Hybrid Equilibrium Propagation: Trading Estimator Bias for Compute

Stefan Stoian

Supervisor(s): Stephanie Tan, Yaqi Guo

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

June 21, 2026

Name of the student: Stefan Stoian
Final project course: CSE3000 Research Project
Thesis committee: Stephanie Tan, Yaqi Guo, Inald Lagendijk

Hybrid Equilibrium Propagation: Trading Estimator Bias for Compute

Stefan Stoian
Faculty Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Van Mourik Broekmanweg 6, 2628 XE Delft

Abstract

Equilibrium Propagation (EP) is a backpropagation-free learning algorithm for energy-based networks; its standard estimator computes the gradient by comparing the equilibrium states reached in a free phase and a single nudged phase, but carries a bias that limits how closely EP can match backpropagation. The centered estimator reduces this bias and improves accuracy, but adds a second nudged phase per update, raising the training cost. To balance accuracy against compute, we introduce *hybrid* EP, a family of estimators that mix the standard and centered updates on a per-batch basis, and show analytically that the mixing probability controls this bias, so that annealing it interpolates between the two regimes. We evaluate three hybrid schedules - a cosine anneal, its inverse, and a fixed stochastic mix - against standard and centered EP on MNIST, Fashion-MNIST, and CIFAR-10, in order of increasing complexity. On the easier tasks the hybrids match centered EP at lower compute. On CIFAR-10 standard EP collapses to near-chance accuracy, and the cosine and inverse schedules collapse with it: each concentrates its biased updates into one long stretch, whereas only the stochastic mix, which spreads the same biased updates evenly across batches, trains stably. The compute savings of hybrid EP are therefore real but task-dependent: they are realized most cleanly when standard EP is itself viable, and training stability is governed not by the number of biased updates but by their distribution over the course of training.

1 Introduction

Backpropagation has been the foundation of deep learning for decades. Despite its success, it has several limitations [1]. First, training algorithms based on backpropagation are difficult to transfer directly to physical systems, because automatic differentiation, the computational backbone of backpropagation, cannot be implemented directly in physical hardware. Second, backpropagation relies on a biologically implausible constraint [1]. Third, it is energy-intensive; as the energy cost of training large models continues to grow while Moore's law approaches its limits [2], these costs are becoming increasingly difficult to sustain.

This has driven growing interest in backpropagation-free learning. Several methods have been proposed as alternatives to backpropagation. Recently, FFZero [2] demonstrated a forward-only framework that can be implemented in simulated photonic systems.

This thesis focuses on Equilibrium Propagation (EP), a backpropagation-free learning method for energy-based neural networks introduced by Scellier and Bengio [3]. EP trains a network by letting its internal state relax to equilibrium in two phases. In the first phase, called the free phase, the network settles naturally into an equilibrium using only the input, with no influence from the target. In the second phase, called the nudged phase, the output is slightly pushed toward the correct target. The learning signal is then obtained by comparing the two equilibrium states: the difference between them indicates how each weight should change to bring the output closer to the target. This comparison yields an estimate of the same gradient that backpropagation would compute, but without the separate backward pass that backpropagation requires.

Recent work has shown that the choice of EP gradient estimator is itself critical. The original two-phase estimator of Scellier and Bengio [3] - the procedure described above - is the one we refer to as standard EP, or std, throughout the paper. Laborieux et al. [4] showed that the gradient std produces is systematically inaccurate in a way that can hinder performance and scaling, and proposed a more accurate variant, which we refer to as centered EP, or cep. The greater accuracy of centered EP, however, comes at a higher computational cost, placing the two estimators at opposite ends of a trade-off between gradient accuracy and training cost.

This additional cost comes from an extra phase. Standard EP requires one free phase and one nudged phase, whereas centered EP adds a second nudged phase that nudges the output in the opposite direction - a cost acknowledged as one of its practical drawbacks [4]. Each centered update therefore buys a more accurate gradient at the price of an extra nudged phase.

Computational cost is a particularly relevant concern for EP. Although one motivation for moving beyond backpropagation is the growing energy cost of training, on standard digital hardware EP does not improve on it: Laborieux et al. report that their accurate, three-phase centered EP runs about 20% slower than backpropagation [4]. This finding underscores that training cost should remain an important aspect of the EP discussion, even though EP is ultimately intended for neuromorphic hardware. One of the factors driving this cost is the number of phases that are performed. A phase is not a single step: it runs the network's dynamics to convergence over many sequential steps [3]. Every additional phase repeats this relaxation, and therefore adds directly to the cost of training.

This trade-off between gradient accuracy and computational cost raises the question of whether centered EP is necessary at every training update. The motivation for this question is that the usefulness of centered EP may vary during training. If standard EP and centered EP produce similar update directions during some stages of training, then standard EP may be sufficient during those stages, while requiring fewer nudged phases. Conversely, when the update directions produced by standard EP and centered EP differ substantially, centered EP may provide a more reliable learning signal.

To investigate this possibility, this thesis introduces hybrid Equilibrium Propagation, a family of estimators that combines standard EP and centered EP on a per-batch basis. At each update, the hybrid estimator uses standard EP with probability p and centered EP with probability $1 - p$. By design, this makes it possible to interpolate between the lower cost of standard EP and the improved gradient estimate of centered EP.

The central research question is therefore: under what conditions can hybrid Equilibrium Propagation reduce the computational cost of centered EP while preserving training stability and classification accuracy?

We answer this question by comparing standard EP, centered EP, and the proposed hybrid EP strategies on MNIST [5], Fashion-MNIST [6], and CIFAR-10 [7]. The comparison focuses on final test accuracy, and computational cost measured by the number of nudged phases. Together, these metrics allow us to evaluate whether hybrid EP can provide a practical middle ground between the lower cost of standard EP and the greater stability and accuracy of centered EP, while also giving insight into how estimator bias affects training.

The thesis is structured as follows: Section 2 introduces the theoretical background of EP and derives the bias of the standard, centered, and hybrid estimators, showing how the mixing probability controls the first-order bias term. Section 3 describes the experimental methodology: it defines the hybrid family of estimators

and the MNIST, Fashion-MNIST, and CIFAR-10 experiments. Section 4 evaluates the strategies on these three datasets of increasing difficulty and synthesizes the comparison. Section 5 summarizes the findings and discusses directions for future work.

2 Background

This section makes precise the trade-off that motivates hybrid EP. We begin by reviewing the equilibrium-propagation (EP) gradient estimator, state the bias of its standard one-sided and centered (symmetric) forms, and then derive the bias of the hybrid estimator that mixes the two. The resulting bias scales as $\mathcal{O}(p\beta + \beta^2)$, where p is the probability of using standard EP and β is the nudging strength. This expression has two regimes: when $p > \beta$, the leading bias is controlled by the first-order term $\mathcal{O}(p\beta)$; when $p \leq \beta$, the first-order contribution is no longer dominant and the estimator reaches the centered-EP bias floor $\mathcal{O}(\beta^2)$.

2.1 The EP gradient estimator

Let θ denote the trainable parameters and s the neuron state of an energy-based network with input x. We adopt the discrete-time formulation of Ernoult et al. [8], also used by Laborieux et al. [4], in which the dynamics are governed by a scalar primitive function $\Phi(\theta, s)$ and the network relaxes to an equilibrium that is a stationary point of Φ. In the *free phase* the network settles to the free equilibrium s_0 from the input alone. In a *nudged phase* with nudging strength β an additional term $-\beta\,\ell(s, y)$ is added to the primitive, where ℓ is the output cost with respect to the target y; the network then relaxes to a nudged equilibrium s_β. Intuitively, β controls how strongly the output is pulled toward the target.

The EP estimator of the loss gradient compares the parameter-gradient of the primitive at two equilibria. It is convenient to define

$$G(\beta) \;:=\; \nabla_\theta \Phi\big(\theta, s_\beta\big), \tag{1}$$

so that $G(0)$ is the free-phase term and $G(\beta)$ a nudged-phase term. The fundamental property of EP [3] is that the true loss gradient at the free equilibrium (equivalently, the gradient obtained by backpropagation through the converged free phase) is the derivative of G at the origin,

$$g^\star \;:=\; \lim_{\beta \to 0} \frac{G(\beta) - G(0)}{\beta} \;=\; G'(0). \tag{2}$$

Any practical estimator uses a finite β and therefore differs from $g^\star$; we call this difference the *bias* of the estimator. Throughout we assume G is smooth in a neighbourhood of $\beta = 0$ and expand it as

$$G(\beta) = G(0) + \beta\, G'(0) + \tfrac{\beta^2}{2}\, G''(0) + \tfrac{\beta^3}{6}\, G'''(0) + \mathcal{O}(\beta^4). \tag{3}$$

2.2 Bias of the standard and centered estimators

The bias of these two estimators was characterized by Laborieux et al. [4]; we restate the scalings here for completeness, as they are the basis for the hybrid analysis that follows.

Standard EP (one-sided). The standard estimator uses a single nudged phase at $+\beta$,

$$\hat{g}_{\mathrm{std}}(\beta) \;=\; \frac{G(\beta) - G(0)}{\beta}. \tag{4}$$

Substituting the expansion (3) gives

$$\hat{g}_{\mathrm{std}}(\beta) = G'(0) + \frac{\beta}{2}\, G''(0) + \frac{\beta^2}{6}\, G'''(0) + \mathcal{O}(\beta^3), \tag{5}$$

so that, subtracting $g^\star = G'(0)$, the bias is

$$\mathrm{bias}_{\mathrm{std}}(\beta) \;=\; \hat{g}_{\mathrm{std}}(\beta) - g^\star \;=\; \frac{\beta}{2}\,G''(0) \;+\; \underbrace{\frac{\beta^2}{6}\,G'''(0) + \mathcal{O}(\beta^3)}_{\mathcal{O}(\beta^2)}. \tag{6}$$

Every term past the first is of order β^2 or higher, so they collapse into a single $\mathcal{O}(\beta^2)$ remainder, leaving

$$\boxed{\mathrm{bias}_{\mathrm{std}}(\beta) \;=\; \frac{\beta}{2}\,G''(0) + \mathcal{O}(\beta^2) \;=\; \mathcal{O}(\beta).} \tag{7}$$

The leading error is *first order* in β, carried by the curvature term $G''(0)$.

Centered EP (symmetric). The centered estimator of Laborieux et al. [4] adds a third phase at $-\beta$ and replaces the free-phase reference by the negatively-nudged equilibrium,

$$\hat{g}_{\mathrm{cep}}(\beta) \;=\; \frac{G(\beta) - G(-\beta)}{2\beta}. \tag{8}$$

The even-order terms of (3) cancel in the symmetric difference,

$$G(\beta) - G(-\beta) = 2\beta\,G'(0) + \frac{\beta^3}{3}\,G'''(0) + \mathcal{O}(\beta^5), \tag{9}$$

which yields $\hat{g}_{\mathrm{cep}}(\beta) = G'(0) + \frac{\beta^2}{6}G'''(0) + \mathcal{O}(\beta^4)$ and hence

$$\boxed{\mathrm{bias}_{\mathrm{cep}}(\beta) \;=\; \hat{g}_{\mathrm{cep}}(\beta) - g^\star \;=\; \frac{\beta^2}{6}\,G'''(0) + \mathcal{O}(\beta^4) \;=\; \mathcal{O}(\beta^2).} \tag{10}$$

The first-order term has vanished, leaving a *second-order* bias. This is the sense in which centered EP is the more accurate estimator - at the cost of a second nudged phase per update.

2.3 Bias of the hybrid estimator

The central contribution of this work is the *hybrid estimator*, which we introduce here as a new way to navigate the accuracy-cost trade-off between standard and centered EP. Rather than modifying the finite-difference rules of either estimator, it leaves both untouched and instead chooses between them on each minibatch: with probability p it applies the standard estimator and with probability $1 - p$ the centered estimator. The realized update is therefore a random variable, and we characterize it through its expectation over this Bernoulli choice,

$$\mathbb{E}\big[\hat{g}_{\mathrm{hyb}}\big] \;=\; p\,\hat{g}_{\mathrm{std}}(\beta) + (1-p)\,\hat{g}_{\mathrm{cep}}(\beta). \tag{11}$$

Because the bias is linear in the estimator, the bias of the expected hybrid update is the same convex combination of the two biases. Using (7) and (10),

$$\begin{aligned}
\mathrm{bias}_{\mathrm{hyb}}(\beta, p) &= \mathbb{E}\big[\hat{g}_{\mathrm{hyb}}\big] - g^\star \\
&= p\,\mathrm{bias}_{\mathrm{std}}(\beta) + (1-p)\,\mathrm{bias}_{\mathrm{cep}}(\beta) \\
&= p\,\frac{\beta}{2}\,G''(0) \;+\; \big[p + (1-p)\big]\,\frac{\beta^2}{6}\,G'''(0) \;+\; \mathcal{O}(\beta^3) \\
&= \underbrace{p\,\frac{\beta}{2}\,G''(0)}_{\mathcal{O}(p\beta)} \;+\; \underbrace{\frac{\beta^2}{6}\,G'''(0)}_{\mathcal{O}(\beta^2)} + \mathcal{O}(\beta^3).
\end{aligned} \tag{12}$$

Crucially, the second-order terms of the standard and centered components combine into a single p-independent contribution: the β^2 coefficient is $p + (1 - p) = 1$. The hybrid bias is therefore the sum of a p-controlled first-order term and a fixed second-order term equal to the centered-EP bias - this second term acts as a *floor*: reducing p cannot push the bias below $\mathcal{O}(\beta^2)$, because that is the irreducible error of the centered estimator itself. Its order is governed by which of the two dominates,

$$\boxed{\mathrm{bias}_{\mathrm{hyb}}(\beta, p) \;=\; \mathcal{O}\big(p\beta + \beta^2\big) \;=\; \begin{cases} \mathcal{O}(p\beta), & p > \beta, \\ \mathcal{O}(\beta^2), & p \le \beta. \end{cases}} \tag{13}$$

Two limits confirm the interpretation. When $p = 1$ the hybrid reduces to standard EP and (12) recovers the first-order bias $\frac{\beta}{2}G''(0)$ of (7); when $p = 0$ the first-order term vanishes and the bias drops to the $\mathcal{O}(\beta^2)$ regime of centered EP.

3 Methodology

3.1 The hybrid family of estimators

The standard (std) and centered (cep) estimators of Section 2 sit at the two ends of an accuracy-cost trade-off. Hybrid EP populates the space between them: on each minibatch it applies the std update with probability p_{std} and the cep update otherwise (Eq. (11)), and a particular member of the family is fixed by choosing how p_{std} is set over training. We compare the two base estimators against three hybrid schedules, which differ only in how the cheaper std updates are distributed across training. All five strategies, used throughout the paper, are summarized here:

- **Standard EP (**std**)**: one free phase and one $+\beta$ nudged phase per update; the cheapest estimator, with first-order bias (Eq. (7)).
- **Centered EP (**cep**)**: adds a second nudged phase at $-\beta$, doubling the nudged-phase cost of std while cancelling the first-order bias (Eq. (10)).
- **Cosine hybrid**: $p_{\mathrm{std}}(e)$ decays from 1 toward 0 along a cosine, so training is std-dominated early and cep-dominated late.
- **Inverse cosine hybrid**: the mirror schedule, with $p_{\mathrm{std}}(e)$ rising from 0 to 1; cep-dominated early and std-dominated late.
- **Stochastic hybrid**: p_{std} held fixed at 0.5, with an independent Bernoulli draw per minibatch, so the two estimators are interleaved at the finest available granularity.

The cosine schedule is our primary hybrid, motivated by a diagnostic experiment (Appendix B) showing that the std estimate tracks the true gradient about as well as cep early in training but drifts away from it later; it therefore spends the cheap std budget early and reserves cep for late training. The cosine profile itself is borrowed from the learning-rate annealing of Loshchilov and Hutter [9], here applied to the std selection probability rather than the learning rate:

$$p_{\mathrm{std}}(e) \;=\; \frac{1}{2}\left(1 + \cos\frac{\pi e}{E}\right). \tag{14}$$

The inverse cosine hybrid runs this schedule backwards,

$$p_{\mathrm{std}}(e) \;=\; \frac{1}{2}\left(1 - \cos\frac{\pi e}{E}\right), \tag{15}$$

reversing the order in which the two estimators are used. Because it spends the same estimators in the same overall proportion as the cosine hybrid but in the opposite order, contrasting the two isolates the effect of *when* each estimator is applied, independently of how often. The stochastic hybrid holds $p_{\mathrm{std}} = 0.5$ throughout.

3.2 Strategy Comparison on MNIST

We evaluate all five strategies of Section 3.1 - std, cep, and the three hybrids (cosine, inverse cosine, and stochastic) - on MNIST, using the single-hidden-layer MLP configuration of Table 4 (Appendix A). Contrasting the cosine schedule with its mirror image, the inverse cosine schedule, isolates the effect of the schedule's *direction* - whether the cheap estimator is better spent early or late - while the stochastic schedule probes what happens when the two estimators are interleaved rather than run in long one-sided phases.

For each strategy we track two quantities. The first is final test accuracy: the classification accuracy on the test split defined by the dataset - for MNIST, the standard set of $10{,}000$ held-out images - measured over all test samples after training. The second is the cumulative nudged-phase count, which we use as a hardware-agnostic measure of compute cost. As argued in Section 1, the number of phases run over training is a good indicator of its cost. Since every strategy performs exactly one free phase per update and differs only in its nudged phases - one for a standard update, two for a centered update - counting nudged phases isolates the cost difference between strategies. Together these two quantities enable an accuracy-versus-cost comparison. Each configuration is repeated across 5 seeds; we report the mean and standard deviation.

3.3 Generalization to Fashion-MNIST

To assess robustness under domain shift while holding the input structure fixed (28×28 grayscale images), we repeat the comparison on Fashion-MNIST [6]. We treat Fashion-MNIST as a lightweight generalization check rather than a full family comparison: the aim is only to confirm that the headline cosine schedule still works under a domain shift, so we evaluate std, cep, and the cosine schedule, and leave the comparative analysis of schedule ordering (the inverse cosine and stochastic schedules) to MNIST and the harder CIFAR-10 model.

We adopt the two-hidden-layer MLP architecture and EP dynamics of Kubo [10], who report a centered-EP baseline on this dataset (Table 4, Appendix A). Following that work, we use $\tanh$ rather than a hard sigmoid, which was reported to cause unstable training on this dataset. We deliberately retain the optimization protocol from our MNIST experiment, rather than the per-layer learning rates used by Kubo: keeping the training protocol, the set of strategies, and the 5-seed protocol fixed across datasets ensures that the strategies are compared under identical optimization conditions, so that any differences in their relative performance can be attributed to the strategies themselves rather than to dataset-specific optimizer tuning. As on MNIST, each strategy is assessed by its final test accuracy and cumulative nudged-phase count, and every configuration is repeated over 5 seeds, for which we report the mean and standard deviation.

3.4 Scaling to CIFAR-10

Finally, we test the approach under substantially increased visual complexity using CIFAR-10 [7] (32×32 RGB natural images). We adopt a convolutional architecture in the style of Laborieux et al. [4], but with a lighter configuration, summarized in Table 5 (Appendix A). The reductions in depth (three convolutional blocks rather than four), feature-map width, and relaxation steps (T_1, T_2) are a direct consequence of our compute budget: each job was restricted to a single A100 MIG slice with a four-hour wall-clock limit, whereas the reference configuration was trained for roughly two days per run. We therefore do not target state-of-the-art accuracy; our objective is a *relative* comparison of strategies under identical, fixed conditions.

This distinction matters because equilibrium propagation does not behave uniformly across datasets. Comparative studies of energy-based learning algorithms [11] have shown that on simple tasks such as MNIST, different EP variants perform almost identically, whereas clear and sometimes large performance gaps emerge as task difficulty increases; in particular, one-sided (standard) EP can degrade severely on harder visual tasks, while centered EP remains comparatively robust. Consequently, the strategy selected on MNIST is not guaranteed to transfer to a convolutional model on CIFAR-10. Rather than assume it does, we treat CIFAR-10 as a setting in which to re-examine the full hybrid family.

Table 1: MNIST: final test accuracy and cumulative nudged-phase count. Accuracy is reported as mean $\pm$ standard deviation over 5 seeds. For the nudged-phase count no deviation is reported: it is fixed by the schedule and identical across seeds for std and cep, while the small seed-to-seed variation of the hybrids is omitted for clarity. The same convention applies to the other tables.

Strategy	Test acc. (%)	Nudged phases
Standard EP	94.73 ± 0.04	23,450
Centered EP	95.04 ± 0.32	46,900
Cosine hybrid	$\mathbf{95.34 \pm 0.17}$	34,962
Inverse cosine hybrid	94.94 ± 0.27	35,361
Stochastic hybrid	95.05 ± 0.16	35,164

Training uses standard CIFAR-10 augmentation (random crop with 4-pixel padding, random horizontal flip), per-layer learning rates, and a two-epoch warmup. On this dataset we evaluate all three hybrid schedules - cosine, inverse cosine, and stochastic - against the `std` and `cep` baselines. Owing to the higher per-run cost, CIFAR-10 experiments use three seeds. As on MNIST, each strategy is assessed by its final test accuracy and cumulative nudged-phase count.

3.5 Implementation details

All experiments build on the publicly available Equilibrium Propagation codebase of Laborieux et al. [4]. We reuse its energy-based models, relaxation dynamics, and gradient computation unchanged; our only addition is a per-minibatch dispatch that, with probability $p_{\mathrm{std}}(e)$, applies the standard update and otherwise the centered update (Eq. (11)).

All training experiments were conducted on the DelftBlue supercomputer at TU Delft [12]. MNIST hybrid training runs were executed using the `gpu-v100` partition (NVIDIA Tesla V100S, 32 GB), with 4 CPU cores and 16 GB RAM per experiment. Fashion-MNIST and CIFAR-10 experiments were run using the `gpu-a100-small` partition (NVIDIA A100, 10 GB MIG instance), with 2 CPU cores and 8 GB RAM per job.

4 Results

We evaluate the strategies of Section 3.1 on three datasets of increasing difficulty - MNIST (Section 4.1), Fashion-MNIST (Section 4.2), and CIFAR-10 (Section 4.3) - and draw the comparison together in Section 4.4. Throughout, std denotes one-sided standard EP, cep centered EP, and $p_{\mathrm{std}}(e)$ the per-batch probability of using a std update at epoch e.

4.1 Strategy Comparison on MNIST

We trained the cosine schedule of Eq. (14) on MNIST against the two baselines it interpolates between, std and cep, and against the other two hybrids - the inverse cosine and stochastic schedules - under the protocol of Section 3 (5 seeds; mean $\pm$ standard deviation). The inverse cosine schedule, which runs the cosine schedule backwards (cep early and std late), lets us read off whether the *direction* of the schedule matters rather than merely the fact of mixing two estimators. The final test accuracies are shown in Figure 1 and the full metrics in Table 1.

The cosine hybrid is the best of the five: it reaches $95.34\% \pm 0.17$, ahead of pure cep ($95.04\% \pm 0.32$) and pure std ($94.73\% \pm 0.04$). On this task it also exhibits lower variance than cep.

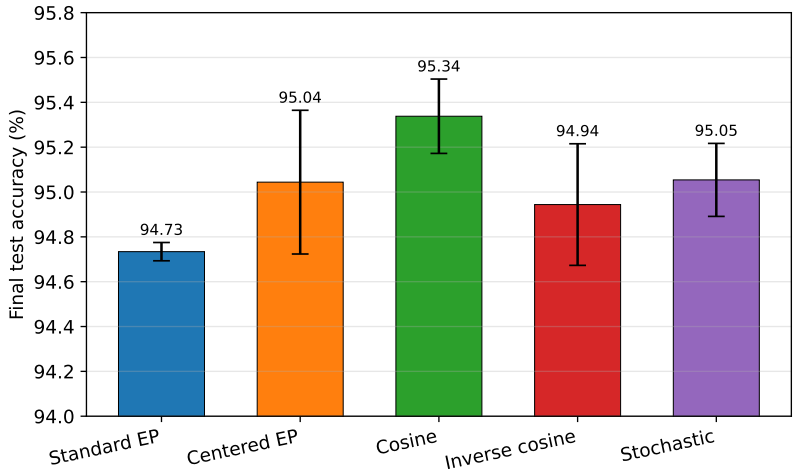


Figure 1: MNIST final test accuracy for standard EP, centered EP, and the three hybrids: the cosine (std $\rightarrow$ cep), inverse cosine (cep $\rightarrow$ std), and stochastic (fixed $50/50$) schedules. Bars are means and error bars are ± 1 standard deviation over 5 seeds. The cosine hybrid attains the highest final accuracy and exceeds the inverse cosine hybrid despite using the same two estimators in the same overall proportion.

The inverse cosine schedule (Eq. (15)) runs the same two estimators in *the same overall proportion* but in the opposite order - cep early and std late - and reaches only $94.94\% \pm 0.27$, below the cosine schedule, cep, and the stochastic schedule, while only marginally above std. This contrast, between the cosine schedule and its mirror image, is the most informative comparison here. Since the two schedules differ in nothing but which estimator is used early versus late, the cosine schedule's advantage cannot come from estimator mixing alone: it comes from the *ordering*. Spending the cheap estimator early, while it is still reliable, and the accurate one late is better than the reverse - consistent with the diagnostic of Appendix B, but established here directly from the two schedules' final accuracies. Finally, the stochastic schedule lands essentially on cep ($95.05\% \pm 0.16$).

On the cost axis, the two baselines bound the field. Standard EP runs exactly one nudged phase per update, marking the lower bound at $23{,}450$ over the 50-epoch run, while centered EP runs two and marks the upper bound at exactly double, $46{,}900$. As expected, the three hybrids land almost exactly halfway between the two baselines, at roughly $35{,}000$ nudged phases - about as far above the std floor as below the cep ceiling. This acts as a control: in nudged-phase terms the hybrids occupy the middle ground they were designed to.

4.2 Generalization to Fashion-MNIST

Before scaling up in complexity, we tested whether the MNIST conclusion survives a domain shift that keeps the input structure fixed (28×28 grayscale images). Using the two-hidden-layer architecture and EP dynamics of Section 3 (5 seeds), we obtain the accuracies shown in Figure 2 and the full metrics in Table 2.

The three strategies finish within 0.13 percentage points of one another (88.60%, 88.73%, and 88.62% for std, cep, and the cosine schedule), a gap well inside the seed-to-seed standard deviation ($\approx 0.16\%$). When the estimators agree, reducing the bias of the gradient estimate buys little final accuracy. Two points carry forward to the convolutional setting. First, the cosine schedule again matches the accuracy of both baselines while issuing about 25% fewer nudged phases than cep ($34{,}962$ versus $46{,}900$), so its compute advantage

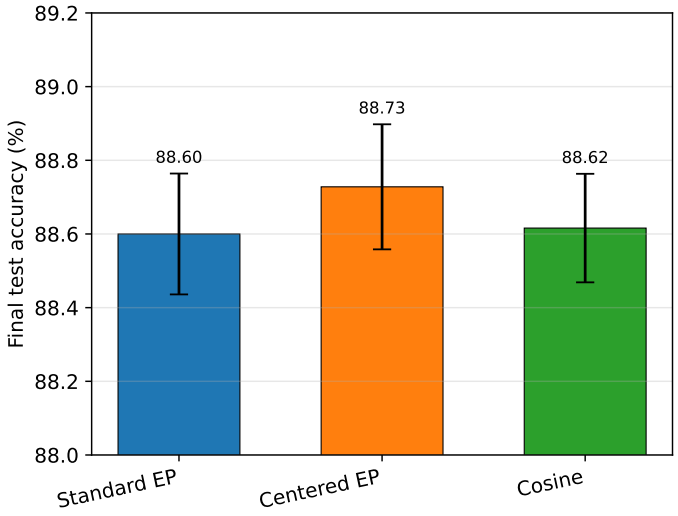


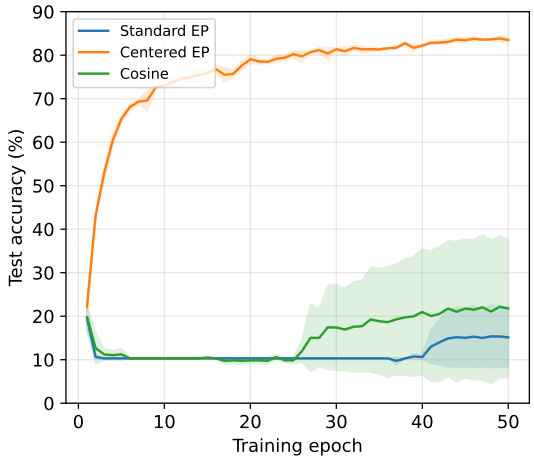
Figure 2: Fashion-MNIST final test accuracy for standard EP, centered EP, and the cosine hybrid (std $\rightarrow$ cep). Bars are means and error bars are ± 1 standard deviation over 5 seeds. The three strategies are comparable, with overlapping error bars.

Table 2: Fashion-MNIST: final test accuracy, and cumulative nudged-phase count. Accuracy is reported as mean $\pm$ standard deviation over 5 seeds; the deviation of the nudged-phase count is omitted for clarity.

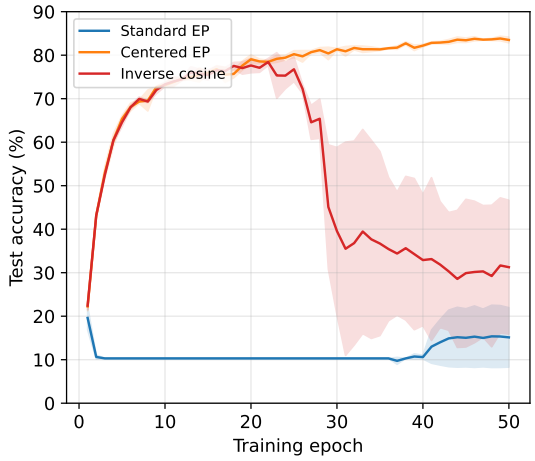
Strategy	Test acc. (%)	Nudged phases
Standard EP	88.60 ± 0.16	23,450
Centered EP	$\mathbf{88.73 \pm 0.17}$	46,900
Cosine hybrid	88.62 ± 0.15	34,962

transfers across the domain shift. Second, and more importantly, std here remains a perfectly viable estimator - it does *not* collapse - which is precisely the precondition that the harder CIFAR-10 model fails to meet (Section 4.3).

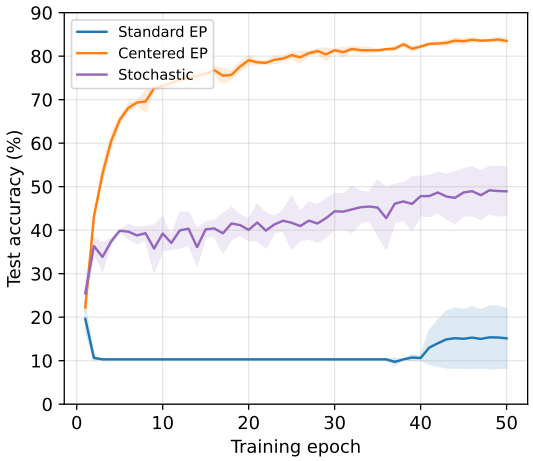
While benchmarking is not the primary aim of this work, these runs provide an additional comparison that appears to be missing from the existing Fashion-MNIST EP literature in the multilayer-perceptron setting. The non-dendritic baseline reported by Kubo [10] (85.27%) uses the symmetric (centered) estimator. Under the same architecture, our centered EP reaches 88.73%, which we attribute to differences in the optimization protocol (cosine-annealed learning rate and weight decay) rather than to the estimator itself. In addition, we report a genuine one-sided standard-EP baseline (88.60%), which to our knowledge is not documented either for this architecture or for the MLP setting, together with the cosine schedule, giving a more complete side-by-side view of the three estimators on Fashion-MNIST. We emphasize that these numbers are specific to shallow fully-connected models and do not approach the accuracy of convolutional energy-based networks on this task (e.g. the deep convolutional Hopfield networks of Scellier et al. [11], at $\approx 93.5\%$).



(a) Cosine hybrid (std $\rightarrow$ cep).



(b) Inverse cosine hybrid (cep $\rightarrow$ std).



(c) Stochastic hybrid (fixed $50/50$).

Figure 3: CIFAR-10 convolutional model: test accuracy versus epoch, with standard EP and centered EP shown in every panel for reference. **(a)** the cosine hybrid; **(b)** the inverse cosine hybrid; **(c)** the stochastic hybrid. Lines are means and bands are ± 1 standard deviation over 3 seeds. Standard EP collapses on this model, dragging down any schedule that relies on a prolonged std-dominated phase; only the stochastic hybrid trains stably.

4.3 Scaling to CIFAR-10

We then evaluated the same ideas on the convolutional model of Table 5 on CIFAR-10 (3 seeds). As discussed in Section 3, EP variants are known to behave very differently as task difficulty increases, so CIFAR-10 is treated as a setting in which to re-examine the schedule rather than one in which the MNIST choice is assumed to transfer. The per-strategy learning curves are shown in Figure 3 and the final metrics in Table 3.

Table 3: CIFAR-10 convolutional model: final test accuracy, and cumulative nudged-phase count. Mean $\pm$ standard deviation over 3 seeds; the deviation of the nudged-phase count is omitted for clarity.

Strategy	Test acc. (%)	Nudged phases
Standard EP	14.91 ± 6.94	19,550
Centered EP	$\mathbf{83.35 \pm 0.28}$	39,100
Cosine hybrid	21.86 ± 16.57	29,118
Inverse cosine hybrid	31.18 ± 15.47	29,503
Stochastic hybrid	49.32 ± 5.27	29,321

The picture changes sharply. On this harder task one-sided standard EP does not merely lose a little accuracy: it collapses, reaching only $14.91\% \pm 6.94$ (individual seeds at 10.0%, 10.0%, and 24.7%, i.e. at or near chance), while centered EP remains robust at $83.35\% \pm 0.28$. This matches prior reports that one-sided EP can degrade severely as visual complexity grows whereas centered EP stays stable [4, 11].

Crucially, this instability also breaks the cosine schedule. Because the schedule (Eq. (14)) is std-dominated throughout its early phase, the cosine schedule inherits the same poor start as pure std; and once p_{cep} grows, the network is no longer able to recover, improving only marginally for the remainder of training. It finishes at $21.86\% \pm 16.57$, the large variance reflecting that some seeds collapse outright (Figure 3a).

The shape of the centered-EP learning curve makes the inverse cosine schedule a natural candidate here. The cep curve improves steeply in the first few epochs and then makes only small gains as it approaches convergence (Figure 3). Because it is cep-dominated early and std-dominated late, the inverse cosine schedule front-loads the accurate updates, under the hypothesis that if the network learns most of what it needs during the accurate cep-driven early phase, then once p_{std} takes over the first-order bias should matter less, and the network should be able to continue drifting slowly toward convergence despite the noisier updates.

The results confirm the first half of this hypothesis but refute the second. Early training does proceed well under the cep-dominated schedule; however, once the std updates take over, the bias does not merely slow progress but actively degrades it, and the learning curve falls off with a steep negative slope driven by the run of consecutive standard-EP steps (Figure 3b). This is consistent with the diagnostic of Appendix B: there, the cosine similarity between the std estimate and the BPTT reference *decreased* relative to that of cep as training progressed, indicating that std updates become increasingly misaligned with the true gradient later in training. Concentrating those updates in the final epochs therefore incurs the largest possible penalty, which is precisely the steep decline we observe. The inverse cosine schedule finishes at $31.18\% \pm 15.47$. Nonetheless, it is worth noting that the inverse cosine schedule is already an improvement over the cosine schedule at the same compute cost.

The stochastic hybrid is designed precisely to avoid this late-stage collapse. The two annealed schedules both fail for the same underlying reason: each concentrates std updates into one contiguous stretch of training, and it is this distribution - not the number of biased updates - that destabilizes the network. By holding $p_{\mathrm{std}} = 0.5$ and selecting the estimator independently for each minibatch (Section 3.1), the stochastic hybrid balances the two estimators at the finest available granularity rather than across whole phases of training. As a result the network is never left in a prolonged one-sided regime: within any short window roughly half the updates are cep, repeatedly re-stabilizing the trajectory before the bias can accumulate.

Both the learning curve (Figure 3c) and the variance (Table 3) identify the stochastic schedule as the most stable and reliable of the three hybrids. It trains without collapse on any seed (every run above 42%), and its standard deviation (± 5.27) is roughly a third of that of the annealed schedules (± 16.57 and ± 15.47). It reaches $49.32\% \pm 5.27$ at 29,321 nudged phases.

In compute terms this is 9,779 fewer nudged phases than centered EP (39,100), a 25% reduction, and - as expected for an unbiased per-minibatch choice - it falls almost exactly at the midpoint between std (19,550) and cep (39,100). Its accuracy sits at a comparable midpoint, recovering 34 percentage points over pure std (14.91%) while remaining 34 points short of pure cep (83.35%). The stochastic hybrid therefore offers the best stability-compute trade-off among the schedules introduced in this paper, but it does not match centered EP: on a model where std is unstable, mixing in std updates to save computation comes at a substantial accuracy cost, and pure cep remains preferable when accuracy is the priority.

A limitation of these CIFAR-10 experiments is the deliberately lightweight convolutional architecture, adopted to fit the four-hour per-job budget on a single A100 MIG slice (Section 3). It is substantially smaller than the reference network of Laborieux et al. [4] in both depth (three convolutional blocks versus four) and width (32-64-128 versus 128-256-512-512), and uses fewer relaxation steps. Therefore, it is plausible that a heavier architecture and longer relaxation would raise the accuracy of the stochastic schedule, and could alter the size of the gap to centered EP.

4.4 Synthesis

Hybrid gains are real but task-dependent. Taken together, the three datasets delineate the conditions under which a hybrid schedule is useful. The benefit of scheduling is contingent on the estimators being individually viable. On MNIST, where std trains stably on its own, the cheap-early/accurate-late schedule was simultaneously the most accurate and the most economical strategy; on Fashion-MNIST, where the strategies reached comparable accuracy, the cosine schedule's contribution reduced to its compute saving; and on the CIFAR-10 model, where std collapses, every schedule containing a prolonged std-dominated phase collapsed with it. Even the stochastic hybrid, which trains reliably on this model, does not recover full centered-EP accuracy, so where one-sided updates are actively harmful the most dependable choice remains pure cep - the very regime whose cost the hybrid was designed to reduce. The compute savings of hybrid EP are therefore real but task-dependent, and are realized most cleanly on tasks where standard EP is itself a viable estimator.

Stability depends on how concentrated biased updates are in time, not on their number. The CIFAR-10 experiments, however, support a sharper statement about *why* the schedules differ. The three hybrid strategies issue nearly the same number of nudged phases (29,118-29,503; Table 3), meaning that each replaces roughly half of the centered updates with standard ones; they differ only in how those standard updates are distributed over training. Yet their final accuracies range from 21.86% to 49.32%. The damage caused by the biased estimator is therefore not proportional to the number of biased updates. Under the stochastic schedule roughly half of the updates within any window of training are one-sided, and the network nevertheless trains without collapse on every seed; the same overall fraction of one-sided updates, concentrated into a single contiguous stretch at the start (cosine schedule) or the end (inverse cosine schedule) of training, is destructive. What is harmful is thus not an individual biased update, or even a bounded burst of them, but their concentration into a long stretch: spread uniformly through training, the small error each biased update introduces is corrected by the centered updates around it; concentrated together, these errors compound until the network reaches a state from which it does not recover. A plausible mechanism, consistent with the diagnostic of Appendix B, is that the misalignment of the std estimate grows as the network advances along its training trajectory, so that consecutive biased updates reinforce rather than cancel one another, while interleaved cep updates repeatedly re-anchor the trajectory before the accumulated error can grow.

The same view explains the collapse of standard EP. This reading has implications beyond the hybrid estimators studied here. Most immediately, it explains the collapse of standard EP itself on CIFAR-10: pure std is the extreme in which one-sided updates fill the entire training, with no centered updates ever arriving to correct the accumulating error, while pure cep is the opposite extreme, with no biased updates at all. Seen this way, the five strategies of Table 3 order themselves by the longest stretch over which one-sided updates

dominate - all of training (std), an extended early (cosine) or late (inverse cosine) phase for the two annealed hybrids, where p_{std} is high and centered updates only rarely interrupt, no more than a few consecutive batches for the stochastic hybrid, and none (cep) - and their final accuracies follow exactly this order. The gap between the two annealed schedules indicates that the placement of this stretch matters as well: a one-sided phase at the start of training prevents the network from ever leaving its initial collapsed state, whereas the same phase placed at the end degrades an already partially trained network, which proves less damaging.

5 Conclusion

This thesis introduced hybrid Equilibrium Propagation, a family of estimators that chooses between the standard and centered EP estimators independently on each minibatch, applying the cheap one-sided update with probability p_{std} and the accurate centered update otherwise. A bias analysis showed that the hybrid estimator carries a bias of $\mathcal{O}(p\beta + \beta^2)$: a p-controlled first-order term sitting above the irreducible second-order floor of centered EP, so that the mixing probability acts as a continuous dial between the two regimes. Within this family we study three schedules that differ only in how the cheap one-sided updates are distributed over training: a cosine-annealed schedule that front-loads them, reserving the accurate centered updates for late training; its mirror image, the inverse cosine schedule; and a stochastic schedule that holds the mixing probability fixed at one half.

The empirical evaluation answered the central research question - under what conditions hybrid EP can reduce the cost of centered EP while preserving stability and accuracy - with a condition that is simple to state: the constituent estimators must be individually viable on the task. On MNIST, where standard EP is stable, the cosine schedule was simultaneously the most accurate strategy (95.34%) and cheaper than centered EP, issuing roughly 25% fewer nudged phases; an inverse-schedule control confirmed that this gain comes from the direction of the schedule rather than from mixing alone. On Fashion-MNIST the strategies reached comparable accuracy, so the cosine schedule's benefit reduced to the same compute saving. On a convolutional model on CIFAR-10, however, standard EP collapsed to near-chance accuracy and dragged down every schedule that relies on a prolonged one-sided phase; only the stochastic hybrid, a constant per-batch $50/50$ mix, trained stably - at the same 25% compute saving but well short of centered-EP accuracy.

Beyond this condition, the CIFAR-10 experiments yielded a more general observation about training with biased gradient estimators. The three hybrid schedules issued nearly identical numbers of nudged phases and differed only in how the one-sided updates were arranged in time, yet their final accuracies differed by almost a factor of two and a half. The damage caused by estimator bias is therefore governed less by the number of biased updates than by their distribution in time: when they are spread out, each is corrected by the centered updates that follow, whereas when they are concentrated into a long stretch the errors compound. This suggests that schedules for mixing estimators of unequal quality in EP should interleave at the finest granularity available rather than partition training into homogeneous phases.

These findings open several directions. The first is to broaden the family of hybrid schedules itself. We studied only three ways of setting the mixing probability p_{std} - a cosine anneal, its inverse, and a fixed stochastic mix - but many others are possible. All three sit near a balanced mix, spending on average roughly half their updates on each estimator, so that their total number of nudged phases falls about midway between standard and centered EP. Skewing this balance is an immediate next step: a schedule that uses, say, one-quarter standard and three-quarters centered updates would sit closer to centered EP in both cost and accuracy, while the opposite skew would trade accuracy for a larger compute saving, tracing out the full trade-off curve between the two estimators rather than only its midpoint. Beyond the overall balance, the schedule can also be made adaptive. Instead of being fixed in advance, the mixing probability could be adjusted during training in response to a measure of how well the cheap estimator currently tracks the true gradient, spending one-sided updates only while they remain reliable. Each of these defines a new member of the hybrid family, and our bias analysis and the run-length effect we observed give concrete criteria for designing them.

A second direction follows from viewing hybrid EP as a middle ground between two anchors. Here the cheap anchor was standard EP and the accurate anchor was centered EP, but centered EP is itself only the first step in a hierarchy of lower-bias estimators: in principle one can cancel higher-order terms in β by adding further nudged phases at additional nudging strengths. Such estimators have attracted little attention, because centered EP is already regarded as expensive - it doubles the number of nudged phases relative to standard EP - and pushing the bias lower still would only make each update more costly, with no obvious way to recover that cost. Hybrid EP changes this calculus: because the expensive estimator is invoked on only a fraction of updates, its per-update cost is amortized over training, so a more accurate but more expensive anchor need not raise the overall budget. This makes higher-order estimators worth revisiting as the accurate anchor of a hybrid schedule, and is a natural candidate for tasks like CIFAR-10 where centered EP alone sets the accuracy ceiling.

Finally, it would be interesting to study hybrid EP on other energy-based networks. We evaluated it on two specific architectures, a multilayer perceptron and a convolutional network, but the hybrid estimators are defined only through the free and nudged phases that EP already provides, so the same construction applies in principle to any network trainable by EP. This includes architectures with different connectivity, such as the continuous Hopfield models for which EP was originally formulated. Whether the compute savings and the run-length effect observed here carry over to such settings is an open question that we leave to future work.

6 Responsible Research

This project is a computational study that involved no human subjects and collected no personal or otherwise sensitive data, so it required no approval from a human research ethics committee. Responsible conduct of research nonetheless bears on how the work is carried out and reported, and we discuss the dimensions most pertinent here: the reproducibility of our results, the avoidance of methodological bias, integrity in crediting external resources and disclosing the use of generative AI, and the broader impact of reducing the cost of training.

6.1 Reproducibility

We build on the publicly available codebase of Laborieux et al. [4] (Section 3) rather than an ad-hoc reimplementation, and we report the complete architectural and optimization configuration of every experiment, including the exact random seeds, in Appendix A. All results are reported as the mean and standard deviation over every seed that was run; no runs were discarded. The hardware used for each experiment is documented in Section 3, and the source code of our extension is made available at `https://github.com/stan-group/stefan_stoian` so that all reported numbers can be regenerated from scratch.

6.2 Bias

The datasets used here contain images of handwritten digits, clothing items, and a set of animals and vehicles, and carry no demographic attributes, so the societal harms typically associated with biased training data do not arise. The more relevant risk in this project is methodological: a comparison between gradient-estimation strategies can be skewed by tuning the shared training setup in favor of one of them. We mitigated this by holding the architecture, optimizer, learning-rate schedule, and random seeds fixed across all strategies within each dataset (Appendix A), and by reporting negative results - most notably the collapse of the annealed hybrid schedules on CIFAR-10 (Section 4.3) - with the same prominence as positive ones. The conclusions are explicitly scoped accordingly: we show the compute savings of hybrid EP to be task-dependent and caution against extrapolating the MNIST-scale findings to deeper models.

6.3 Use of generative AI

In accordance with the existing guidelines on the use of generative AI in bachelor's end projects, a Large Language Model was used in two capacities. First, as a coding assistant for the experiment and plotting scripts: debugging code, solving code errors, generating boilerplate code, and formatting print statements so that the training process and the aggregation of results are presented clearly in the console. Second, as a writing aid for this report: checking grammar, improving the style, and assisting with LaTeX formatting, all of which were subsequently reviewed, edited, and approved by the author. Generative AI did not replace critical thinking and was not used to produce, for example, the research question, the theoretical bias analysis, the experimental design, or the interpretation of the results.

6.4 Beyond the project

One of the goals of this work is to reduce the computational cost of training energy-based models, contributing to the broader effort toward energy-efficient, hardware-friendly alternatives to backpropagation. Its most direct anticipated impact is therefore positive: lowering the energy footprint of learning algorithms intended for neuromorphic hardware. As with any work that improves training efficiency, the benefit is not tied to a particular application, and cheaper training generically lowers the barrier to deploying machine-learning systems, beneficial or otherwise. Given the small scale of the models and datasets studied here, we consider this risk remote; the main recommendation attached to our findings is methodological rather than societal - hybrid schedules should not be adopted without first verifying that the standard estimator is stable on the task at hand.

References

[1] Rongguang Ye, Chenhao Ye, Chao Huang, Ming Tang, and Yunhao Liu. Beyond-backpropagation training: Methods, applications, and perspectives. 2026.

[2] Yaqi Guo, Fabian Braun, Bastiaan Ketelaar, Stephanie Tan, Richard Norte, and Siddhant Kumar. Local learning for stable backpropagation-free neural network training towards physical learning. *arXiv preprint arXiv:2603.24790*, 2026.

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

[4] 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*, 15:633674, 2021.

[5] Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[6] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[8] 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.

[9] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[10] Yoshimasa Kubo. Dendritic neural networks with equilibrium propagation. *arXiv preprint arXiv:2605.08135*, 2026.

[11] 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: 52705–52731, 2023.

[12] Delft High Performance Computing Centre (DHPC). *DelftBlue Supercomputer (Phase 2)*, 2024. `https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2`.

A Network Architectures and Training Configurations

This appendix collects the full architectural and optimization details of the experiments. Table 4 reports the multilayer-perceptron configurations used for the MNIST and Fashion-MNIST experiments, and Table 5 reports the convolutional configuration used on CIFAR-10, compared against the reference of Laborieux et al. [4].

Table 4: MLP architectures and training configurations for the MNIST and Fashion-MNIST experiments.

	MNIST	**Fashion-MNIST**
Architecture	784-256-10	784-256-256-10
Hidden layers	1	2
Activation	$\tanh$	$\tanh$
Free-phase steps T_1	100	120
Nudged-phase steps T_2	20	12
Nudging strength β	0.25	0.1
Optimizer	SGD	SGD
Learning rate	0.05	0.05
Momentum	0.9	0.9
Weight decay	10^{-4}	10^{-4}
LR schedule	Cosine ($\eta_{\min} = 10^{-5}$)	Cosine ($\eta_{\min} = 10^{-5}$)
Batch size	128	128
Epochs	50	50
Input normalization	Standard	$\mu = 0.2860,\ \sigma = 0.3530$
Seeds	42, 123, 456, 789, 1337	42, 123, 456, 789, 1337

Table 5: CIFAR-10 convolutional architecture and training budget: our configuration versus the reference of Laborieux et al. [4].

	Ours	**Laborieux et al. [4]**
Conv. feature maps	32-64-128	128-256-512-512
Conv. layers	3	4
Kernel size	3×3	3×3
Pooling	2×2 max	2×2 max
Free-phase steps T_1	150	250
Nudged-phase steps T_2	20	30
Nudging strength β	0.35	0.5
Loss	MSE	MSE
Batch size	128	128
Epochs	50	120
Learning rates	0.10-0.07-0.05-0.03	0.25-0.15-0.1-0.08-0.05
Seeds	42, 123, 456	-
Weight decay	3×10^{-4}	3×10^{-4}
Momentum	0.9	0.9
LR schedule	Cosine annealing	Cosine annealing
GPU	A100 (MIG slice)	RTX 2080 Ti
Time budget	4 h / job	$\sim$2 days / run

B Diagnostic: gradient alignment during training

To understand how gradient-estimate quality evolves during training - and thereby to motivate the direction of the cosine schedule (Section 3.1) - we ran a diagnostic experiment measuring how well each estimator tracks the ground-truth gradient as training proceeds. An MLP is trained on MNIST [5] with standard EP for 50 epochs, under the same configuration as the main MNIST experiment (Table 4). At each of 12 checkpoints ($e \in \{0, 1, 2, 3, 5, 7, 10, 15, 20, 30, 40, 50\}$) we compute three gradients on a fixed diagnostic batch: BPTT (ground truth via backprop through the converged free phase), std, and cep, and report the cosine similarity of std and cep with BPTT, averaged over 5 seeds. This diagnostic was run on a local workstation - Windows 11, Intel Core i5 13500HX (14 cores, 2.5 GHz), 32 GB RAM, and an NVIDIA RTX 4060 (8 GB VRAM) - rather than on the DelftBlue cluster used for the main training experiments (Section 3). The resulting trajectories are shown in Figure 4.

Two observations follow. First, in the earliest epochs ($e \leq 3$) the two estimators are statistically nearly identical: both align with BPTT at a cosine similarity above 0.99, and the gap between them is within the seed-to-seed noise. The first-order bias of std is simply not large enough to matter while the weights are still close to their initialization. Second, as training proceeds the std estimate becomes both less aligned with BPTT and noticeably noisier across seeds, whereas cep remains the more reliable direction. In other words, the expensive symmetric estimator earns its extra nudged phase mostly in the later stages of training.

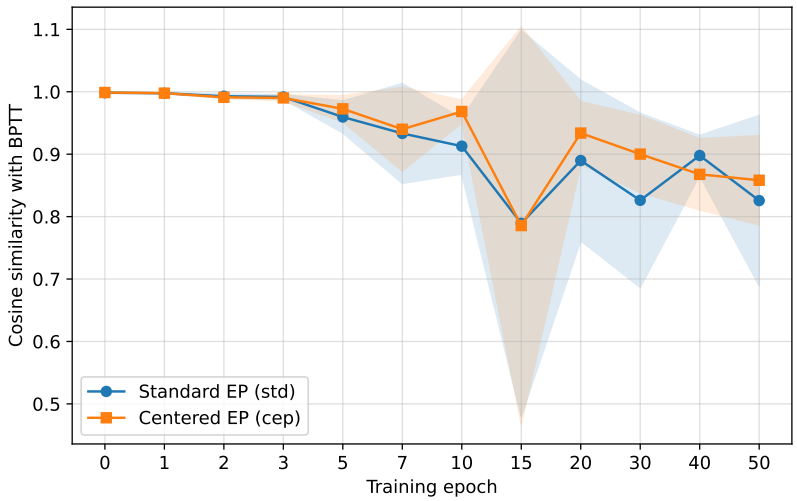


Figure 4: Cosine similarity of the standard-EP (std) and centered-EP (cep) gradient estimates with the BPTT gradient, as a function of training epoch. Lines are means and shaded bands are ± 1 standard deviation across 5 seeds. In the first few epochs the two estimators are nearly identical; as training progresses cep tracks BPTT at least as well as std and with smaller seed-to-seed variance.