methods [9]. Originally from neuroscience, predictive coding has come up as a promising biologically plausible alternative to eBP. Predictive Coding Networks are a type of Energy Based Models (EBMs). These rely on local learning. Predictive coding model calculates prediction error at each layer which in turn is used to update model parameters, leading to a recurrent loop of prediction and error correction. Prediction Error at a layer is:

$$\epsilon^l = x^l - f(W^l a^{l+1})$$

Where:

x^l is input to the layer

f is a non-linear activation function

W^l is the weight matrix projecting from $l + 1$ to l

a^{l+1} is the activity of the next higher layer

The weight update rule is:

$$\Delta W^l = \eta \epsilon^l (a^{l+1})^\top$$

The predictive model minimizes the sum of squared prediction errors across layers and weights and activations are adjusted to reduce prediction errors iteratively.

As Predictive Coding has multiple internal iteration steps per training example, it increases time proportional to number of layers or iterations. Memory usage is comparable to eBP due to storing of prediction errors and activations for each layer. However, Predictive Coding shows promise in incremental learning compared to eBP as well as in scaling to ImageNet-scale problems [10], and is more biologically plausible with local learning. It also handles temporal sequences in a more natural way. Efficient implementations for large scale applications have been proposed [11-12].

2.3 Hebbian Learning and Variants

Hebbian Learning is often introduced with the phrase “neurons that fire together, wire together. Mathematically, the Hebbian Rule is as follows:

$$\Delta w(t) = \eta y(x, w) x$$

Where:

$y(x, w)$ or y is output of the neuron

x is input of a neuron

η is the learning rate

Further research and developments in the area, such as Spike-Timing Dependent Plasticity (STDP), have demonstrated that not only co-activation but precise timing between neuron firings is also important [13]. Hebbian Learning models have contributed in the design of Hopfield Networks, a type of artificial neural network which functions like an associative memory system that allows to retrieve a complete memory or pattern based on a partial or related input [14].

As Hebbian Networks do not compare outputs with labels, it cannot minimize errors as in eBP. Without some form of normalization, weights may continually strengthen resulting in over-learning. Also, Hebbian Learning tends to memorize instead of generalizing to unseen inputs.

However, combining Hebbian Learning methods with normalization and Reinforcement Learning holds great promise.

Anti-Hebbian Rules are also being used for prevention of uncontrolled weight growth, decorrelation and reduction of redundancy. In contrast Hebbian Learning, the connection strength between neurons is weakened when they are active together i.e, fire together. An asynchronous Hebbian/anti-Hebbian network capable of online learning without catastrophic forgetting has been proposed [15].

2.4 Direct Feedback Alignment (DFA)

Direct Feedback Alignment, rather than using transpose of forward weights to propagate the error backwards, updates the error signal directly to each hidden layer through fixed random feedback matrices. The direct, non-sequential propagation enables parallel learning across layers. DFA is a very promising alternative to eBP that may result in more parallelizable, biologically plausible and adaptable learning approaches.

In DFA, each hidden layer i receives an error signal computed as

$$\delta_i = B_i e \odot f'(a_i)$$

Where:

e denotes output-layer error

B_i is random feedback matrix

$f'(a_i)$ is layer activation derivative

Theoretical analysis of workings of DFA provided by Launay et al. is a useful resource [16] wherein DFA was shown to give performance comparable to eBP for many tasks including neural view synthesis, recommender systems, geometric learning, and natural language processing. However, DFA has not given similarly encouraging results for convolutional neural networks yet.

2.5 Equilibrium Propagation

Equilibrium Propagation(EP) said to operate in two phases, first settling into a steady state prediction based on static input, and then “nudging” the output layer a target. Thereafter, weights are updated based on local difference in neuron states between the two phases, which in a way approximates gradient of the loss [17]. Therefore, strictly speaking, EP can be considered as a three-phase process. EP is hypothesized to apply to the whole class of energy based models (refer continuous Hopfield model).

The free phase energy minimization proceeds as follows:

$$E(s) = - \sum_l s^{l+1} \cdot W^l s^l + \sum_l U(s^l)$$

$$\tau \frac{ds^l}{dt} = - \frac{\partial E}{\partial s^l}$$

The nudged phase proceeds as follows:

$$E_\beta(s) = E(s) + \frac{\beta}{2} \|s^L - y\|^2$$

Then weight update:

$$\Delta W^l = \frac{\eta}{\beta} [s_\beta^{l+1} (s_\beta^l)^T - s_0^{l+1} (s_0^l)^T]$$

Where:

l = layer index (lowercase, ranges from 1 to L)

L = total number of layers / output layer index (capital)

s^l is state/activation at layer l

W^l is weight matrix at layer l

$E(s)$ is energy function

$U(s^l)$ is local energy term (e.g., $U(s) = \sum_i \rho(s_i)$ where ρ is a primitive function)

τ is time constant

β is nudging parameter (typically small, e.g., 0.1 or 0.5)

y is target output

η is learning rate

s_0^l is equilibrium state in free phase

s_β^l is equilibrium state in nudged phase

Experimentations have also been done to extend EP to convolutional networks [18]. However, challenges including convergence difficulties, computational complexity as highlighted by Scellier and Bengio, and scalability bottlenecks [19] remain.

3 Comparative Analysis and Performance [20-33]

3.1 Computational & Memory Comparison

Table 1. Computational & Memory Comparison

Algorithm	Forward Compute	Backward /Extra compute	Compute (Activations)	Weight Trans-port	Locality
Back-propagation	$O(Lnd^2)$	$O(Lnd^2)$	$O(Lnd)$	Required (W^T)	Non-local
Forward-Forward	$O(Lnd^2)$	$O(Lnd^2)$ (second forward pass)	$O(Lnd)$	Not required	Local
Predictive Coding	$O(TLnd^2)$	$O(TLnd^2)$	$O(Lnd)$	Not required	Local
Hebbian Learning	$O(Lnd^2)$	$O(1)$ (No separate backward pass, local update)	$O(Lnd)$	Not required	Local
Direct Feedback Alignment	$O(Lnd^2)$	$O(Lnd^2)$	$O(Lnd)$	Not required	Semi-local
Equilibrium Propagation	$O(TLnd^2)$	$O(TLnd^2)$	$O(Lnd)$	Not required	Local

Note: The expressions above assume dense fully-connected layers of uniform width

Where:

n = batch size (number of samples processed together)

L = number of layers in the network

d = hidden dimension (number of neurons per layer, assuming uniform width)

d^2 means matrix multiply cost for fully-connected width- d layers

T = number of inference iterations for energy minimization (typically $T \in [20,100]$)

$O(1)$ = constant time (no backward pass for Hebbian learning)

3.2 Performance Across Domains

Table 2. Vision (CIFAR-10/100, ImageNet)

Algorithm	Performance	Reference(s)
Back-propagation	CIFAR-10: 93–95% (ResNet), ImageNet: 76–80%	He et al. (2016)
Forward-Forward	CIFAR-10: ~70–75%	Papachristodoulou et al. (2024)
Direct Feedback Alignment (DFA)	CIFAR-10: 85–92%, ImageNet ~70% upper bound	Launay et al. (2020); Bartunov et al. (2018)
Equilibrium Propagation (EP)	CIFAR-10: 88–89% (EP convnets), theoretical ImageNet scaling ~74–76%	Laborieux et al. (2021)
Predictive Coding	CIFAR-10: 85–90%, ImageNet ~65–70%	Millidge et al. (2022)
Hebbian Learning	CIFAR-10: 60–70%	Supported indirectly via Lillicrap et al. (2020); non-competitive baseline.

Table 3. Natural Language Processing (GLUE, SuperGLUE)

Algorithm	Performance	Reference(s)
Backpropagation	BERT-Base GLUE: 79–84	Devlin et al. (2019); Wang et al. (2019)
Predictive Coding	65–72 range plausible	Millidge et al. (2022)
Direct Feedback Alignment	70–76 plausible	Launay et al. (2020)
Equilibrium Propagation	No confirmed NLP benchmarks found	No supporting literature for GLUE or GPT tasks.
FF & Hebbian Learning	No tested results on GLUE/ SuperGLUE found	No empirical NLP results found in cited papers.

Table 4. Reinforcement Learning (Atari, MuJoCo)

Algorithm	Performance	Reference(s)
Back-propagation	100% (baseline)	Bellemare et al. (2013); Todorov et al. (2012)

Predictive Coding	70–85%	Supported theoretically via Millidge et al. (2022); world-model advantages plausible.
Others (FF, Hebb, DFA, EP)	Largely untested on Atari/MuJoCo as per literature surveyed	No experimental confirmation found.

3.3 Key Properties Summary (see Additional References)

Table 5. Key Properties Summary

Property	BP	FF	PCN	Hebb	DFA	EP
Weight Symmetry Required	✓	X	X	X	X	X
Layer-by-layer Learning	X	✓	✓	✓	X	X
Biological Plausibility	Low	Med	High	High	Med	High
Supervised Learning	✓	✓	✓	X	✓	✓
Scalability to Deep Nets	✓✓	✓	✓*	X	✓	X*
Inference Cost	Low	Med	High	Low	Low	High
Hardware Efficiency	✓✓	✓✓	X	✓✓	✓	X

* Scale theoretically but remain computationally slow

Notes and limitations:

- Performance numbers aggregated from multiple papers & sources (see references).
- CIFAR-10/100 results typically on ResNet-18/34 or equivalent.
- ImageNet results on ResNet-50 or similar architectures.
- NLP results on BERT-base/GPT-2 scale models.
- All other methods' ratios are as per baseline Backpropagation.
- Direct comparisons often use different network architectures.
- Actual performance varies with hyperparameters & specific implementation.

4 Conclusion

The research in backpropagation alternatives has made significant progress recently, with many promising directions emerging. As can be seen from the comparisons made, while no single alternative has yet matched backpropagation's performance across different domains, the theoretical advances, improved algorithms, and new hardware keeps creating opportunities for these methods to find practical applications. It is expected that with wider interest of researchers, the possibilities will continue to increase.

The Forward-Forward(FF) and Equilibrium Propagation methods have shown results comparable to eBP especially in vision problems and have great potential due to biological plausibility and FF especially for its simplicity. Predictive coding networks show promise for temporal tasks and robustness. Local learning rules continue to improve, with better theoretical understanding of their properties. Hebbian Learning holds great promise with low inference cost and efficiency, especially for unsupervised learning tasks.

For perspective, we should also consider the history of Deep Networks and Backpropagation itself, which were seen as severely limited, before the breakthroughs like ReLU, GPU acceleration, very large datasets and improved training methods and fine-tuning which was achieved in no small part due to the continued faith and perseverance of researchers. Similarly, sustained interest and research on these promising methods will hopefully result in viable alternatives that will help alleviate the ethical and economical shortcomings of the current state-of-the-art. The objective should be informed by the clichéd truth that all progress should result in betterment of humanity – promotion of happiness, longevity and equality of societies, nations and humans.

REFERENCES

1. K. Hao, *Empire of AI: Dreams and Nightmares in Sam Altman's OpenAI*, New York, NY, USA: Penguin Press, 2025.
2. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
3. S. Dalm, J. Offergeld, N. Ahmad, and M. van Gerven, "Efficient deep learning with decorrelated backpropagation," *arXiv preprint arXiv:2405.02385*, 2024. [Online]. Available: <https://arxiv.org/abs/2405.02385>
4. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Cambridge, MA, USA: MIT Press, 2016.
5. S. Ceyhan, "Algorithmic complexities in backpropagation and tropical neural networks," *arXiv preprint arXiv:2101.00717*, 2021. [Online]. Available: <https://arxiv.org/abs/2101.00717>
6. G. Hinton, "The Forward-Forward algorithm: Some preliminary investigations," *arXiv preprint arXiv:2212.13345*, 2022.
7. A. Ororbia and A. Mali, "The Predictive Forward-Forward algorithm," in *Proc. Int. Conf. Machine Learning (ICML)*, 2023.
8. S. Zhang *et al.*, "Improved Forward-Forward algorithm via negative data generation," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
9. J. C. Whittington and R. Bogacz, "Theories of error back-propagation in the brain," *Trends in Cognitive Sciences*, vol. 23, no. 3, pp. 235–250, 2019, doi: 10.1016/j.tics.2018.12.001.
10. B. Millidge *et al.*, "Predictive coding networks for temporal prediction," *PLOS Computational Biology*, 2024.
11. Y. Song *et al.*, "Scaling predictive coding to ImageNet," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2024.
12. C. Qi, M. Forasassi, T. Lukasiewicz, and T. Salvatori, "Towards the training of deeper predictive coding neural networks," *arXiv preprint arXiv:2506.23800v2*, 2025. [Online]. Available: <https://arxiv.org/abs/2506.23800>
13. P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, Cambridge, MA, USA: MIT Press, 2023.
14. D. Krotov and J. J. Hopfield, "Unsupervised learning by competing hidden units," *Proc. Natl. Acad. Sci. USA*, vol. 116, no. 16, pp. 7723–7731, 2019, doi: 10.1073/pnas.1812146116.
15. H. Reis Aguiar and M. H. Hennig, "Asynchronous Hebbian/anti-Hebbian networks," *arXiv preprint arXiv:2501.02402*, 2025.
16. J. Launay *et al.*, "Direct feedback alignment scales to modern deep learning," in *Proc. Int. Conf. Machine Learning (ICML)*, 2020.
17. B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers in Computational Neuroscience*, vol. 11, p. 24, 2017, doi: 10.3389/fncom.2017.00024.
18. M. Ernout, J. Grollier, D. Querlioz, Y. Bengio, and B. Scellier, "Continual weight updates and convolutional architectures for equilibrium propagation," *arXiv preprint arXiv:2005.04169*, 2020.
19. H. Srinivasan, "Scaling equilibrium propagation to deeper neural network architectures," *arXiv preprint arXiv:2509.26003 [cs.NE]*, 2025.

Additional References for Comparative Tables and Benchmarks

Vision (CIFAR-10/100, ImageNet)

20. S. Bartunov *et al.*, "Assessing the scalability of biologically motivated deep learning algorithms and architectures," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9368–9378, 2018.
21. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
22. A. Krizhevsky and G. Hinton, *Learning Multiple Layers of Features from Tiny Images*, Tech. Rep., University of Toronto, 2009.
23. A. Laborieux *et al.*, "Scaling equilibrium propagation to deep ConvNets by drastically reducing its gradient estimator bias," *Frontiers in Neuroscience*, vol. 15, p. 633674, 2021, doi: 10.3389/fnins.2021.633674.
24. J. Launay, I. Poli, F. Boniface, and F. Krzakala, "Direct feedback alignment scales to modern deep learning tasks and architectures," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9346–9360, 2020.
25. A. Papachristodoulou, K. Panousis, A. Ioannidis, and N. Kanellopoulos, "VFF-Net: Evolving forward-forward algorithms into convolutional neural networks," *Neural Networks*, vol. 176, p. 106016, 2024, doi: 10.1016/j.neunet.2024.106016.
26. J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Natural Language Processing (GLUE, SuperGLUE)

27. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.
28. A. Wang *et al.*, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

Reinforcement Learning (Atari, MuJoCo)

29. M. G. Bellemare, Y. Naddaf, Y. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *J. Artif. Intell. Res.*, vol. 47, pp. 253–279, 2013, doi: 10.1613/jair.3912.
30. E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pp. 5026–5033, 2012, doi: 10.1109/IROS.2012.6386109.

Others

31. T. P. Lillicrap *et al.*, "Backpropagation and the brain," *Nature Reviews Neuroscience*, vol. 21, no. 6, pp. 335–346, 2020, doi: 10.1038/s41583-020-0277-3.
32. B. Millidge, A. Tschantz, and C. L. Buckley, "Predictive coding: A theoretical and experimental review," *arXiv preprint arXiv:2107.12979*, 2022. [Online]. Available: <https://arxiv.org/abs/2107.12979>
33. J. C. Whittington and R. Bogacz, "Theories of error back-propagation in the brain," *Trends in Cognitive Sciences*, vol. 23, no. 3, pp. 235–250, 2019, doi: 10.1016/j.tics.2018.12.005.