analytics

MDPI

Review

Backpropagation-Free Continual Learning: A Unified Energy-Based Framework, Reproducible Benchmark, and Open Challenges

Cesar Hernando Valencia Niño

Grupo de Investigación UNITEL, Universidad Santo Tomás, Bucaramanga 680001, Colombia; cesar.valencia@ustabuca.edu.co

Abstract

Backpropagation has been the dominant credit-assignment mechanism behind the deep-learning revolution, yet it is global, memory-hungry, biologically implausible, and costly on hardware dominated by data movement. In parallel, predictive coding, equilibrium propagation, contrastive Hebbian learning and forward-only schemes have shown that useful credit assignment can also arise from local variables and the relaxation of an energy function. This manuscript is a review with a reproducible benchmark, not a proposal of a new learning rule. We organise existing backpropagation-free methods and continual-learning strategies under a common Local Equilibrium Learning (LEL) formulation, re-validate the core theoretical links numerically, and use a controlled continual-learning benchmark as an auditable case study within the review. The theoretical contribution is unification and verification, not priority over predictive coding, equilibrium propagation or replay. The empirical contribution is diagnostic rather than exhaustive: a benchmark of catastrophic forgetting with local networks that makes the review reproducible. We compare regularisation (EWC, SI), noise replay and generative "dreaming" across MNIST, Fashion-MNIST and CIFAR-10 in two independent implementations. Generative replay with local generators turns class-incremental Split-MNIST forgetting (chance level, the same as EWC and SI) into $86.8 \pm 0.2\%$, which is 96% of the joint-training ceiling and 97.4% in the task-incremental setting; the same mechanism reaches 82.0% on Fashion-MNIST and 37.0% on CIFAR-10. A matched control that holds the retention mechanism fixed and swaps only the learning rule shows that backpropagation with the same generative replay reaches a comparable or slightly higher accuracy than LEL (90.3% vs. 86.7% class-incremental); the benefit is attributable to generative replay as a retention mechanism, not to the local learning rule itself, which we state plainly rather than let the headline number imply otherwise. The CIFAR-10 number is intentionally reported as a limitation of a shallow raw-pixel local model, not as a claim of competitiveness with modern CNNs or transformers trained by backpropagation. We separate positive results; negative results and open limitations; and release scripts, seeds, dataset preparation and configuration files so that this paper can serve as an honest entry point for backpropagation-free continual learning.

Keywords: predictive coding; equilibrium propagation; local learning; continual learning; catastrophic forgetting; generative replay; energy-based models; backpropagation-free learning; neuromorphic computing

Check for updates

Academic Editor: Alessandro Provetti

Received: 4 July 2026
Revised: 16 October 2026
Accepted: 30 October 2026
Published: 13 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

How a neural network decides which of its many parameters to change in response to an error, the *credit-assignment* problem, is the engine of deep learning. For four decades, that engine has been backpropagation of error [1], and its dominance has become self-reinforcing: hardware and software have co-evolved around it, an instance of the *hardware lottery* in which an algorithm prevails in part because it fits the machines available to run it [2]. The brain, however, is the original learning system and the one these models were meant to emulate, and it almost certainly does not implement backpropagation in its literal form [3,4], which has long motivated a search for credit-assignment rules that are *local*, energy-efficient and biologically tenable. This survey reviews that search and studies, empirically, what it buys us in one concrete arena: learning continually without forgetting.

That dominance is earned: from convolutional networks to transformers [5], the modern era is built on it. But backpropagation carries three structural constraints that the community has come to treat as unavoidable. Backpropagation is *global*: the gradient of each weight requires a signal propagated from the output back through the entire network. It is *memory-intensive*: the activations of the forward pass must be stored until the backward pass consumes them. And it *separates training from inference*. These properties have no clear biological substrate and make data movement, not arithmetic, the dominant energy cost of training, the well-known bottleneck of von Neumann hardware [6].

A long-standing alternative replaces the global backward pass with *local* learning, driven by the relaxation of an energy function. Predictive coding [7,8], equilibrium propagation [9], contrastive Hebbian learning [10] and, more recently, forward-only schemes all assign credit using quantities available locally at each neuron or synapse. These methods are biologically motivated [10,11] and map naturally onto in-memory and neuromorphic substrates [12]. They are, however, scattered across formulations, notations, benchmarks and codebases, which obscures what is established, what is open, and how the pieces fit together, most of all for *continual learning*, where catastrophic forgetting remains a central obstacle to deployable AI [13].

Backpropagation-free learning is usually organised along other axes: biological plausibility [10,11], neuromorphic and in-memory hardware efficiency [6,12], or sample efficiency in supervised benchmarks. Each of these is a legitimate lens, but none of them turns locality into a claim that can be measured and falsified on its own terms: biological plausibility is judged against neuroscience data the present survey does not collect, and hardware efficiency is usually projected rather than benchmarked end to end (Section 7.1). We instead adopt continual learning as the organising lens of this survey, not as the primary historical motivation for backpropagation-free research, but as the setting in which its defining structural property, locality, has a direct and testable behavioural consequence. We give three reasons. First, continual learning stresses precisely the property backpropagation lacks: a continual learner must update online, without revisiting a stored computational graph of the past, which is the native regime of local, equilibrium-based methods, so the comparison is not incidental but structural. Second, energy-based local networks are *generative*, so they can rehearse the past by sampling their own model rather than storing data, a raw-data-free analogue of hippocampal replay during sleep [14]; this converts a qualitative biological analogy into a mechanism with a measurable computational role (Section 3.3). Third, the intersection of local learning and continual learning is one where the literature is already active but fragmented, spanning predictive coding with replay [15], equilibrium propagation, and forward-only schemes, each evaluated under different protocols, yet it lacks a unified, reproducible account; we document this inconsistency in evaluation protocols, benchmarks and baselines in detail in Section 3.3. Continual learning therefore lets us turn an abstract debate about credit assignment into one concrete, measurable question,

namely whether a backpropagation-free network can keep learning without forgetting, that organises the rest of the paper, without claiming that this is the only or even the historically primary lens through which backpropagation-free learning should be read.

Scope and contribution.

The manuscript is submitted as a *Review*. Its centre of gravity is therefore the synthesis of backpropagation-free credit assignment and continual learning, while the experiments serve as a reproducible benchmark embedded in that review. The theoretical material re-derives and numerically checks known energy-based correspondences; the benchmark tests one concrete continual-learning mechanism under controlled conditions; and the hardware and application sections explain why locality matters beyond this small case study. They should not be read as independent claims that the paper solves hardware deployment, modern vision or general artificial intelligence.

This paper is explicitly *not* a proposal of a new learning rule. Predictive coding, equilibrium propagation and generative replay all predate this work, including combinations of predictive coding with replay for continual and sequential learning [15]. The novelty claimed here is limited and concrete: *synthesis, reproducibility and diagnostic benchmarking*. We make four contributions, each with a different status:

(C1) A **synthesis contribution**: A taxonomy of backpropagation-free credit assignment (Section 2) and continual-learning strategies (Section 3) in a common language;
(C2) A **theoretical/reproducibility contribution**: A unified energy-based formulation (LEL) and a numerical re-validation of established results, including Lyapunov descent, the envelope-theorem local update and recovery of backpropagation in the small-nudge limit (Section 4);
(C3) A **reproducible benchmark contribution within the review**: A controlled case study of local continual learning, regularisation vs. replay vs. generative "dreaming", across three datasets and two implementations (Sections 5 and 6);
(C4) A **negative-results contribution**: An explicit account of where the approach fails or remains unproven, especially depth, CIFAR-10, compute cost and projected rather than measured hardware energy (Section 7).

All experiments are released as small, dependency-light scripts to serve as a reproducible entry point, mapped to their scripts and headline findings in Section 5. The paper should be read as a review-first contribution; the benchmark is used to make the synthesis executable and falsifiable.

Organisation.

Section 2 surveys backpropagation-free credit-assignment methods. Section 3 surveys continual-learning strategies and the conceptual boundaries needed to interpret the benchmark. Section 4 states the unified LEL formulation and validates the theoretical correspondences. Section 5 gives the reproducibility contract. Section 6 reports theoretical checks, main empirical results, negative results and diagnostic limitations. Section 7 covers hardware context, representative applications and open challenges. Section 8 concludes. Appendix A fixes the vocabulary used throughout, and Appendix B records implementation details for the released code.

Search strategy and study selection.

This manuscript is a narrative, taxonomy-building review with an embedded diagnostic benchmark, not a systematic review with meta-analysis, and we describe the search process at that level of rigour rather than overstate it. Coverage was built from Google Scholar, arXiv (cs.LG, cs.NE, q-bio.NC) and Semantic Scholar, using combinations of the terms *predictive coding*, *equilibrium propagation*, *contrastive Hebbian learning*, *forward-forward*

algorithm, *feedback alignment*, *target propagation*, *backpropagation-free*, *local learning rule*, *catastrophic forgetting*, *continual learning*, *elastic weight consolidation*, *synaptic intelligence*, *generative replay*, and *neuromorphic computing*, restricted to work published or posted between January 2016 and mid-2026, seeded by well-established papers in each subfield and extended by forward and backward citation chasing from them. Inclusion required that a work either (i) propose or analyse a credit-assignment rule that is local or explicitly backpropagation-free, (ii) propose or evaluate a continual-learning retention mechanism relevant to the taxonomy of Section 3, or (iii) provide a neuromorphic or in-memory hardware result germane to Section 7; preprints were included when they were the most direct or most recent source for a specific claim, and are labelled as not-yet-peer-reviewed where that is the case (e.g., [16]). We do not claim exhaustiveness or report a PRISMA-style flow of records screened and excluded, appropriate for a systematic review of primary outcomes but not attempted here; the taxonomy table (Table 1) is the explicit, checkable record of which families and mechanisms were judged in scope, and Section 3.4 states the boundary of that scope directly for the two cases a reviewer of this manuscript asked us to clarify.

2. Backpropagation-Free Credit Assignment

Credit assignment without a global backward pass takes several forms. The local alternatives can be organised by *how* they assign credit, and a key distinction, made precise by classical energy-based analyses [8,9], is that several of them coincide with backpropagation in a common limit (Section 2.5) while differing sharply in their dynamics and hardware fit. Figure 1 maps the three families; Figure 2 contrasts the two computational styles.

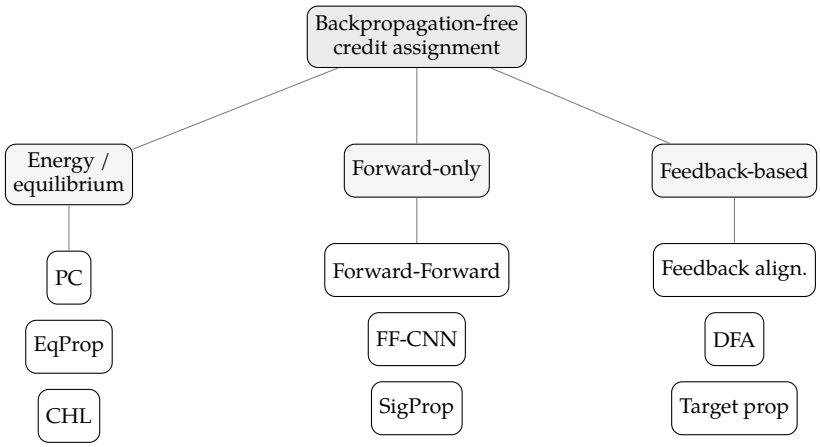


Figure 1. Taxonomy of backpropagation-free credit assignment. Energy/equilibrium and feedback-based methods retain an error signal (and recover backpropagation in a limit, Theorem 2); forward-only methods discard it entirely. LEL belongs to the energy/equilibrium family. Leaf references: PC [7,8]; EqProp and CHL [9]; Forward-Forward [17]; FF-CNN [18]; SigProp [19]; feedback alignment [4]; DFA [20]; target propagation [21].

The families surveyed below are not independent inventions but successive answers to one question: how to learn by settling an energy rather than propagating a gradient. Hopfield networks stored patterns as energy minima [22]; Boltzmann machines introduced the clamped-versus-free contrast [23]; predictive coding cast cortical perception as hierarchical error minimisation [7]; and equilibrium propagation made the energy-based gradient local and hardware-realisable [9]. In parallel, the complementary-learning-systems theory explained why fast and slow learning must coexist [14], and modern Hopfield networks showed that their single-step retrieval rule is transformer attention [22], connecting the energy-based programme to contemporary deep learning. The rest of this section organises the same body of work by *mechanism* rather than by date.

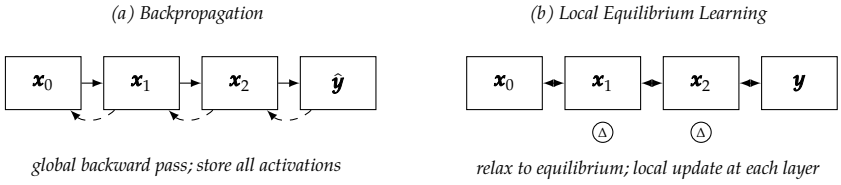


Figure 2. Block-level contrast. (**a**) Backpropagation runs a forward pass, stores every activation, and then propagates an exact error backwards through the whole network (dashed). (**b**) Local equilibrium learning relaxes all states to a fixed point (bidirectional arrows) and updates each weight from purely local quantities (Δ), with no backward pass and no stored graph. The two coincide in the small-nudge limit (Theorem 2).

2.1. Predictive Coding

Origins and biological motivation.

Predictive coding (PC) originates in computational neuroscience as a theory of cortical perception. Rao and Ballard proposed that the visual cortex is organised as a hierarchy in which each level maintains a generative model of the level below and continuously *predicts* its activity; only the residual *prediction error*, the part of the signal that the model failed to anticipate, is propagated upward [7]. This inverts the classical feedforward picture: feedback connections carry predictions, while feedforward connections carry errors. The framework explained a range of extra-classical receptive-field effects and provided a concrete, neurally grounded account of how cortical microcircuits could perform inference.

The free-energy formulation.

Friston generalised this idea into the *free-energy principle*, casting perception and learning as the minimisation of a variational free energy that upper-bounds Bayesian surprise [11]. Under Gaussian assumptions, the variational free energy reduces to a sum of squared, precision-weighted prediction errors, recovering the Rao–Ballard energy as a special case and embedding it in the language of approximate Bayesian inference. Bogacz provides a step-by-step derivation that makes the framework accessible to a machine-learning audience [11].

Energy, inference and the neural microcircuit.

In the discriminative machine-learning form used throughout this paper (Definition 1), a PC network defines the energy (2) as a sum of layerwise squared errors $\boldsymbol{e}_\ell = \boldsymbol{x}_\ell - f(\boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1} + \boldsymbol{b}_\ell)$. Inference is not a single forward pass but a *relaxation*: the hidden states descend the energy until the errors are mutually consistent. A key feature, central to the biological and hardware appeal of PC, is that this computation maps onto an explicit microcircuit with two populations, *value* (or representation) neurons that hold the states $\boldsymbol{x}_\ell$ and *error* neurons that hold the residuals $\boldsymbol{e}_\ell$, connected by synapses that perform the local update of Lemma 1. Both the error and the presynaptic activity needed to change a synapse are physically available at that synapse, so no separate backward pathway is required. Figure 3 shows one such junction.

Relation to backpropagation.

Whittington and Bogacz showed that a PC network with local Hebbian plasticity *approximates* backpropagation and equals it in a suitable limit [8]; we formalise and numerically validate this correspondence in Theorem 2. Song et al. identified a "prospective configuration" regime in which inference-then-learning can *outperform* gradient descent by settling neurons to their post-learning state before any weight update [10]. Incremental and parallel variants reduce the relaxation cost further [10,24], making the iterative inference increasingly a feature rather than a liability.

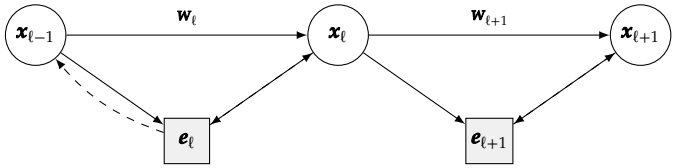


Figure 3. Predictive-coding microcircuit. *Value* neurons (circles) hold the states $\boldsymbol{x}_\ell$; *error* neurons (squares) hold the prediction residuals $\boldsymbol{e}_\ell = \boldsymbol{x}_\ell - f(\boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1})$. Solid arrows are the forward predictions that compute errors; dashed arrows are the local error signals that drive the relaxation and the Hebbian weight update (Lemma 1). Every quantity a synapse needs is available locally, unlike backpropagation's transposed backwards pathway.

Versatility.

Beyond classification, predictive-coding machinery supports generation, reconstruction, associative memory and learning on arbitrary graph topologies [25,26]. A notable connection: modern continuous-state Hopfield networks showed that their single-step retrieval rule *is* the attention operation of transformers [22], linking the energy-based programme directly to the dominant contemporary architecture.

Strengths and open limitations.

PC is local, biologically grounded, admits a clear probabilistic interpretation, and is naturally generative (the same energy that performs discrimination can be run top-down to synthesise inputs), a property we exploit for continual-learning "dreaming" in Section 6. Its principal limitations are practical: the inference relaxation is iterative and therefore slower than a single forward–backward pass on von Neumann hardware (Section 7), and training degrades in very deep networks because the layerwise errors become exponentially imbalanced, an open problem we reproduce in Section 6 and discuss in Section 7. Whether predictive coding can become a general-purpose substitute for backpropagation, rather than a theoretically illuminating but narrower alternative, remains an open and actively debated question in its own right [27].

2.2. Equilibrium Propagation and Energy-Based Learning

Energy-based models and the two-phase procedure.

Equilibrium propagation (EqProp) [9] learns over an energy-based model whose state settles to a fixed point of an energy $E(\boldsymbol{x}, \boldsymbol{\theta})$. Training proceeds in two phases. In the *free phase* the network relaxes to an equilibrium $\boldsymbol{x}^0$ with the output free. In the *nudged phase* a small force of strength β pulls the output toward the target, and the network relaxes to a slightly perturbed equilibrium $\boldsymbol{x}^\beta$ (Figure 4). The central result, the *gradient theorem*, is that the parameter update

$$\Delta\boldsymbol{\theta} \propto \frac{1}{\beta}\left(\frac{\partial E}{\partial \boldsymbol{\theta}}(\boldsymbol{x}^\beta, \boldsymbol{\theta}) - \frac{\partial E}{\partial \boldsymbol{\theta}}(\boldsymbol{x}^0, \boldsymbol{\theta})\right) \tag{1}$$

converges, as $\beta \to 0$, to the exact gradient of the loss, using only quantities local to each synapse and a *single* circuit for both phases. This is the same small-nudge limit we re-derive and validate for the predictive-coding form in Theorem 2.

Relation to Hopfield networks and contrastive Hebbian learning.

EqProp generalises the energy-descent dynamics of continuous Hopfield networks [22], whose equilibria are energy minima, and is closely related to contrastive Hebbian learning (CHL), in which weights are updated by the difference between a "clamped" and a "free" phase [9]. The lineage is older still: the two-phase structure descends directly from the positive (clamped) and negative (free) phases of the classical Boltzmann-machine learning rule [9], which already updated weights from the difference between two equilibrium

statistics. The energy-based methods PC, EqProp and CHL all compute the backpropagation gradient in a common infinitesimal-inference limit [8,9], so they are different dynamical realisations of the same underlying credit assignment.

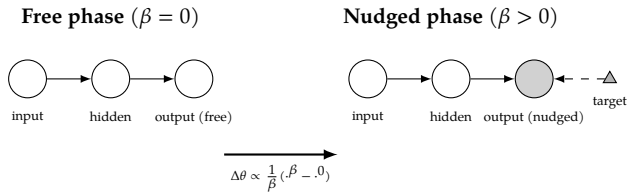


Figure 4. Equilibrium propagation. The network first relaxes with the output free (state $\boldsymbol{x}^0$), then with the output gently nudged toward the target by β (state $\boldsymbol{x}^\beta$). The local weight change is proportional to the difference between the two equilibria and, as $\beta \to 0$, equals the backpropagation gradient, using one circuit and only local quantities.

Extensions and hardware.

EqProp has been extended to convolutional and spiking architectures, and its single-circuit, local character makes it a leading candidate for analogue and neuromorphic implementation. Spike-driven EqProp (EqSpike) projects inference- and training-energy reductions of several orders of magnitude relative to GPUs and can learn online on neuromorphic substrates [12]. A practical drawback, the requirement of an infinitesimal nudge and, in the original formulation, symmetric weights, has motivated work that reduces the nudge-induced bias and broadens the comparison of these schemes with backpropagation [28,29].

Gradient-estimator bias and exactness.

A subtlety specific to EqProp is that a single finite nudge yields a *biased* estimate of the gradient, and this bias was identified as the main reason for poor scaling to harder visual tasks. Laborieux et al. cancel it with symmetric positive and negative nudges, enabling training of deep convolutional networks [28]; holomorphic equilibrium propagation computes *exact* gradients through finite-size oscillations rather than an infinitesimal nudge [30]; and comparative studies analyse the family of energy-based algorithms specifically for analogue computing [29].

Limitations.

Like PC, EqProp inherits the cost of an iterative relaxation and, historically, the difficulty of scaling to deep and large networks; both are active research fronts (Section 7).

2.3. Forward-Only Learning

A more radical departure discards the backward pass, and the propagated error itself, entirely. The Forward-Forward (FF) algorithm [17,18] replaces the forward–backward pair with *two forward passes*: one on "positive" (real) data and one on "negative" (contrastive) data. Each layer is trained by a purely local objective, to make a scalar "goodness" (e.g., the sum of squared activations) high for positive data and low for negative data, so credit assignment uses only quantities present at that layer, with no retrograde signal at all. The consequences are attractive for hardware: training can be pipelined and streamed, no computation graph is stored, and the weight-transport problem disappears. Signal-propagation and forward-projection schemes assign credit through a single forward sweep or closed-form projections [19,31]. Training with purely local error signals (auxiliary per-layer losses) has likewise reached respectable benchmarks [32]. The principal open challenges are designing effective negative data and closing the accuracy gap to backpropa-

gation on large models; this is the youngest and least saturated branch of the family, and the one we flag as the most promising for future work (Section 7).

Variants. A fast-growing literature refines FF along several axes. Convolutional extensions inject label information across all spatial positions through Fourier- or morphology-based spatial labels, enabling FF to train CNNs [18]; and channel-wise competitive learning partitions each layer's output channels by class and computes a group-wise goodness, sharpening class selectivity [17].

Open challenges. Generating effective negative data is ad hoc; the goodness objective relies on a discrete threshold; and inference can scale linearly with the number of classes. Closing the accuracy gap to backpropagation while resolving these issues is the defining open problem of this branch.

2.4. Feedback Alignment and Target Propagation

An intermediate class keeps a propagated error signal but avoids the exact transposed weights that backpropagation requires (the *weight-transport problem*). *Feedback alignment* replaces the transposed forward weights in the backward pass with fixed random matrices, yet the forward weights learn to partially *align* with these random feedback weights, so useful gradients still flow [4,33,34]. *Direct feedback alignment* routes the output error directly to every hidden layer through independent random projections, removing the sequential backward chain altogether [20]. *Target propagation* takes a different route: instead of propagating gradients it propagates *targets*, computed by learned approximate inverses, so that each layer-to-layer pair forms an autoencoder; difference target propagation corrects the inversion error [21]. These methods are more biologically plausible than backpropagation, especially in models that separate dendritic compartments or local error channels [35,36], but a recurring limitation is that approximation error accumulates across layers, which hampers scaling to very deep networks.

Scaling. Direct feedback alignment has been shown to scale to transformers and geometric learning; sign symmetry analyses show that only the *sign* of feedback weights needs to align with the forward ones [4]. The persistent limitation is accuracy at scale, where gradient approximation eventually costs performance.

Table 1. Qualitative comparison of credit-assignment families. "Local" means each update uses only quantities at the two neurons a synapse connects; "recovers BP" means the update equals the backpropagation gradient in a limit. "Activation memory" is the memory needed to *compute one layer's update* (Lemma 1), not the network's total simultaneous state, which for a full multilayer relaxation is $O(\text{depth} \times \text{width})$ for the energy-based rows as well, the same order as backpropagation; see the qualification in Section 2.5.

Family	Local Update	Recovers BP	Activation Memory	Neuromorphic Fit	Scales to Deep Nets
Backpropagation [1]	no	—	$O(\text{depth})$	poor	yes
Predictive coding [8]	yes	yes ($\beta \to 0$)	$O(\text{width})$	good	hard (active)
Equilibrium prop. [9]	yes	yes ($\beta \to 0$)	$O(\text{width})$	strong	hard (active)
Contrastive Hebbian [9]	yes	yes	$O(\text{width})$	good	limited
Forward-Forward [17]	yes	no	$O(\text{width})$	good	emerging
Feedback alignment [4]	partial	approx.	$O(\text{depth})$	medium	limited
Target propagation [21]	partial	approx.	$O(\text{depth})$	medium	limited

2.5. Theoretical Relationships and the Error-Locality Spectrum

A recurring confusion in this literature is whether these methods are rivals, special cases, or the same algorithm in disguise.

A common limit.

The deepest connection is that predictive coding, equilibrium propagation and contrastive Hebbian learning all compute the backpropagation gradient in a shared *infinitesimal inference limit*: when the output is nudged (rather than clamped) by an amount $\beta \to 0$, the local, equilibrium-based weight change aligns with the true loss gradient [8,9], which exhibits the three as different dynamical realisations of one underlying credit-assignment principle. Our Theorem 2 and its numerical validation (cosine $\to 1$ as $\beta \to 0$) instantiate this limit for the formulation used here. Away from the limit, e.g., with a fully clamped output, the local rule instead descends the model's own free energy (Theorem 1), which is *not* the task loss; this is why the measured alignment saturates below one under clamping.

A spectrum of error locality.

The families can be arranged on a spectrum according to how much of an error signal they propagate. At one end, backpropagation propagates an exact global error. Feedback alignment and target propagation keep a propagated error but approximate the transport [4,21]. Energy-based methods (PC, EqProp) replace propagation with relaxation: the error is computed locally and reconciled across layers by the dynamics rather than by an explicit backward sweep. At the far end, forward-only methods discard the propagated error entirely, learning each layer from a local objective on its own activations [32]. Moving along this spectrum trades fidelity to the exact gradient for locality, parallelism and hardware compatibility (Figure 5).

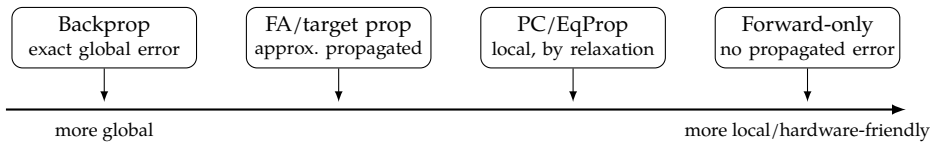


Figure 5. The spectrum of error locality. Methods are ordered by how much of an explicit error signal they propagate, from backpropagation's exact global error to forward-only learning's none. Moving rightward trades gradient fidelity for locality, parallelism and compatibility with in-memory/neuromorphic hardware; LEL sits in the energy-based middle.

Where LEL sits.

The Local Equilibrium Learning formulation used in this study is the discriminative, energy-based (predictive-coding-style) member of this family that we use as a single, transparent vehicle for the rest of the paper; it is defined formally in Section 4.

Which family for which constraint.

The relationships above go beyond taxonomy: they imply a practical decision rule, because the families trade the same quantities differently (Table 1). When the binding constraint is *activation memory*, training very wide or long-horizon models, or learning on-device, the energy-based methods hold a qualitative, not asymptotic, advantage: the per-layer weight update itself only needs $O(\text{width})$ quantities (Lemma 1) and there is no stored, replayable autodiff graph to keep alive from a forward pass to a later backward sweep, but a multilayer relaxation still holds all layers' states simultaneously, so total state memory remains $O(\text{depth} \times \text{width})$, the same order as backpropagation's stored activations (Section 4.4). When it is *hardware co-location*, analogue crossbars or neuromorphic chips, equilibrium propagation and predictive coding are the strongest fit, since the update reads only quantities present at the synapse and requires no *separate*, independently stored backward weight matrix, though it still needs symmetric read access to the transpose of the same forward weights, a requirement it shares with backpropagation and does not eliminate (Section 7.3). When it is *raw scalability to very deep networks today*, backpropagation and,

among the alternatives, feedback alignment remain ahead, because energy-based depth is still an open problem (Section 7). When it is *streaming or pipelined training with no stored graph*, forward-only methods are the most natural, at the current cost of an accuracy gap. And when it is *continual learning without stored data*, the focus of this survey, the energy-based members gain a real structural advantage, because the very same network can generate its own rehearsal data (Sections 4 and 6). No single family dominates; the right choice is dictated by which of these resources is scarce.

Depth of coverage and scope.

The treatment above is not evenly distributed: predictive coding and equilibrium propagation (Sections 2.1 and 2.2) receive more theoretical and empirical depth than forward-only learning (Section 2.3) or feedback alignment and target propagation (Section 2.4), and this asymmetry deserves an explicit justification rather than an implicit one. It follows directly from the continual-learning lens adopted in the Introduction, not from a judgement that energy-based methods are superior overall. First, only the energy-based families are natively *generative*: the same relaxation dynamics that perform inference can run in reverse to sample from the model, which is the structural property this survey's benchmark exploits (Section 3.3); forward-only and feedback-alignment methods have no analogous built-in generative mode, so they cannot currently be tested on the same diagnostic without adding a separate generator. Second, the unifying LEL formulation used as the theoretical and empirical vehicle throughout Sections 4–6 is itself an energy-based, predictive-coding-style model (Section 2.5), so the numerical re-validation of Lyapunov descent and the backpropagation limit is naturally deeper for that family. This choice of vehicle, not a claim of general superiority, is why the benchmark in Sections 5 and 6 tests only the energy-based family: forward-only learning and feedback alignment are surveyed at the taxonomy level (Sections 2.3 and 2.4) but are not benchmarked here, which we flag as an explicit scope limitation rather than a verdict on their merit. Indeed, on other axes the balance reverses: forward-only learning is explicitly flagged as "the youngest and least saturated branch of the family, and the one we flag as the most promising for future work" (Section 2.3), and direct feedback alignment already scales to transformer and geometric-learning architectures where energy-based methods do not yet (Section 2.4). Extending the present diagnostic benchmark to forward-only and feedback-alignment learners, using an external generator or a stored-exemplar baseline to supply rehearsal data, is listed as future work in Section 7.3.

3. Continual Learning: Problem and Methods

Continual learning asks a network to acquire a sequence of tasks over time while retaining what it learned before. That setting aligns with the structural traits of local, energy-based methods (online updates, a native generative direction) and is where credit-assignment alternatives to backpropagation are most likely to matter. Section 6 then studies how local networks fare on this problem.

3.1. Catastrophic Forgetting, Scenarios and Metrics

The phenomenon.

When a neural network is trained on a sequence of tasks, learning a new task tends to overwrite the weights that encoded earlier ones, so performance on the old tasks collapses, *catastrophic forgetting*, identified already in early connectionist models [13,37]. It is the single largest obstacle to lifelong, on-device and streaming learning, and it is not specific to backpropagation: any method that descends a task loss by adjusting shared weights is exposed to it, as our own negative result for local learning confirms (Section 6).

Three scenarios.

The difficulty depends sharply on what the model is asked to do at test time. Van de Ven and Tolias distinguish three canonical scenarios [13]: (i) *task-incremental* (Task-IL), where the task identity is given at test time and the model needs only to discriminate within a task (e.g., a multi-head output); (ii) *domain-incremental* (Domain-IL), where the structure is fixed but the input distribution shifts and the task identity is unknown; and (iii) *class-incremental* (Class-IL), where the model must choose among all classes seen so far without any task label. Class-IL is by far the hardest because it additionally requires calibrating outputs *across* tasks, and it is precisely the regime in which weight-regularisation methods collapse. We report Class-IL as our primary metric and Task-IL as a secondary one (Section 6); the three are contrasted in Figure 6.

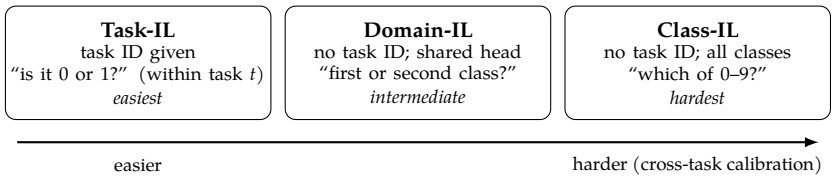


Figure 6. The three continual-learning scenarios [13], illustrated on Split-MNIST. Difficulty grows left to right with how much the model must resolve at test time; class-incremental learning, our primary metric, additionally requires calibrating outputs across tasks, the regime where regularisation methods fail.

Evaluation.

Standard metrics include the average accuracy over all tasks after the full sequence, the *forgetting* (drop from the peak accuracy of each task to its final accuracy), and *backward/forward transfer* [13]. Common benchmarks are Split-MNIST and Permuted-MNIST, Split-Fashion-MNIST and Split-CIFAR; we use Split versions of all three image datasets, and additionally report two informative bounds, the *joint-training ceiling* (all classes at once) and the *oracle* (replay of stored real data), which isolate the contribution of the continual mechanism from that of the base classifier.

3.2. Methods: Regularisation, Replay and Architecture

Three families dominate the continual-learning literature [13,37–39].

Regularisation.

These methods add a penalty that discourages changes to weights deemed important for past tasks. Elastic Weight Consolidation (EWC) weights a quadratic anchor by the diagonal of the Fisher information, an estimate of each parameter's importance [40]; Synaptic Intelligence (SI) accumulates an analogous importance online along the optimisation path [41]; and Memory Aware Synapses (MAS) measures the sensitivity of the network's outputs to each parameter, allowing unsupervised importance estimates [42]. A related, distillation-based variant, Learning without Forgetting (LwF), instead anchors *behaviour* rather than weights: it matches the previous model's responses on the new task's data through a knowledge-distillation loss, requiring no stored old data and no per-parameter importance estimate [43]. A central, well-replicated empirical fact, which our experiments reproduce, is that these methods work in task-incremental settings but *fail* in class-incremental ones, because a quadratic (or distillation) penalty cannot teach the network to separate classes that were never seen together [13].

Replay.

Replay-based methods revisit past experience while learning new tasks. The simplest, *pseudo-rehearsal*, labels random inputs with the previous model and rehearses them [44]; it is cheap but, as we show, fails in high dimension because random inputs lie off the data manifold. *Deep generative replay* instead trains a generative model (e.g., a VAE or GAN) to synthesise past data, which the solver then rehearses [44]; *brain-inspired replay* generates internal representations from the network's own context-modulated feedback, achieving strong class-incremental performance without storing raw data [45]. Several refinements have followed: *latent/feature replay* synthesises in the space of deep features rather than raw pixels, which is cheaper and more stable; *diffusion-based* generative replay improves sample fidelity; and *generative negative replay* rehearses contrastive rather than positive samples [46]. *Exemplar* methods instead store a small memory of real examples: iCaRL combines such a memory with nearest-mean classification and distillation [47], while Gradient Episodic Memory (GEM) uses the stored examples as constraints, projecting each update so that it does not increase the loss on any past task [48]; its averaged variant, A-GEM, relaxes this to a single aggregate constraint, trading a little retention for a large gain in speed and memory [49]. Energy-based networks are themselves generative, which makes "dreaming" (replaying self-generated samples) a natural, raw-data-free form of replay and the strategy studied in Section 6.

Architecture.

A third family allocates dedicated capacity per task or gates a shared network by context, isolating tasks at the cost of growing parameters or requiring task identity. Context-dependent gating with synaptic stabilisation is one representative example [50]; PackNet is another, packing several tasks into a single fixed-size network by iteratively pruning and freezing a task-specific subset of weights, which eliminates forgetting outright at the price of a per-task binary mask [51].

Table 2 compares the principal methods along the axes that matter for deployable, biologically grounded continual learning: whether they store raw data, whether they are generative, whether they avoid backpropagation, and whether they succeed in the demanding class-incremental scenario.

Table 2. Comparison of continual-learning methods. "Class-IL" summarises the reported behaviour in the class-incremental scenario under the assumptions listed in the table; "BP-free" marks methods whose main credit-assignment step does not require backpropagation. The last row is the strategy benchmarked in this review.

Method	Family	Stores Data	Generative	BP-Free	Class-IL
EWC [40]	regularisation	no	no	no	poor
SI [41]	regularisation	no	no	no	poor
MAS [42]	regularisation	no	no	no	poor
LwF [43]	distillation	no	no	no	poor
iCaRL [47]	exemplar	yes	no	no	moderate
GEM [48]	exemplar	yes	no	no	good
Experience replay	exemplar	yes	no	no	good
DGR [44]	gen. replay	no	yes (BP)	no	good
Brain-inspired replay [45]	gen. replay	no	yes (BP)	no	strong
Pseudo-rehearsal [44]	gen. replay	no	noise	no	poor (high-dim)
PackNet [51]	architecture	no	no	no	task-IL
Context gating [50]	architecture	no	no	no	task-IL
LEL + dreaming (this study)	gen. replay	**no**	**yes (local)**	**yes**	**effective (this study's protocol)**

The table makes the intended niche of this review explicit. To the best of our knowledge, the combination studied here, raw-data-free generative replay with local credit assignment in an energy-based classifier/generator, occupies a comparatively underexplored part of the class-incremental literature. The point is not to claim that no related method exists, but to provide a transparent benchmark for this particular combination of locality, replay and memory accounting.

3.3. Local Learning Meets Continual Learning

The intersection of local, energy-based learning and continual learning is recent and unusually active. Because energy-based networks are generative, "dreaming", replaying self-generated samples, is a natural strategy for them. Predictive-coding networks have been used for lifelong and sequential learning [15], and the underlying principle of *prediction-error-driven consolidation*, updating memory in proportion to surprise, fits naturally with continual adaptation. The *complementary learning systems* theory argues that the brain avoids catastrophic interference by pairing a fast, sparse hippocampal learner with a slow, distributed neocortical one, and interleaves replay of the former into the latter during consolidation [14], a picture reinforced by synaptic-level models of how a single synapse can retain traces at multiple timescales to resist overwriting [52], precisely the fast-learn-then-rehearse structure that generative replay imitates. More broadly, several brain mechanisms, hippocampal replay during sleep, spike-timing-dependent plasticity and sparse distributed memory, are now mapped onto machine-learning methods through associative-memory consolidation models [16,53] and surveys of brain-inspired continual learning [54]. Our empirical case study (Section 6) sits squarely in this line. Closely related work already exists in predictive coding, equilibrium propagation and generative replay; the claim here is not that any one component is new. Rather, to the best of our knowledge, prior work has not combined the same review taxonomy, reproducibility protocol and diagnostic continual-learning benchmark under the LEL formulation. The case study is included to make that synthesis executable and testable: the same energy-based framework classifies and dreams under a fairness-controlled protocol, leading to a conclusion that the continual-learning literature increasingly supports, parity in raw accuracy and a clear advantage of replay over unprotected learning.

Benchmarks, baselines and reproducibility.

Cumulative progress in this area has been hampered by inconsistent evaluation, which makes published numbers hard to compare. Three choices matter. The *benchmark*: The community has converged on Split- and Permuted-MNIST, Split-Fashion-MNIST and Split-CIFAR, but the difficulty depends entirely on which of the three scenarios (task-, domain- or class-incremental) is reported, and papers do not always state it [13]. The *metric*: Average accuracy over seen tasks, the forgetting (peak-minus-final accuracy), and backward/forward transfer capture different facets and should be reported together [13]. The *bounds*: Comparisons are only interpretable relative to a joint-training ceiling and, ideally, an oracle that replays stored data; without them, a high number may reflect a strong base classifier rather than an effective continual mechanism. For backpropagation-free methods specifically, early studies used incompatible protocols and ad hoc baselines, which has motivated unified, simplified benchmarking suites and frameworks [24]. We adopt these practices throughout Section 6, fixed scenarios, both bounds, multiple seeds, two independent implementations, and released code, precisely so that our results are reproducible and fairly comparable rather than another incommensurable data point.

3.4. Conceptual Boundaries and Design Principles

Interpreting experiments in this area requires keeping three levels of analysis separate. Table 3 fixes the vocabulary: credit-assignment mechanisms (backpropagation, predictive coding, LEL), continual-learning mechanisms (EWC, replay, architectural gating) and evaluation protocols (Split-MNIST, Class-IL, Domain-IL) are independent choices, and many apparent disagreements in the literature arise from conflating them. With that vocabulary in place, Tables 4 and 5 identify the seven replaceable components of a local continual learner and the main options for each; Tables 6 and 7 provide a structured benchmark progression where each rung exposes one additional difficulty. A method should normally demonstrate competence at a given rung before claiming the next.

Table 3. Three levels that must be kept separate when studying backpropagation-free continual learning. Many apparent disagreements in the literature come from mixing these levels rather than from genuine algorithmic disagreement.

Level	Examples	Question It Answers
Credit assignment	Backpropagation, predictive coding, EqProp, LEL, feedback alignment, target propagation, forward-only learning	How does a parameter know whether and how to change?
Continual mechanism	EWC, SI, MAS, LwF, replay, generative replay, architectural isolation, gating	How is old knowledge protected or rehearsed while learning new tasks?
Evaluation protocol	Task-IL, Domain-IL, Class-IL; Split-MNIST, Split-Fashion-MNIST, Split-CIFAR; joint and oracle bounds	What must the model decide at test time, and what resources were allowed during training?

Design patterns for local continual systems.

The design patterns in Tables 4 and 5 make the design space concrete by viewing a local continual learner as a pipeline with seven replaceable parts: an inference dynamics, a local synaptic update, a retention mechanism, a memory policy, a generator, an output-calibration scheme, and a hardware substrate. This modular view is useful because most improvements change only one part. A new relaxation schedule should be tested while keeping replay fixed; a new generator should be tested while keeping the classifier and task order fixed; a new memory policy should be compared under the same Class-IL evaluation. Otherwise, a positive result may be impossible to attribute.

This table also explains why the present paper is conservative. We fix the architecture and use simple local generators so that the role of replay can be seen clearly. A more competitive system would almost certainly change several rows at once, for example a deeper residual local architecture, a feature-space generator and a neuromorphic implementation. Such a system may be the right engineering target, but it is a poor first scientific test because success would not reveal which component mattered. The baseline here is deliberately smaller so that future work can make controlled substitutions.

A benchmark ladder.

A useful research programme should not jump directly from Split-MNIST to large-scale vision or language. Local methods have additional moving parts, and each benchmark should expose one new difficulty. We propose the ladder in Tables 6 and 7 as a practical progression. The lower rungs test whether the local rule works and whether forgetting is

real; the middle rungs test raw-data-free replay and generator fidelity; the upper rungs test scaling, representation learning and hardware.

Table 4. Design patterns for constructing a backpropagation-free continual learner, part I: the algorithmic core. Each row identifies a component that can be changed independently, together with the main diagnostic that should accompany the change.

Component	Typical Choices	Diagnostic to Report
Inference dynamics	clamped predictive-coding relaxation, nudged EqProp-style relaxation, incremental/parallel activity updates	Energy decrease, equilibrium residuals, number of steps to stability and sensitivity to the state step size
Synaptic update	Hebbian local error rule, contrastive difference, forward-only local objective, feedback-aligned local target	Whether the update is truly local and whether it aligns with backpropagation in the relevant limit
Retention mechanism	no defence, quadratic anchoring, replay, generative dreaming, task gating	Final accuracy and forgetting under Task-IL and Class-IL, not only one scenario
Memory policy	no data, compact statistics, generator parameters, feature buffers, raw exemplars	Exact number and type of stored quantities per class or per task

Table 5. Design patterns for constructing a backpropagation-free continual learner, part II: generation, calibration and substrate.

Component	Typical Choices	Diagnostic to Report
Generator	Oja/PCA subspace, local autoencoder, energy sampler with prior, associative memory	Sample fidelity, oracle gap, and whether generated samples remain on the data manifold
Calibration	single head, multi-head, task-conditioned output, nearest-mean classifier	Whether cross-task scores are comparable without task identity
Hardware substrate	CPU/GPU simulation, analogue crossbar, spiking neuromorphic processor, mixed-signal prototype	Measured latency and joules, separated from projected operation counts

Table 6. A benchmark ladder for backpropagation-free continual learning, rungs 1–4 (the diagnostic core used in this benchmark). Moving upward adds one major difficulty at a time, making failures easier to diagnose.

Rung	Benchmark	Purpose	Failure It Exposes
1	Two-moons or low-dimensional toy tasks	Verify local learning and forgetting under controlled geometry	Under-training mistaken for retention
2	Single-task MNIST/Fashion	Check stationary accuracy against a matched backpropagation model	Base local learner too weak
3	Split-MNIST Class-IL	Test cross-task calibration and replay under a standard protocol	Regularisation collapse and last-task bias
4	Generator-capacity ablation	Separate replay from generator fidelity	Poor samples blamed on continual learning

A method should normally pass a rung before claiming the next. For example, a high Class-IL number without a joint-training ceiling and an oracle replay bound does not show that the continual mechanism is strong; it may only show that the base task is easy. Conversely, a low CIFAR-10 number does not by itself refute the mechanism if the joint

ceiling is also low. The ladder is meant to keep the interpretation attached to the bottleneck actually being tested.

Table 7. Benchmark ladder, rungs 5–8 (scaling beyond this benchmark's scope).

Rung	Benchmark	Purpose	Failure It Exposes
5	Split-Fashion-MNIST and raw-pixel CIFAR-10	Probe dataset transfer and base-classifier ceiling	Low absolute accuracy caused by weak representation
6	Feature-space Class-IL	Test whether local replay works when representation quality improves	Hidden dependence on raw-pixel generators
7	Deep local networks	Test whether depth remedies balance equilibrium errors	Vanishing local error in early layers
8	Hardware-in-the-loop replay	Measure energy and latency in the intended substrate	Energy advantage that exists only on paper

Taxonomy scope: what is included and what is not.

The taxonomies of Sections 2 and 3 are broad on the credit-assignment axis but narrow on two others, and we state the boundary explicitly rather than leave it implicit. First, *non-energy backpropagation-free architectural methods for continual learning*, dynamically expanding networks, progressive columns, and hard architectural isolation or gating combined with a non-energy-based local rule such as feedback alignment or forward-only learning, are named in the continual-mechanism row of Table 3 ("architectural isolation, gating") and in the taxonomy of Section 2, but they are *not* exercised by the benchmark of Sections 5 and 6, which tests only regularisation and replay paired with the energy-based LEL formulation, for the reasons given in the depth-of-coverage discussion of Section 2.5. They are inside the scope of the taxonomy and outside the scope of the empirical case study. Second, *continual learning on graph-structured data*, incremental training of graph neural networks under an evolving node, edge or graph distribution, is outside the scope of both the taxonomy and the benchmark: the credit-assignment and continual-learning mechanisms surveyed here are developed for vector- and image-structured inputs (MLPs and CNNs), and message-passing architectures raise locality and credit-assignment questions of their own, which this survey does not address, e.g., whether a local update can respect the graph's connectivity rather than a fixed layer topology. We flag continual graph learning as a natural, currently unexplored extension rather than silently omit it.

4. A Unified Energy-Based Formulation (LEL)

To study these ideas concretely and reproducibly, we fix a single, minimal member of the energy-based family, which we call Local Equilibrium Learning (LEL). LEL is used here only as a notational umbrella, not as a new algorithmic family: it is deliberately standard, essentially discriminative predictive coding, so that the theory below restates established results in one notation and the experiments of Section 6 can be read without translating between formalisms.

4.1. Model

Definition 1 (Network as an energy system)**.** *A network of layers $\ell = 0, \dots, L$ has neuron states $\boldsymbol{x}_\ell \in \mathbb{R}^{n_\ell}$, input clamped to $\boldsymbol{x}_0 = \boldsymbol{d}$, and parameters $\boldsymbol{\theta} = \{\boldsymbol{W}_\ell, \boldsymbol{b}_\ell\}$. With $\boldsymbol{a}_\ell = \boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1} + \boldsymbol{b}_\ell$, prediction $\boldsymbol{\mu}_\ell = f(\boldsymbol{a}_\ell)$ and error $\boldsymbol{e}_\ell = \boldsymbol{x}_\ell - \boldsymbol{\mu}_\ell$ ($f \in C^1$), the free energy is*

$$F(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{2} \sum_{\ell=1}^{L} \| \boldsymbol{x}_\ell - f(\boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1} + \boldsymbol{b}_\ell) \|^2. \tag{2}$$

This is the standard predictive-coding energy [7,8]; inference minimises F over hidden states and learning minimises it over $\boldsymbol{\theta}$.

4.2. Inference as Relaxation

Hidden states follow $\dot{\boldsymbol{x}}_\ell = -\partial F/\partial \boldsymbol{x}_\ell$, with $\partial F/\partial \boldsymbol{x}_\ell = \boldsymbol{e}_\ell - \boldsymbol{W}_{\ell+1}^\top (f'(\boldsymbol{a}_{\ell+1}) \odot \boldsymbol{e}_{\ell+1})$. Figure 7 shows the resulting train-time loop.

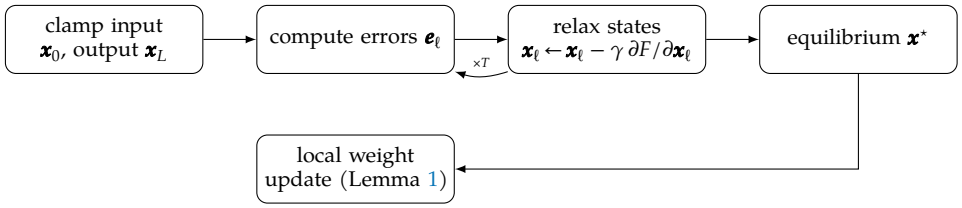


Figure 7. The LEL train-time loop. With input and output clamped, the network iterates error computation and state relaxation T times to an equilibrium $\boldsymbol{x}^\star$ (the inner loop), after which a single *local* Hebbian update is applied. There is no backward pass; the equilibrium itself performs the credit assignment (Theorem 1).

Assumption 1 (Coercivity and smoothness)**.** *The activation $f \in C^1$ is Lipschitz with constant L_f, and the free energy $F(\cdot, \boldsymbol{\theta})$ of (2) is coercive in the hidden states $\boldsymbol{h}$ for fixed $\boldsymbol{\theta}$, i.e., $F(\boldsymbol{h}, \boldsymbol{\theta}) \to \infty$ as $\|\boldsymbol{h}\| \to \infty$; this holds in particular whenever f is bounded (e.g.,* tanh *or logistic units), since then $\|\boldsymbol{h}_\ell\| \to \infty$ forces the error term $\|\boldsymbol{e}_\ell\| = \|\boldsymbol{h}_\ell - f(\boldsymbol{a}_\ell)\| \to \infty$. For the discrete-time result below we additionally take $\nabla_{\boldsymbol{h}} F(\cdot, \boldsymbol{\theta})$ to be Lipschitz with constant L_g on the sublevel set $\{F \le F(\boldsymbol{h}(0), \boldsymbol{\theta})\}$.*

Proposition 1 (Energy is a Lyapunov function)**.** *Under Assumption 1, along the continuous-time inference dynamics $\dot{\boldsymbol{h}}_\ell = -\partial F/\partial \boldsymbol{h}_\ell$,*

$$\frac{dF}{dt} = -\sum_\ell \left\| \frac{\partial F}{\partial \boldsymbol{h}_\ell} \right\|^2 \le 0,$$

with equality only at equilibrium; trajectories remain in the compact sublevel set $\{F \le F(\boldsymbol{h}(0), \boldsymbol{\theta})\}$ and converge to the equilibrium set. This is the classical energy-descent property of Hopfield and PC networks [22]; numerically, F decreases monotonically ($5.30 \to 2.68$) with $\|\partial F/\partial \boldsymbol{h}\| \to 0$.

Proof. By the chain rule and (2), $\frac{dF}{dt} = \sum_\ell \langle \partial F/\partial \boldsymbol{h}_\ell, \dot{\boldsymbol{h}}_\ell \rangle = -\sum_\ell \|\partial F/\partial \boldsymbol{h}_\ell\|^2 \le 0$, with equality iff every partial derivative vanishes, i.e., at an equilibrium. Boundedness of the trajectory is not automatic from $F \ge 0$ being non-increasing alone, since that only bounds $F(\boldsymbol{h}(t), \boldsymbol{\theta})$; it follows because the sublevel set $\{F \le F(\boldsymbol{h}(0), \boldsymbol{\theta})\}$ is compact under the coercivity of Assumption 1, and the trajectory never leaves it since F is non-increasing. Since $F \ge 0$ on this compact set, it converges; by LaSalle's invariance principle, the trajectory approaches the largest invariant set contained in $\{\dot{F} = 0\}$, namely the equilibrium set. Uniqueness of the equilibrium follows when the Hessian $\partial^2 F/\partial \boldsymbol{h}^2$ is positive definite (moderate spectral radii of the $\boldsymbol{W}_\ell$), a contractive regime. □

Corollary 1 (Discrete-time descent)**.** *Under Assumption 1, the discrete update actually used throughout this paper (Figure 7), $\boldsymbol{h}_\ell \leftarrow \boldsymbol{h}_\ell - \gamma\, \partial F/\partial \boldsymbol{h}_\ell$, is monotonically non-increasing in F whenever the step size satisfies $0 < \gamma \le 1/L_g$, with*

$$F(\boldsymbol{h}^{(k+1)}) \le F(\boldsymbol{h}^{(k)}) - \left(\gamma - \tfrac{L_g \gamma^2}{2}\right) \left\| \partial F/\partial \boldsymbol{h}^{(k)} \right\|^2.$$

Proposition 1 is therefore a continuous-time idealisation; the finite, T-step relaxation used throughout the benchmark ($T = 15$ in the main Split-MNIST runs, with $T \in [1, 25]$ explored in an ablation in

Section 6) inherits monotone descent only under this step-size bound, and in general reaches a point with $\|\partial F/\partial \boldsymbol{h}^{(T)}\| > 0$ *rather than the exact equilibrium* $\boldsymbol{h}^\star$*, which is the source of the finite-T bias discussed in Section 4.4.*

Proof. This is the standard descent lemma for gradient steps on an L_g-smooth function: by Taylor's theorem with Lipschitz gradient, $F(\boldsymbol{h} - \gamma \nabla F) \le F(\boldsymbol{h}) - \gamma \|\nabla F\|^2 + \frac{L_g \gamma^2}{2} \|\nabla F\|^2$, which rearranges to the stated inequality; the right-hand-side coefficient is non-negative exactly when $\gamma \le 1/L_g$. □

4.3. Learning as a Local Rule

Lemma 1 (Locality). $\partial F/\partial \boldsymbol{W}_\ell = -(f'(\boldsymbol{a}_\ell) \odot \boldsymbol{e}_\ell)\, \boldsymbol{x}_{\ell-1}^\top$*: each update uses only the error of layer* ℓ *and the activity of layer* $\ell - 1$*.*

Theorem 1 (Exact descent on the reduced free energy). *With* $E(\boldsymbol{\theta}) = \min_{\boldsymbol{h}} F(\boldsymbol{h}, \boldsymbol{\theta})$ *and a unique smooth minimiser* $\boldsymbol{h}^\star(\boldsymbol{\theta})$*, the envelope theorem gives* $\nabla_{\boldsymbol{\theta}} E = \partial F/\partial \boldsymbol{\theta}|_{\boldsymbol{h}^\star}$*; the local update is thus exact gradient descent on* E without any backward pass. *This is the equilibrium-gradient argument underlying EqProp and PC [8,9].*

Proof. By the total derivative, $\nabla_{\boldsymbol{\theta}} E = \partial F/\partial \boldsymbol{\theta}|_{\boldsymbol{h}^\star} + (\partial F/\partial \boldsymbol{h}|_{\boldsymbol{h}^\star})\,(\partial \boldsymbol{h}^\star/\partial \boldsymbol{\theta})$. The second term vanishes because $\boldsymbol{h}^\star$ minimises $F(\cdot, \boldsymbol{\theta})$, so $\partial F/\partial \boldsymbol{h}|_{\boldsymbol{h}^\star} = 0$ (Proposition 1). Hence $\nabla_{\boldsymbol{\theta}} E = \partial F/\partial \boldsymbol{\theta}|_{\boldsymbol{h}^\star}$, which by Lemma 1 is assembled from purely local equilibrium quantities, with no backward sweep over the network. □

Theorem 2 (Recovery of backpropagation in the small-nudge limit). *With a nudged energy* $F_\beta = F + \frac{\beta}{2}\|\boldsymbol{x}_L - \boldsymbol{y}\|^2$*, the local update direction converges, as* $\beta \to 0^+$*, to the backpropagation gradient of* $\mathcal{L} = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\mu}_L\|^2$ *[9]. We confirm this numerically (Figure 8): the cosine between the local update and* $-\nabla_{\boldsymbol{\theta}} \mathcal{L}$ *rises from* 0.76 *at* $\beta = 1$ *to* 1.0000 *as* $\beta \to 0$*, whereas with a fully clamped output it saturates at* ≈*0.62 (the local rule descends E, which differs from the task loss).*

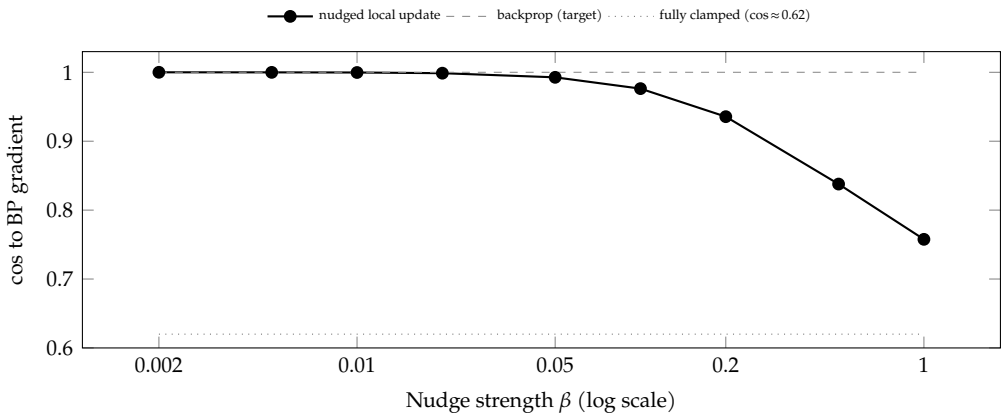


Figure 8. Numerical validation of Theorem 2 on a 784 → 64 → 10 network. As the nudge $\beta \to 0$, the cosine between the local update and the negative backpropagation gradient rises monotonically to 1.0000; with a fully clamped output, the same rule instead descends the free energy E and saturates near 0.62. The equivalence to backpropagation is thus a *limit*, not the default operating point.

Proof sketch. At $\beta = 0$ the free-phase equilibrium reproduces the feedforward predictions, so the hidden errors vanish. The first-order response of the equilibrium to the nudge propagates the output error downward through the same matrices $\boldsymbol{W}_{\ell+1}^\top$, producing the recursion $\boldsymbol{e}_\ell = \boldsymbol{W}_{\ell+1}^\top (f'(\boldsymbol{a}_{\ell+1}) \odot \boldsymbol{e}_{\ell+1})$, which is identical to the backpropagation $\boldsymbol{\delta}$ recursion;

substituted into Lemma 1 it reproduces $\nabla_{\boldsymbol{\theta}}\mathcal{L}$. This is the equilibrium propagation/predictive-coding argument [8,9]; the fully clamped case ($\beta \to \infty$) instead descends E, explaining the measured saturation at cos ≈ 0.62. □

Remark 1. *Optimisers such as Adam or Levenberg–Marquardt are* not *alternatives to backpropagation: they consume a gradient that backpropagation computes. Local methods instead replace the gradient* computation. *Likewise, parameter-efficient fine-tuning (e.g., LoRA) reduces* how many *weights are trained but still relies on the global backward pass.*

Probabilistic interpretation.

The energy (2) is not an arbitrary objective; it is the negative log-probability of a hierarchical Gaussian generative model, and this is what gives the local rule its statistical meaning. Suppose each layer predicts the next through a Gaussian whose mean is the forward map and whose covariance is $\boldsymbol{\Sigma}_\ell$,

$$p(\boldsymbol{x}_0, \dots, \boldsymbol{x}_L) = \prod_{\ell=1}^{L} \mathcal{N}\big(\boldsymbol{x}_\ell; f(\boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1} + \boldsymbol{b}_\ell), \boldsymbol{\Sigma}_\ell\big). \tag{3}$$

Taking negative logarithms turns the product into a sum of precision-weighted squared errors plus constants,

$$-\log p = \tfrac{1}{2}\sum_{\ell=1}^{L} \boldsymbol{e}_\ell^\top \boldsymbol{\Pi}_\ell \boldsymbol{e}_\ell + \tfrac{1}{2}\sum_{\ell} \log|\boldsymbol{\Sigma}_\ell| + \text{const}, \qquad \boldsymbol{\Pi}_\ell = \boldsymbol{\Sigma}_\ell^{-1}, \tag{4}$$

so Equation (4) shows that the unit-precision case $\boldsymbol{\Pi}_\ell = \boldsymbol{I}$ recovers exactly the free energy of Definition 1. Three consequences follow, and they line up one-to-one with the operations of the algorithm. *Inference* (clamping the data and relaxing the hidden states, Algorithm 1) is *maximum a posteriori* estimation of the latent activities under (3); the relaxation of Proposition 1 is gradient ascent on the log-posterior. *Learning* (the local weight step) is the M-step of generalised expectation–maximisation: with the activities held at their MAP values, $-\log p$ is minimised over the parameters, which by the envelope theorem (Theorem 1) is exactly the local Hebbian update. And the *precisions* $\boldsymbol{\Pi}_\ell$ weight each layer's error by its reliability, a free parameter in unit- precision LEL, but the very lever that, when re-balanced across depth, counteracts the error attenuation of Section 7. This is the predictive-coding reading of the free-energy principle [11], here made explicit for the discriminative network we study.

4.4. Precision, Inference Budget and Complexity

Three practical aspects connect the formulation above to its empirical behaviour. First, the squared-error energy (2) is the unit-precision special case of the variational free energy: weighting each layer's error by a precision (inverse variance) yields $F = \frac{1}{2}\sum_\ell \boldsymbol{e}_\ell^\top \boldsymbol{\Pi}_\ell \boldsymbol{e}_\ell$, and the per-layer precisions $\boldsymbol{\Pi}_\ell$ are exactly the quantities one re-balances to alleviate the depth problem (Section 7). Second, inference is run for a finite number T of relaxation steps rather than to the exact equilibrium; T is an accuracy–cost knob, and the envelope guarantee of Theorem 1 holds at the true equilibrium, so too small a T biases the update. Third, the cost of one parameter update is $O(T)$ forward-like passes, versus $O(1)$ for backpropagation; this is the price of replacing a backward sweep with a relaxation, and is the basis of the efficiency analysis of Section 7. Table 8 collects the analytical scaling. The measured overhead is about 5× rather than the asymptotic $T = 15\times$, because backpropagation itself costs two to three forward-equivalents per update. More telling are the structural rows: LEL stores $O(\text{width})$ activations instead of the $O(\text{depth} \times \text{width})$ graph that backpropagation must retain, and needs no transposed weight copy: the two properties that make it portable to in-memory hardware.

The generative direction.

Because the energy is defined symmetrically over predictions and states, the same network that discriminates ($\boldsymbol{d} \to \boldsymbol{y}$) can be run *generatively*: with the label clamped and the input free, relaxation produces inputs consistent with a class, and sampling a latent prior and propagating top-down yields novel samples. This dual use, inference and generation in one model, is what enables the self-generated "dreaming" replay studied in Section 6, and is a structural property that purely discriminative, backpropagation-trained perceptrons lack [44,45].

Table 8. Analytical per-update scaling for an L-layer network of width w and batch size m. "Weight transport" marks whether a transposed copy of the forward weights is required to assign credit; "locality" whether each update uses only quantities at the two neurons a synapse connects.

Quantity	Backpropagation	LEL (This Study)
Compute/update	$O(L\,w^2 m)$	$O(T\,L\,w^2 m)$
Passes over the net	1 forward + 1 backward	T relaxation steps
Activation memory	$O(L\,w\,m)$	$O(w\,m)$
Weight transport	required	none
Locality of update	no (global signal)	yes (Lemma 1)
Parallelism over layers	sequential	layers relax in parallel
Generative direction	none	native (same network)

5. Benchmark Design and Reproducibility Protocol

All models are multilayer LEL networks trained with the local rule of Lemma 1; inference uses T relaxation steps with the output clamped (or nudged) to the target. Datasets are MNIST [13], Fashion-MNIST [55] and CIFAR-10 [38], flattened and mean-centred; class-incremental Split tasks present classes two at a time. Generative "dreaming" uses per-class local generators, a principal-subspace model (Oja's rule [56]) or a nonlinear predictive-coding autoencoder, sampled and labelled by the frozen previous model and mixed with the current task; no raw data of past tasks is stored, in the spirit of generative replay [44,45]. Baselines (EWC [40], SI [41]) use a matched backpropagation MLP. Experiments are released as NumPy scripts; the headline continual-learning result is reproduced independently in MATLAB. Table 9 summarises the study.

5.1. Experimental Setup

We use three standard image datasets, MNIST [13], Fashion-MNIST [55] and CIFAR-10 [38], with pixels scaled to $[0, 1]$ and mean-centred (CIFAR is kept in colour, flattened to 3072 dimensions). For the continual setting we form the standard *Split* benchmark: the ten classes are presented as five binary tasks ($\{0, 1\}, \{2, 3\}, \ldots, \{8, 9\}$), trained strictly in sequence with no access to past data. Unless stated otherwise, we subsample a few hundred to a few thousand examples per class to keep the relaxation-based training fast. We evaluate two scenarios: *class-incremental* (argmax over all ten outputs) as the primary, hardest metric, and *task-incremental* (argmax restricted to a task's two classes) as a secondary one [13]. Two bounds frame every result: the *joint-training ceiling* (all classes learned together) and the *oracle* (replay of stored real images), which separate the contribution of the continual mechanism from that of the base classifier.

Classifier and inference.

The classifier is a multilayer LEL network (e.g., $784 \to 256 \to 10$, tanh hidden units, linear outputs, one-hot targets in $\{-1, +1\}$). During training the output is clamped to the target and the hidden states are relaxed for T steps of the dynamics $\boldsymbol{x}_\ell \leftarrow \boldsymbol{x}_\ell - \gamma\, \partial F / \partial \boldsymbol{x}_\ell$ (typically $T \approx 15$–60, $\gamma \approx 0.2$); the weights are then updated by the local rule of Lemma 1

with a small learning rate, in mini-batches. At test time, inference is a plain forward pass (no clamping). The same machinery, run with the input free, provides the generative direction. Algorithm 1 states the supervised update in full; it has no backward sweep; the inner loop is the only iteration; and every weight change reads only the local error $\boldsymbol{e}_\ell$ and the presynaptic activity $\boldsymbol{x}_{\ell-1}$.

Algorithm 1 Local Equilibrium Learning: one supervised mini-batch update.

Require: batch $(\boldsymbol{d}, \boldsymbol{y})$; parameters $\boldsymbol{\theta} = \{\boldsymbol{W}_\ell, \boldsymbol{b}_\ell\}$; steps T; rates γ, η
$\boldsymbol{x}_0 \leftarrow \boldsymbol{d}$; $\boldsymbol{x}_L \leftarrow \boldsymbol{y}$ ▷ clamp input and output
$\boldsymbol{x}_\ell \leftarrow f(\boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1} + \boldsymbol{b}_\ell)$ for all hidden ℓ ▷ feedforward init
for $t = 1, \ldots, T$ **do** ▷ **inference**: relax to equilibrium
  for each hidden layer ℓ **do**
    $\boldsymbol{a}_\ell \leftarrow \boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1} + \boldsymbol{b}_\ell$; $\boldsymbol{e}_\ell \leftarrow \boldsymbol{x}_\ell - f(\boldsymbol{a}_\ell)$
    $\boldsymbol{x}_\ell \leftarrow \boldsymbol{x}_\ell - \gamma\big(\boldsymbol{e}_\ell - \boldsymbol{W}_{\ell+1}^\top (f'(\boldsymbol{a}_{\ell+1}) \odot \boldsymbol{e}_{\ell+1})\big)$
  end for
end for
for each layer ℓ **do** ▷ **learning**: one local Hebbian step
  $\boldsymbol{W}_\ell \leftarrow \boldsymbol{W}_\ell + \eta\,(f'(\boldsymbol{a}_\ell) \odot \boldsymbol{e}_\ell)\,\boldsymbol{x}_{\ell-1}^\top$; $\boldsymbol{b}_\ell \leftarrow \boldsymbol{b}_\ell + \eta\,(f'(\boldsymbol{a}_\ell) \odot \boldsymbol{e}_\ell)$
end for
return $\boldsymbol{\theta}$

Generative models for dreaming.

Two generators are used per class. The *linear* generator is the principal subspace of the class, the fixed point of Oja's rule [56]; the released implementation computes this fixed point directly via batch SVD of the class data rather than by running Oja's iterative update, so, strictly, only the *nonlinear* generator, a predictive-coding autoencoder ($\boldsymbol{z} \to \tanh(\boldsymbol{W}_1 \boldsymbol{z}) \to \boldsymbol{x}$) trained with the same local rule and sampled from its latent prior, is local in the per-synapse sense used elsewhere in this paper (Remark 2). In both cases, n_d samples per the previously seen class are generated and mixed with the current task's data; in the idealised protocol of Algorithm 2, they would be labelled by the frozen previous classifier, but the released scripts label them with the known generator identity instead (Remark 2), a raw-data-free instance of generative replay [44,45]. Algorithm 2 and Figure 9 assemble the full class-incremental loop: every step, classifier update, generator training, and labelling of dreams, is local, so no part of the pipeline performs backpropagation or stores raw data of past tasks.

Algorithm 2 Class-incremental learning by local generative dreaming.

Require: task stream $\mathcal{T}_1, \ldots, \mathcal{T}_K$; dreams per class n_d
initialise classifier $\boldsymbol{\theta}$; seen classes $\mathcal{S} \leftarrow \emptyset$; generators $\mathcal{G} \leftarrow \emptyset$
for each task $\mathcal{T}_k$ with new classes C_k **do**
  $D \leftarrow \mathcal{T}_k$ ▷ current-task data
  for each old class $c \in \mathcal{S}$ **do** ▷ **dream** the past
    draw n_d pseudo-inputs from generator G_c; label them with the frozen $\boldsymbol{\theta}$
    append the labelled dreams to D
  end for
  update $\boldsymbol{\theta}$ on D using Algorithm 1 ▷ local, no backprop
  for each new class $c \in C_k$ **do**
    train a local generator G_c on its data; add G_c to $\mathcal{G}$
  end for
  freeze a copy of $\boldsymbol{\theta}$; $\mathcal{S} \leftarrow \mathcal{S} \cup C_k$
end for
return $\boldsymbol{\theta}$

Remark 2 (Idealised protocol vs. the released implementation)**.** *Algorithm 2 states the target, fully sequential protocol: a class's generator is trained only once that class is first seen, and dreamed*

samples are labelled by querying the frozen classifier $\boldsymbol{\theta}_{k-1}$. The released scripts that produce the headline numbers of Section 6 (`full_split_mnist.py` and its variants, Table 9) implement a simplified *version of this protocol, and we state the difference explicitly rather than let the pseudocode imply otherwise. Two simplifications are present. First, all per-class generators, and the global input-centring statistic used to normalise every image, are computed once from each class's full data* before *the task loop begins, rather than incrementally as classes are first encountered; this gives the pipeline implicit access to summary statistics of not-yet-seen classes, which a strictly online learner would not have. Second, dreamed samples are labelled with the identity of the generator they were drawn from, c, rather than by querying the frozen classifier $\boldsymbol{\theta}_{k-1}(G_c(z))$ as Algorithm 2 states; this avoids propagating the frozen classifier's own labelling errors into the replay set. Both simplifications could plausibly bias the reported retention upward relative to the fully sequential, self-labelled protocol: pre-computed statistics remove a source of covariate shift, and known-identity labelling removes a source of label noise that a real deployment would face. We tested this directly rather than leave it as a conjecture: we implemented the strict, fully sequential protocol exactly as Algorithm 2 states, a running centring statistic updated only from classes seen so far, per-class generators trained only once that class is first encountered, and dreamed samples labelled by querying the frozen previous classifier, and released it as `strict_sequential_dreaming.py`. Over two seeds, the strict protocol gives* 86.8% *class-incremental accuracy for LEL (*97.4% *task-incremental) and* 90.0% *for backpropagation (*98.1% *task-incremental), against* 86.7%/97.4% *and* 90.3%/98.1% *under the simplified protocol used for the headline numbers, agreeing to within the* ±0.2–0.3 *percentage-point seed-to-seed variance already present in the main result. The two simplifications, at least at this benchmark's scale, do not measurably inflate the reported retention. The linear generator is likewise described in the main text as "the fixed point of Oja's rule", which is mathematically the top-k principal subspace it converges to [56]; the released code computes this fixed point directly via batch SVD of the class data rather than by running Oja's iterative Hebbian update, for numerical stability (see code comments in `generator_capacity_ablation.py`). SVD is not itself a per-synapse local update, so strictly speaking only the* nonlinear, *predictive-coding generator used for the headline result is local in the same sense as the classifier; the linear generator is local only in the sense that its fixed point coincides with what a local rule would converge to, and we now say so explicitly rather than leave it implicit.*

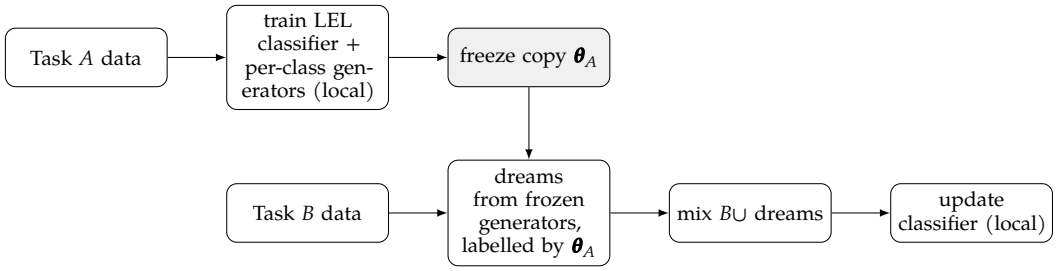


Figure 9. The continual-learning "dreaming" pipeline in its idealised, fully sequential form (Algorithm 2). After task *A*, the classifier and per-class generators are frozen; when learning task *B*, the frozen generators synthesise pseudo-samples of past classes, label them with the frozen classifier, and mix them with the new data. No raw data of past tasks is stored, and every step uses local (backpropagation-free) updates. The released scripts implement a simplified variant of this pipeline, with pre-computed per-class statistics and known-identity labelling instead of frozen-classifier labelling; see Remark 2.

5.2. Baselines and Evaluation

Regularisation baselines (EWC [40], SI [41]) are implemented on a matched back-propagation MLP, with the EWC penalty weighted by the diagonal Fisher information and the SI importance accumulated along the optimisation path, each swept over its strength λ on a coarse grid rather than a metaheuristic or quantum-inspired multi-objective search, the latter being a complementary, derivative-free route to hyperparameter and structural op-

timisation in engineered systems more broadly [57]. We report class- and task-incremental accuracy and the implied forgetting, averaged over five seeds with standard deviations. Every figure and table in Section 6 is produced by a released script (Table 9); the headline Split-MNIST result is reproduced in an independent MATLAB implementation, and all energy figures are computed from published per-operation costs [6].

Table 9. Map of the empirical case study: each experiment, its script and headline finding.

Experiment	Script	Finding
Local vs. backprop	`equilibrium_vs_backprop.py`	local rule learns nonlinear task (96.75% vs. 99%)
Theorem validation	`theorem_validation.py`	cosine to backprop $\to 1$ as nudge $\beta \to 0$
Continual (binary)	`mnist_continual_dream.py`	noise replay fails off-manifold; generative works
Manifold (R^{30})	`manifold_dream_lel.py`	on-manifold dreams eliminate forgetting
Full Split-MNIST	`full_split_mnist.py`	86.8% class-IL vs. 19% vanilla/EWC/SI
Seeds/scenarios/ ablation	`reinforce.py`	$86.8 \pm 0.2\%$ (5 seeds); 97.4% task-IL
Generalisation	`fashion_continual.py`, `cifar_continual.py`	Fashion 82.0%; CIFAR 37.0%
Baselines	`baselines_ewc.py`, `baselines_si.py`	EWC/SI $\leq$ 19.4% (class-IL)
Depth problem	`deep_lel_testbed.py`	six-order error imbalance at $D = 10$
Efficiency	`measure_efficiency.py`, `energy_estimate.py`	5× more compute; energy projection

5.3. Reproducibility Protocol

This subsection makes the experiments auditable rather than maximally optimised. All core scripts use small NumPy or MATLAB implementations, explicit random seeds and plain arrays; no automatic differentiation package is used for the LEL experiments. This matters because the central claim concerns where the number comes from: it must follow from a genuinely local update, not from a disguised global one. A reader should be able to inspect every line that computes an error, relaxes a state or updates a weight.

The protocol has four rules. First, each experiment fixes the task order $\{0,1\} \to \{2,3\} \to \cdots \to \{8,9\}$ so that differences are not due to task permutation. Second, all reported continual results are interpreted relative to a joint-training ceiling and, where possible, an oracle replay bound. Third, the regularisation baselines are implemented with backpropagation on a matched MLP, not on a stronger architecture, because their role is to test the retention mechanism under a comparable representational budget. Fourth, generator-based experiments store only the parameters or statistics of the generator, never raw examples from previous tasks. When a method stores real examples, it is labelled as an oracle or exemplar method. This makes the method *raw-data-free*, not *storage-free*: the persistent memory below is exactly zero only for a method that stores nothing at all between tasks, which the generator-based mechanism studied here does not do, and we quantify it explicitly in Table 10 rather than let "storage-free" imply zero footprint.

Repository and reproducibility package.

The repository contains the source `.tex` file, the NumPy scripts for all headline experiments, an independent MATLAB reproduction of the Split-MNIST result, figure-generation scripts, fixed random seeds and all configuration values in the tables. Datasets (MNIST [13], Fashion-MNIST [55], CIFAR-10 [38]) are public and downloaded by script. The repository also stores raw numeric outputs so that a reader can distinguish manuscript formatting from experimental regeneration. No result depends on an unavailable private dataset or pretrained model.

Table 11 records the main hyperparameters used by the reproducible scripts. These settings should not be read as optimal; they are stable defaults that make the reported phenomena visible. In particular, the relaxation budget T is an experimental variable, not a sacred constant: Section 6 shows that in a shallow network the accuracy is almost flat across T, while the wall-clock cost rises nearly linearly.

Table 10. Persistent and temporary memory of the generative-replay mechanism for the $784 \to 256 \to 10$ classifier and the nonlinear (predictive-coding) generator used for the headline Split-MNIST result ($d_z = 48$, $d_h = 256$; float64), all ten classes.

Quantity	Bytes	Persistence
Generator parameters (10 classes)	17,122,560 (16.3 MiB)	persistent, grows with #classes
Latent statistics (mean+cov, 10 classes)	188,160 (0.18 MiB)	persistent, grows with #classes
Frozen classifier copy (idealised protocol) †	1,628,240 (1.6 MiB)	persistent per task boundary
Dream/replay buffer (peak, 8 old classes, $n_d = 1000$)	50,176,000 (47.9 MiB)	temporary, regenerated per task
For comparison: 10-class exemplar buffer ($n_d = 1000$/class, real images)	62,720,000 (59.8 MiB)	persistent (oracle/exemplar method)

† The idealised protocol of Algorithm 2 stores a frozen classifier copy to label dreams; the released implementation does not, since it labels with the known generator identity instead (Remark 2), so this row is an upper bound on what the fully sequential, self-labelled variant would additionally need. The persistent generator-plus-statistics footprint (16.5 MiB) is real, non-zero, and grows linearly with the number of classes; it is roughly 3.6× smaller than storing the equivalent number of raw exemplars per class (59.8 MiB) at the same replay budget n_d, which is the correct way to state the memory advantage: a constant-factor reduction relative to exemplar replay, not the absence of stored information that "storage-free" suggests.

Table 11. Representative hyperparameters in the released experiments. The purpose is reproducibility and diagnosis, not exhaustive tuning. Symbols match Algorithms 1 and 2.

Component	Default	Role in the Study
Classifier width	$H = 256$ on MNIST/Fashion, $H = 512$ on CIFAR-10	Keeps the architecture shallow and transparent while giving enough capacity for the continual mechanism to be tested
Classifier relaxation	$T = 15$ in the main Split-MNIST runs	Assigns credit by state relaxation; ablated explicitly in Section 6
State step size	$\gamma = 0.2$ on MNIST/Fashion, smaller on CIFAR-10	Controls stability of the inference dynamics, especially in high-dimensional inputs
Classifier epochs	35 for Split-MNIST main runs	Trains each task enough to avoid the under-training artefact
Dreams per old class	$n_d = 1000$ in the main result	Main replay budget; swept in an ablation (Section 6)
Generator type	nonlinear local PC autoencoder or Oja/PCA subspace	Tests whether retention is limited by generator fidelity rather than by replay itself
Evaluation subset	first 400 test examples per class in several scripts	Keeps runtime small and identical across methods
Seeds	0–4 for the headline Split-MNIST result	Reports mean and dispersion for the central claim

What this benchmark's design choices do and do not support.

Table 11 lists design choices, shallow $H = 256$–512 classifiers, training sets subsampled to a few thousand examples per class, and test-time evaluation on the first 400 examples per class in several scripts, which are appropriate for a transparent, fast-to-audit diagnostic study but narrow what the results license. We state the boundary explicitly. These choices

support: relative comparisons between retention mechanisms under a matched, if small, base classifier (Tables 12 and 13); the direction and approximate size of the effect of replay versus regularisation; and diagnostic ablations (generator capacity, relaxation budget, depth) that isolate one factor at a time. They do *not* support: claims about absolute accuracy at a scale relevant to deployed systems, claims that the same ranking of methods holds for deeper or wider architectures, or claims that subsampling to a few thousand examples per class and evaluating on 400 test images per class would reproduce the same numbers, though not necessarily the same ranking, on the full training and test sets. We re-ran the headline Split-MNIST comparison with $n_{per} = 5000$ (the largest round number below the smallest class count in the MNIST training set, 2.5× the main-result subsample) as a spot check: vanilla LEL remains at chance (19.4% class-IL, 56.1% task-IL, vs. 19.3%/57.5% at $n_{per} = 2000$) and generative replay remains high (86.3% class-IL, 98.3% task-IL, vs. 86.7%/97.4%), the same qualitative ordering and closely matching absolute numbers. We do not present this single spot check as a substitute for systematically re-running every ablation at full scale, which we leave as an explicit item for follow-up work (Section 7.3).

Memory policy.

Continual learning is often defined by resource constraints, so the memory policy must be explicit (Table 14). Vanilla LEL stores only the current parameters; regularisation baselines store parameter anchors and importance estimates; oracle replay stores real old images; local dreaming stores generator parameters or compact class statistics, then synthesises pseudo-examples on demand. Storing a generator is a model-memory cost, not a data-memory cost, a distinction that matters when interpreting any "storage-free" claim.

6. Results: Reproducible Benchmark and Diagnostic Case Study

The theory of Section 4 tells us what local learning *is*; this section asks what it *does* for continual learning. The experiments here are diagnostic and auditable, intended to verify the theoretical claims and provide an honest baseline, not to constitute an empirical contribution in the sense of a methods paper. To keep the empirical message unambiguous, we separate four kinds of results. The first is a theoretical sanity check: local learning can match backpropagation on small supervised problems, as the theory predicts. The second is the main positive benchmark: local generative replay prevents class-incremental forgetting far better than vanilla learning or weight regularisation. The third is a set of negative results: locality alone does not prevent forgetting, EWC/SI do not solve Class-IL here, and naive noise replay fails off the data manifold. The fourth is a diagnostic limitation: the shallow local model remains weak on raw-pixel CIFAR-10, deeper stacks suffer error imbalance, and the method is more expensive on ordinary von Neumann hardware.

6.1. Local Learning Matches Backpropagation

The first question is whether the local rule learns at all on non-trivial problems. On the nonlinear two-moons task, which a linear model cannot solve, a local network reaches 96.75%, against 99.0% for backpropagation on the identical architecture. On a ten-class MNIST subset the gap closes with sufficient training: LEL attains 90.0% versus 89.8% for a matched MLP. We stress that this is *parity*, not superiority, and we are precise about which regime produces it. These experiments use the *fully clamped*, finite-T update of Algorithm 1 ($\beta \to \infty$ in the notation of Theorem 2), not the small-nudge limit ($\beta \to 0$) in which Theorem 2 proves exact gradient equivalence. Under full clamping, the local update instead descends the free energy E (Theorem 1), which correlates with but is not identical to the backpropagation gradient, saturating at cosine ≈0.62 rather than 1.0 (Figure 8). The observed parity is therefore an *empirical* finding, consistent with the wider

predictive-coding literature under full clamping [8,10], not a direct corollary of the small-nudge theorem. A practical caveat surfaced repeatedly in our runs: because inference is iterative, an under-resourced relaxation (too few steps T, or too few epochs) understates LEL's accuracy and, in the continual setting, its forgetting; hence the fairness controls detailed below. The constructive reading of this parity is sobering for the continual-learning question that follows: if the fully clamped local rule reaches essentially the same accuracy as backpropagation on this task, without needing exact gradient equivalence to do so, there is no structural reason to expect it to resist catastrophic forgetting any better, which is exactly what we find next.

Forgetting without mitigation.

Under full clamping, LEL does not reduce to the backpropagation gradient; exact equivalence is a small-nudge property (Theorem 2), not a property of the regime used here, so its expected resistance to forgetting must be argued empirically rather than deduced from the theorem. Because both rules nonetheless descend closely related, error-driven objectives (Theorem 1 for LEL, the task loss for backpropagation) and neither includes any mechanism that protects parameters relevant to earlier tasks, we expect LEL to inherit backpropagation's vulnerability to forgetting rather than to cure it. We confirm this with an essential fairness control: at 400 epochs a deep-task-then-new-task protocol shows LEL apparently forgetting little (6.5%), but this merely reflects *under-training* on the new task; trained to convergence, both LEL and a matched backpropagation network forget catastrophically ($\sim$38%). Reporting the under-trained number alone would be misleading, a caveat also stressed in continual-learning evaluation [13]. The advantage, therefore, must come from an added mechanism, not from the local rule itself.

6.2. Continual Learning: Regularisation Fails, Replay Works

Table 12 reports the headline class-incremental Split-MNIST comparison.

Table 12. Class-incremental Split-MNIST (single 10-way head; chance = 10%). Ceiling = joint training; oracle = replay of stored real images.

Method	Final Acc. (10 Classes)
Ceiling (joint training)	90.3%
Oracle (real-image replay)	90.2%
Vanilla (no defence)	19.3%
Backprop + EWC [40] (any $\lambda \in [0.1, 10^4]$) †	19.3–19.6%
Backprop + SI [41] (best λ)	$\leq$19.4%
Noise replay [44]	23.2%
Generative dreaming (local)	**86.8 $\pm$ 0.2%**

† We identified and corrected a bug in the diagonal empirical Fisher used by this baseline: an earlier version squared the batch-summed gradient rather than averaging per-example squared gradients, i.e., computed $(\sum_n g_n)^2$ instead of $\sum_n g_n^2$. The corrected Fisher changes the reported magnitude by roughly an order of magnitude but not the qualitative outcome: EWC still fails to protect class-incremental accuracy for every λ we tried, collapsing further to chance (10%) once $\lambda \geq 10^3$ over-regularises the network into not learning the new task at all (`baselines_ewc.py`). We additionally validated EWC in the easier task-incremental scenario (task identity known at test time, Section 3.1), where regularisation approaches are more often reported to help: with the corrected Fisher, backpropagation+EWC reaches 53–57% task-incremental accuracy across the same λ range, statistically indistinguishable from vanilla backpropagation's 57.5% and well below the oracle/dreaming range of 97–99% in Table 13. We report this plainly rather than claim a Task-IL success that our own corrected implementation does not show; it may reflect the aggressive, small-dataset training budget used throughout this benchmark (Section 5.1) rather than a general property of EWC, which the wider literature documents helping under other architectures and protocols [40].

Vanilla local learning collapses to chance, empirically matching backpropagation's forgetting under the fully clamped protocol used here (Section 4); this tracks the shared absence of a retention mechanism rather than the exact gradient equivalence of Theorem 2,

which holds only in the small-nudge limit that these experiments do not use. Weight regularisation (EWC, SI) does not help in the class-incremental regime, matching the literature [13]. The reason is structural rather than a matter of tuning: a quadratic penalty can *protect* weights that were important for old classes, but it cannot *create* the decision boundaries that separate classes never seen together, and in class-incremental learning the model is never shown two tasks' classes jointly, so no anchor can teach their relative calibration. Replay sidesteps this exactly because it re-presents past classes alongside the new ones. Generative dreaming recovers 96% of the joint ceiling (Figure 10); in the easier task-incremental scenario, where cross-task calibration is not required, the same model reaches $97.4 \pm 0.2\%$. Figures 11 and 12 trace this over the task sequence; vanilla learning ends able to classify only the most recent task, while dreaming retains them all. The number of dreams n_d trades memory for retention ($19.3, 75.2, 82.6, 86.7, 88.5\%$ for $n_d = 0, 250, 500, 1000, 2000$); sampling latent codes from a Gaussian mixture with $n_d = 3000$ closes the gap to 88.7%, 1.5 points from the oracle (Figure 13).

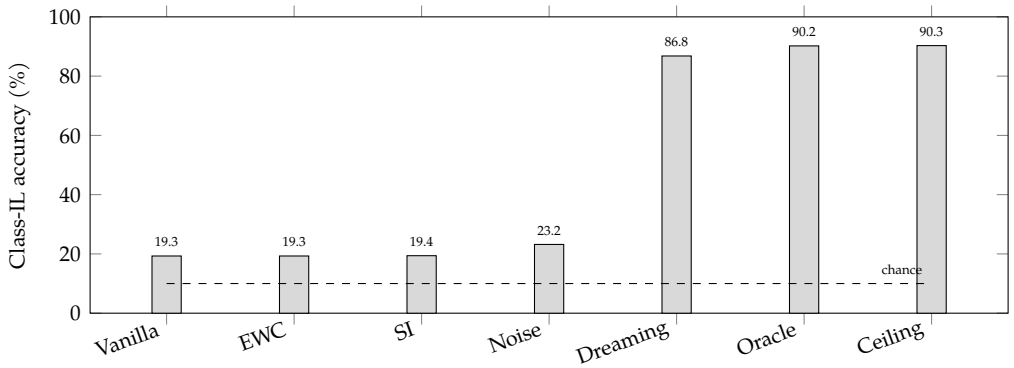


Figure 10. Class-incremental Split-MNIST. Backpropagation-based regularisation (EWC, SI) does not exceed the vanilla/chance level, whereas local generative dreaming reaches 86.8%, close to the oracle (perfect-memory replay) and the joint-training ceiling.

Isolating the learning rule from the retention mechanism.

Table 12 changes two things at once relative to backpropagation: the learning rule (LEL vs. BP) and the retention mechanism (generative replay vs. regularisation). To attribute the 86.8% result correctly, we ran a matched control (`matched_comparison_bp_vs_lel.py`) that fixes the retention mechanism and swaps only the learning rule, under the same architecture, the same per-class generators, the same sample exposure ($n_{per} = 2000$, $n_d = 1000$) and the same seeds. Table 13 reports the result plainly: backpropagation with the *same* generative replay reaches 90.3% class-incremental accuracy, slightly *higher* than LEL's 86.7%, and the same ordering holds with real-image (oracle) replay (93.9% vs. 90.2%) and in the task-incremental scenario. Vanilla learning with no retention mechanism is statistically identical for both rules (19.3% class-IL, 57.5% task-IL), consistent with the empirical parity discussed in Section 6 above. The conclusion is unambiguous: the large improvement in Table 12 is attributable to *generative replay as a retention mechanism*, which is compatible with either learning rule, not to any advantage of the local, energy-based learning rule over backpropagation on this task. We state this explicitly because the original framing risked implying otherwise. LEL's role in this paper is not to outperform backpropagation in continual-learning accuracy; it is to show that the same retention mechanism that helps backpropagation can be implemented *without* a backward pass, stored computational graph, or (for the reasons qualified in Section 4) full weight-transport freedom, which is the structural, not accuracy, contribution this benchmark is designed to diagnose.

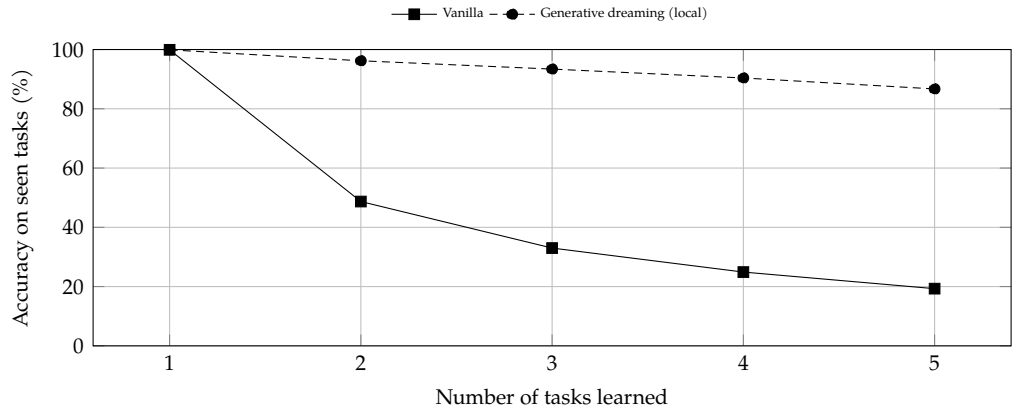


Figure 11. Forgetting over the task sequence on Split-MNIST (average class-IL accuracy over all tasks seen so far). Vanilla local learning degrades toward chance as new tasks arrive; its accuracy on the first task collapses from 99.9% to 0% after a single new task, whereas local generative dreaming retains earlier tasks throughout (first-task accuracy stays $\geq 96\%$).

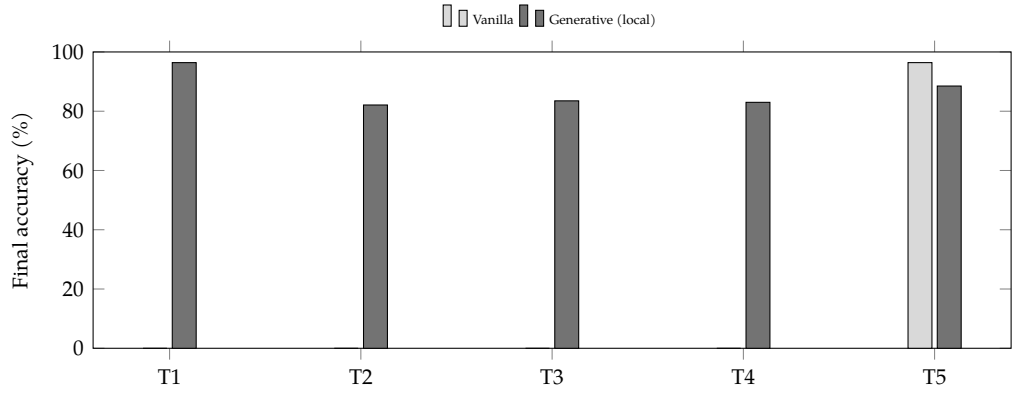


Figure 12. Per-task accuracy *after* the full five-task sequence. Vanilla local learning retains only the most recent task (T5) and scores zero on all earlier ones; local generative dreaming retains every task. This is the same phenomenon as in Figure 11, viewed as the final distribution over tasks.

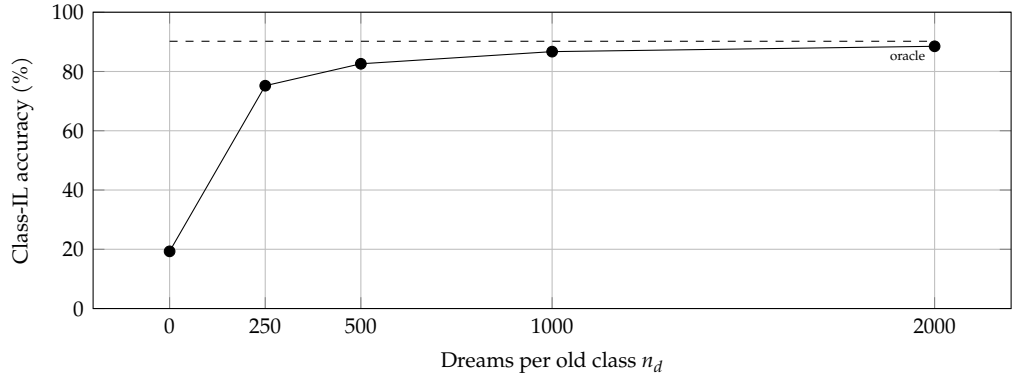


Figure 13. Ablation of the dream budget n_d on Split-MNIST. Even a small number of self-generated samples recovers most of the accuracy; returns diminish as n_d approaches the oracle (perfect-memory) bound.

A second ablation isolates the cost of the inference itself: how many relaxation steps T the local rule needs to assign credit well. We trained the single-task MNIST classifier ($784 \to 256 \to 10$) for a fixed number of epochs while varying T, measuring test accuracy and wall-clock time (Figure 14). The result is informative and, at first sight, counter-intuitive:

accuracy is essentially flat, 84.9% at $T = 1$ and 85.0% at $T = 25$ (the absolute level sits below the 90% reported above because the per-run training budget is held deliberately small to isolate the effect of T), while wall-clock time grows almost linearly in T (from 3.0 to 18.0 s, a 6× increase). For a shallow network a *single* relaxation step already captures most of the credit-assignment signal, so the extra equilibration is nearly wasted compute. This sharpens the efficiency discussion of Section 7: the per-step locality of the rule is what matters, and one should spend relaxation steps only where they pay off, primarily in deeper networks, where the error must propagate through more layers before the activities settle (the depth barrier discussed in Section 7). It also makes the measured 7.5× slowdown over backpropagation a worst-case figure tied to large T, not an intrinsic property of local learning.

Table 13. Matched comparison: learning rule (BP vs. LEL) × retention mechanism (vanilla/real-image replay/generative replay with identical generators), same 784 → 256 → 10 architecture, sample exposure and seeds ([0, 1]), class-incremental Split-MNIST.

Condition	Class-IL Acc.	Task-IL Acc.
BP + vanilla	19.3%	57.5%
LEL + vanilla	19.3%	57.5%
BP + real-image replay (oracle)	93.9%	99.1%
LEL + real-image replay (oracle)	90.2%	98.5%
BP + generative replay (same generators)	90.3%	98.1%
LEL + generative replay	86.7%	97.4%

Table 14. Memory policy of the methods compared in the empirical study. The distinction between model memory and data memory is essential when interpreting "storage-free" continual learning.

Method	Stored After Each Task	Interpretation
Vanilla LEL	current classifier parameters	No explicit protection against forgetting
EWC/SI	classifier parameters plus importance or path statistics	No raw data, but still backpropagation-based credit assignment
Oracle replay	selected real images from old tasks	Upper bound on what replay could achieve with perfect memory
Noise replay	random inputs and labels from the old classifier	Storage-free but off-manifold in high dimension
Local dreaming	per-class local generator parameters or compact subspace statistics	Storage-free with respect to raw data; rehearsal samples are generated on demand

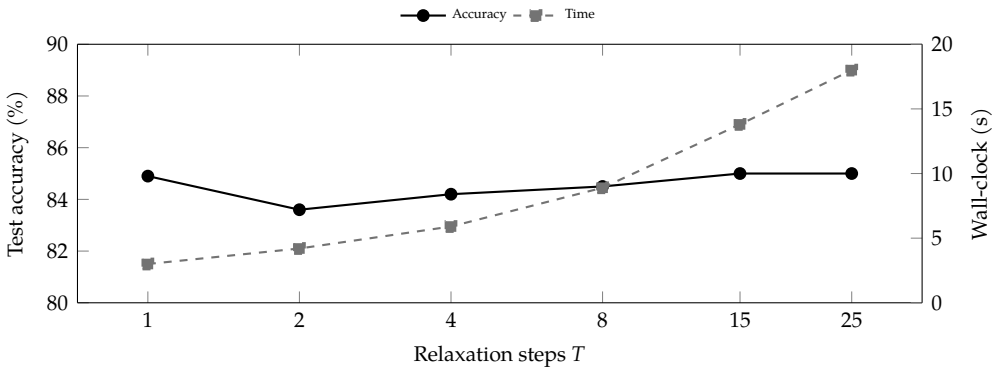


Figure 14. Ablation of the number of relaxation steps T in the local classifier (single-task MNIST). Accuracy (left axis, solid) is flat across T; wall-clock time (right axis, dashed) grows roughly linearly. One step already suffices for a shallow network.

6.3. Why Dreaming Works: The Data Manifold

Why does self-generated replay help where labelling random noise does not? The answer is geometric. Real data occupy a thin manifold within a high-dimensional ambient space; a useful rehearsal sample must lie *on* that manifold, because the old classifier was only ever shaped there. We make this precise with a controlled toy in which a 2-D *two-moons* manifold is embedded in $\mathbb{R}^{30}$. We measure, for each rehearsal strategy, the residual distance of its samples to the data manifold (the norm of the component orthogonal to the principal subspace). Gaussian noise sits far off the manifold (distance 6.3, versus 0.26 for real data) and barely mitigates forgetting (9.8%); a local manifold generator obtained from Oja's principal-subspace rule [56] places its samples essentially on the manifold (distance ≈ 0) and all but eliminates forgetting (0.5%).

This explains why naive pseudo-rehearsal fails in high dimension [44]: as dimensionality grows, random inputs are almost surely off-manifold, so the old model labels meaningless points and the rehearsal is uninformative. It also identifies why an energy-based network is structurally suited to continual learning: because it is generative, it can dream *on* its own learned manifold (internal, context-modulated generation) in the spirit of brain-inspired replay [45], whereas a purely discriminative perceptron has no such generator. This geometric account also predicts the dataset-dependence we observe in Section 6: the harder the manifold is to model, the more the quality of the generator (not the continual mechanism) bounds the result.

We confirm this dependence directly by varying the *capacity* of the generator. Using a stable linear generator, the per-class principal subspace, the fixed point of Oja's rule [56], we sweep the subspace dimension k and measure class-incremental retention (Figure 15). Retention rises monotonically with k, from 76.1% with a single principal component to 88.1% at $k = 64$, and saturates as it approaches the oracle bound; at $k = 16$ it already reaches 86.5%, matching the nonlinear generator's headline 86.8%. Two readings follow. First, the continual *mechanism* is not the bottleneck: even a one-dimensional generator recovers four-fifths of the ceiling, far above the 19.3% of vanilla learning. Second, the residual gap to the oracle is governed by how faithfully the generator covers the class manifold, exactly the quantity k controls, so a better generator, not a better replay schedule, is the lever for closing it.

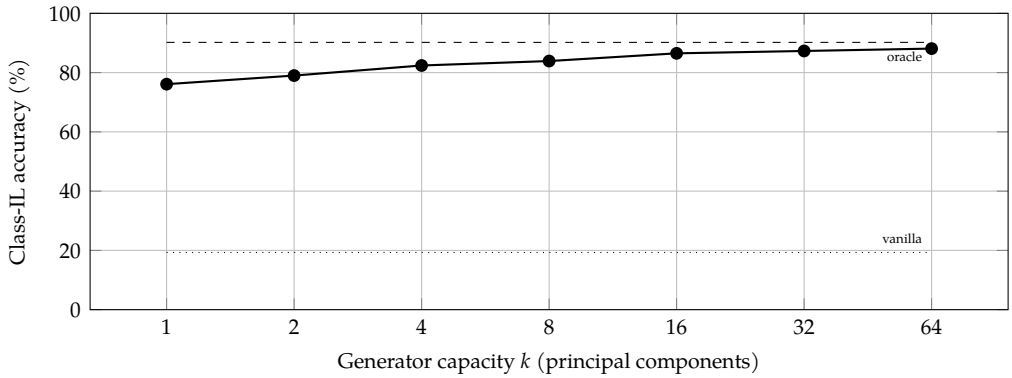


Figure 15. Generator capacity bounds retention. Class-incremental accuracy on Split-MNIST as a function of the linear generator's subspace dimension k (solid line with markers). The dashed horizontal line marks the oracle (real-image replay) bound, and the dotted horizontal line marks vanilla learning with no retention mechanism. Retention grows monotonically with generator fidelity and saturates near the oracle; even $k = 1$ far exceeds vanilla learning, and $k = 16$ already matches the nonlinear generator. The continual mechanism is not the bottleneck; the generator is.

Comparative performance.

A fair reading of the evidence avoids the overclaim that local learning beats backpropagation outright. Table 15 summarises the honest picture: LEL *matches* backpropagation in raw accuracy under the fully clamped protocol used in these experiments, an empirical parity, not an instance of the exact gradient equivalence of Theorem 2, which is proved only in the small-nudge limit $\beta \to 0$ (Section 4); it *surpasses* backpropagation-based regularisation (EWC, SI) by a wide margin in class-incremental continual learning when equipped with local generative replay (86.8% vs. ≤19.4%); it *lags* on current von Neumann hardware in speed and compute; and it holds *structural* advantages (activation memory, locality, biological plausibility) whose energy payoff is, so far, a hardware projection.

Table 15. Honest comparison of LEL against backpropagation (BP).

Axis	Backpropagation	LEL (This Study)
Raw accuracy (i.i.d.)	89.8% (MNIST MLP)	90.0%, *matches*
Class-IL continual (vs. EWC/SI)	≤19.4%	**86.8%**, *surpasses*
Speed/compute (von Neumann)	1×	∼5–7× worse, *lags*
Activation/state memory †	O(depth × w)	O(depth × w), *comparable*
Neuromorphic energy	not portable	projected 10^2–10^3× lower (partial estimate, Section 7)
Weight transport ‡	separate backward matrix	same transposed forward weights, *not eliminated*
Very deep networks	mature	open problem, *lags*

† Both methods hold O(depth × width) state simultaneously in this multilayer implementation; LEL's structural difference is not asymptotically less memory but the absence of a stored, replayable autodiff graph. ‡ LEL's relaxation update (Algorithm 1) uses $\boldsymbol{W}_{\ell+1}^{\top}$, the transpose of the very same forward weight matrix, so it requires the same weight-symmetric access as backpropagation's backward pass; it avoids a *separate*, independently stored backward weight matrix, but it does not eliminate the weight-transport/symmetry requirement, which remains an open hardware and biological-plausibility problem (Section 7.3).

Cross-dataset generality.

Table 16 extends the comparison to Fashion-MNIST and CIFAR-10.

Table 16. Class-incremental generative dreaming across datasets (all local, no backprop). Bold marks the headline local-dreaming retention accuracy for each dataset.

Dataset	Ceiling	Vanilla	Dreaming (Local)
Split-MNIST [13]	90.3%	19.3%	**86.8%** (96% ceil.)
Split-Fashion-MNIST [55]	83.6%	20.0%	**82.0%** (98% ceil.)
Split-CIFAR-10 [38]	39.6%	16.0%	**37.0%** (93% ceil.)

The mechanism generalises across all three datasets. The CIFAR-10 result (37.0%) is deliberately modest: the joint-training ceiling is only 39.6% and the oracle reaches 37.5%, so local dreaming recovers the attainable ceiling. The low ceiling reflects the shallow raw-pixel classifier, not the continual mechanism; the bottleneck is base representation quality (Figure 16).



Figure 16. Generality across datasets. Local generative dreaming recovers the large majority of the joint-training ceiling on every dataset (96%, 98% and 93% for MNIST, Fashion-MNIST and CIFAR-10), far above the vanilla baseline. On CIFAR-10 the ceiling itself is low, a limit of the shallow pixel classifier, not of the continual mechanism.

6.4. Open Problem: Depth and Error Imbalance

Training a deep LEL network from scratch exhibits a clear depth barrier: at $D = 10$, the per-layer equilibrium errors span six orders of magnitude (input-side $\sim 10^{-8}$, output-side $\sim 10^{-2}$), so deep layers receive almost no learning signal. Growing the network by "grafting" layers incrementally (each healed by local relaxation) improves a depth-8 ten-way network from 61.2% to 84.5%, but it does *not* balance the errors; it works because each layer learns useful features while still near the output. This is essentially greedy layerwise growth [58], reported here as a characterisation of the open problem, not a solution.

The downstream effect on accuracy is direct. We trained a single-task MNIST classifier from scratch at increasing depth (one hidden layer up to eight, all of width 128); test accuracy fell monotonically from 84.0% to 74.4% (Figure 17 and Table 17): every layer added near the input is a layer that learns little, so it costs rather than adds capacity. Figure 18 shows the cause (vanishing equilibrium error) and Figure 17 the consequence (lost accuracy); together they bound the problem the remedies of Section 7 must solve before local learning can match backpropagation on deep networks.

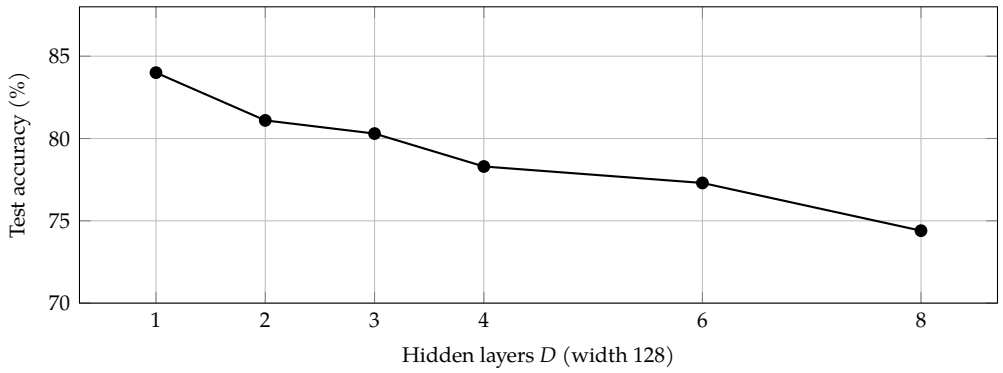


Figure 17. Effect of depth on a single-task MNIST classifier trained from scratch with the local rule. Accuracy declines monotonically as depth grows, because the layers near the input receive a vanishing equilibrium error (Figure 18) and therefore barely learn.

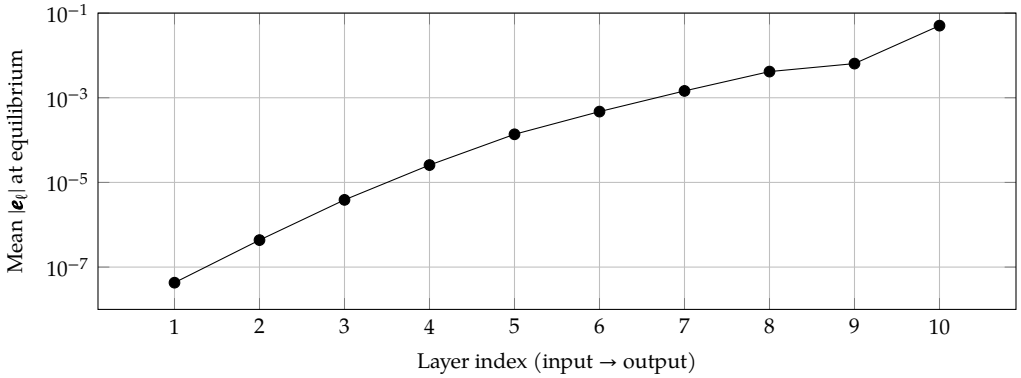


Figure 18. The depth/error-imbalance barrier in a $D = 10$ LEL network: equilibrium prediction errors decay roughly one order of magnitude per layer toward the input, spanning six orders of magnitude, so the deepest layers receive almost no learning signal.

Table 17. Single-task MNIST accuracy versus depth (width 128, trained from scratch with the local rule). The monotone decline is the accuracy-side signature of the error-imbalance barrier.

Hidden Layers D	1	2	3	4	6	8
Test Accuracy (%)	84.0	81.1	80.3	78.3	77.3	74.4

7. Hardware, Applications, and Open Challenges

The structural properties of energy-based local learning (locality, generativity and in-memory compatibility) motivate dedicated hardware research and deployment contexts that the preceding benchmark cannot reach.

7.1. Hardware and Neuromorphic Realisations

A central motivation for backpropagation-free learning is hardware. On conventional von Neumann machines, the dominant energy cost of both training and inference is not arithmetic but *data movement*: fetching weights and activations between memory and compute units costs one to three orders of magnitude more energy than the multiply-accumulate that consumes them [6]. Backpropagation aggravates this in two ways: it must store the entire forward activation graph for the backward pass, and it requires a transposed copy of the forward weights to route the error, the *weight-transport problem* [4], which has no natural physical realisation. Local methods avoid both: the error and presynaptic activity needed to update a synapse are available in situ, so the update can, in principle, happen where the weight is stored.

In-memory and analogue computing.

This locality is precisely what in-memory and analogue accelerators exploit. In a memristor or resistive crossbar, the synaptic weight *is* the conductance of a device, so a matrix–vector product is performed in place by Ohm's and Kirchhoff's laws; computation and memory are co-located and weight movement is eliminated [59]. For deep learning, this means lower data-transfer cost, higher compute efficiency per joule, and training and inference on the same device [60]. The feasibility of *training* in such memory is no longer hypothetical: analogue non-volatile memory has trained networks to software-equivalent accuracy on MNIST and CIFAR by performing the multiply-accumulate in place at the location of the weights [61], exactly the regime in which a local update rule is most at home (Figure 19). A crossbar can compute both $\boldsymbol{W}\boldsymbol{x}$ and the transpose product $\boldsymbol{W}^{\top}\boldsymbol{e}$ that the relaxation dynamics require by driving current along rows or columns of the *same* physical device, which avoids storing and moving a separately materialised transposed copy of

the weights in memory. This is a genuine hardware-level mitigation specific to resistive crossbars, not a general resolution of the weight-symmetry requirement: it does not extend to biological synapses, and even on this substrate it still relies on the forward and relaxation passes reading the same physical device symmetrically, which is the open weight-symmetry problem flagged in Section 7.3.



Figure 19. Why locality matters for energy. (**a**) On von Neumann hardware, weights and activations are shuttled between memory and compute, and this data movement, not the arithmetic, dominates the energy budget [6]; backpropagation worsens it by storing the activation graph and transporting transposed weights. (**b**) In an analogue crossbar, the weight is the device, so the multiply-accumulate happens in place, and local rules update it without moving data, the regime where energy-based local learning is projected to gain orders of magnitude [12,62].

In situ local training on crossbars.

The point that matters for this survey is that crossbars favour *local* learning rules. Self-adaptive in situ learning has been demonstrated in multilayer memristor networks [63]; parallel in situ training has been realised with electrochemical RAM [64]; and *activity-difference* training, updating weights from the difference between two network states, a contrastive/equilibrium-style local rule that avoids an explicit backward pass, has been run directly on memristor crossbars [62]. These results make energy-based, local credit assignment a natural algorithmic match for analogue hardware. The principal obstacles are device-level: nonlinear and stochastic conductance updates and device-to-device variability degrade training accuracy, and overcoming them is an active hardware–algorithm co-design problem [65].

Neuromorphic processors and on-chip learning.

Digital neuromorphic processors provide event-driven, massively parallel substrates with on-chip plasticity: IBM TrueNorth runs one million spiking neurons below 100 mW [66]; Intel Loihi pioneered on-chip local learning [67]; its successor Loihi 2 and SpiNNaker2 are current event-driven platforms in which states and weights are held in local memory beside the compute [66,68]. Local rules are a natural fit, and EqSpike is the clearest algorithmic example: it recasts equilibrium propagation in a spike-driven form and, when mapped to silicon, is projected to cut inference energy by about three orders of magnitude and training by about two orders of magnitude relative to a GPU, while supporting online on-chip learning [12]. A caveat: on-chip learning can itself dominate the energy budget, so the update rule must be hardware-aware [65].

Spiking implementations.

Bridging continuous local-learning theory and spiking hardware is an active area. Surrogate-gradient methods make spiking networks trainable by smoothing the non-differentiable spike [69], and analogue surrogate-gradient training has been demonstrated on neuromorphic substrates [64]. The conversion of rate-based relaxation into event-driven dynamics and the device variability such mappings must tolerate remain the central trade-offs [65].

7.2. Representative Application Contexts

The benchmark focused on continual learning, but local equilibrium networks are also relevant where on-device adaptation, limited memory, or biological plausibility constrain the design.

Continual and lifelong learning.

This is the application tested here. Sequential learners must update after deployment without retaining every past example, and energy-based local networks offer a natural raw-data-free replay mechanism because they can generate samples from their own learned structure [13,70,71]. The benchmark in Section 6 should be read as a minimal reproducible case of that idea, not as a final solution for all lifelong-learning settings.

Perception and representation learning.

Predictive coding began as a model of cortical perception and remains useful as a generative representation learner: a hierarchy that minimises prediction error can also learn features and reconstruct missing or corrupted inputs [7,11,72,73]. For the present paper, this matters because replay quality depends on whether the generator captures the data manifold; the CIFAR-10 limitation is therefore partly a representation limitation.

Edge, robotics and neuromorphic deployment.

Local updates are attractive when devices face strict memory, bandwidth and energy budgets, as in embedded perception, adaptive robotics and streaming sensorimotor control [12,64,66–68]. Spike-based neuromorphic processors are the substrate most often proposed for this regime [74], and local, hardware-compatible learning rules for spiking recurrent networks now exist for both feedforward credit assignment [75] and networks with internal recurrent dynamics [76], though whether neuromorphic hardware nets out as an energy win once the full system, not just the arithmetic, is counted remains an open, actively scrutinised question [77]. The claim is conditional: on ordinary CPUs the iterative relaxation is slower than backpropagation, but on in-memory or neuromorphic substrates locality may reduce data movement. That hardware advantage remains to be measured for the full continual pipeline.

7.3. Open Challenges and Research Programme

The evidence assembled in this survey, namely coherent theory, a reproducible case study and a hardware case that is strong in principle, also marks where the field needs to go next. Three barriers must be overcome before local continual learning can move beyond image benchmarks, and the community is actively working on all three.

Depth and scaling.

The depth/error-imbalance barrier is the central open problem of local learning. Backpropagation itself only conquered very deep networks through residual connections [78] and normalisation [79]; local learning still lacks a comparably mature toolkit, and importing these ideas is part of the active effort below. As our testbed shows (Section 6, Figure 18), training a deep energy-based network from scratch leaves the layers near the input with vanishingly small equilibrium errors, so they barely learn; vanilla predictive coding is known to degrade once the stack grows beyond a handful of layers. Several complementary remedies are under active investigation, which we group by the mechanism they target.

Parameterisation, re-balancing and better schedules.

Depth-aware parameterisation, adapting μP-style scaling to inference dynamics, keeps activity learning rates stable across width and depth. Precision-weighting of per-layer errors redistributes the energy during relaxation, the natural remedy for the imbalance

our testbed exhibits (Figure 18). Prospective-configuration-style training separates inference from plasticity, making the learning rule easier to analyse and improving sample efficiency [10,24].

EqProp at scale and shared benchmarks.

Reducing the bias of the finite-nudge gradient estimator enabled scaling to deep convolutional networks [28]; standardised benchmarks are beginning to make cross-method comparison fair [24]. The problem is actively contested but not closed, and our reproducible testbed (Figure 18) is a small, transparent instrument for studying it.

Nudge, weight symmetry and the two-phase requirement.

Energy-based methods classically require either an infinitesimal nudge (EqProp) or clamped outputs (PC), and often symmetric weights, requirements that are awkward biologically and on hardware. Recent work relaxes the infinitesimal-nudge assumption with finite-nudge formulations, while feedback-alignment results show that exact weight symmetry is not strictly necessary for useful learning [4]. Reducing or eliminating the two-phase structure and quantifying the bias introduced by finite nudges remain open.

Extending the benchmark beyond energy-based methods.

The diagnostic benchmark in Sections 5 and 6 tests only the energy-based (LEL) family, because its native generative mode is what supplies rehearsal data without stored exemplars (Section 2.5). Forward-only learning and feedback alignment have no equivalent built-in generator, so testing them on the same continual-learning diagnostic requires pairing them with an external generative model or a stored-exemplar replay baseline and re-measuring the same class-, task- and domain-incremental protocol (Section 5.3). This would separate what the continual-learning result owes to *locality in general* from what it owes specifically to the energy-based family's generative capacity, closing the scope limitation acknowledged in Section 2.5.

Modern architectures and foundation models.

Almost all local-learning results concern multilayer perceptrons and convolutional networks; extending local credit assignment to the architectures that dominate contemporary AI, attention and transformers [5], is largely unexplored. Sequential tasks remain dominated by LSTM networks and their variants [80], although reservoir-computing approaches such as echo state networks, which train only a linear readout while leaving the recurrent reservoir weights fixed, already avoid backpropagating through the recurrence entirely [81]; a local formulation of attention-based sequence processing has not yet demonstrated comparable accuracy on standard benchmarks. Whether an energy-based or forward-only formulation of self-attention can match backpropagation is, to our knowledge, an open and high-value question. Two adjacent lines of work show that backpropagation-free ideas are already reaching transformer-scale models, although neither is local in the neuron-level sense used throughout this survey. First, at the *architecture* level, the Energy Transformer replaces the attention block with a Hopfield-style energy minimisation, so that inference becomes gradient descent on an explicit energy function [82], showing that the energy-based view of Section 4 is not confined to small multilayer networks. Second, at the *optimisation* level, zeroth-order, forward-pass-only fine-tuning (MeZO) removes the backward pass and its activation memory entirely by estimating the gradient from paired forward evaluations, at the cost of more forward passes and, for very large models, slower convergence than backpropagation [83]; this is backpropagation-free in the sense of requiring no backwards-mode automatic differentiation, but it still optimises a single global scalar loss rather than a spatially local energy, so it is best read as a complementary, coarser-grained route to the same structural goal, avoiding stored activation graphs, rather

than an instance of the local family taxonomised in Section 2. On the continual-learning side, catastrophic forgetting during LLM fine-tuning is an active and separate research thread: prior work treats pretrained transformers as continual learners across task sequences [84] and mitigates forgetting with model-generated rehearsal data [85], an LLM-scale analogue of the generative "dreaming" mechanism studied in Section 6.3 of this survey, but built on backpropagation and standard fine-tuning rather than a local energy-based learner. To our knowledge, no existing study combines a local, energy-based credit-assignment rule with a transformer-scale foundation model under a continual-learning protocol; closing that gap is a natural, high-value extension of the benchmark presented here, and we flag it explicitly as outside the scope of the present diagnostic study, which uses small MLPs and CNNs by design (Section 5.3).

Theory and evaluation.

Theoretical understanding is advancing (Lyapunov stability of the inference dynamics [22], the infinitesimal-inference limit that links these methods to backpropagation [8,9]), but generalisation guarantees, convergence rates under realistic conditions, and the interaction between inference accuracy and learning remain partially open. On the empirical side, the field has historically suffered from inconsistent benchmarks; unified evaluation suites and reproducible studies such as this one are needed for cumulative progress [24].

Computational and energy trade-offs.

On von Neumann hardware LEL is *more* expensive: the T-step relaxation costs $\sim 5\times$ the operations of backpropagation and was $\sim 7.5\times$ slower in our CPU measurement; we claim no speed advantage there. Its structural benefits are (i) activation memory $O(\text{width})$ vs. $O(\text{depth} \times \text{width})$, and (ii) locality, which permits execution on in-memory and neuromorphic substrates that backpropagation cannot easily use because of the weight-transport problem [4].

The structural reason is data movement: off-chip DRAM access costs about two orders of magnitude more energy than the arithmetic it feeds [6], so on von Neumann hardware training energy is dominated by shuttling weights and activations rather than by multiply-accumulate operations. Because every LEL update reads only quantities already present at the synapse, it can execute where the weight is stored, eliminating the dominant data-movement term and projecting a 120–1200× reduction relative to backpropagation on DRAM, consistent with figures reported for spike-driven equilibrium and surrogate-gradient learning on neuromorphic substrates [12,64,69]. Backpropagation cannot occupy this regime at all because its backward pass requires a transported copy of the weights. We stress that this is a projection from per-operation figures, not a device measurement; the numbers in Table 18 quantify the on-hardware cost.

Empirical failures and scope limits.

The negative results are first-class findings, not caveats appended after the fact. They separate what failed empirically from what remains outside the scope of this benchmark.

Negative empirical results.

Vanilla local learning does not beat backpropagation in accuracy or forgetting; under the fully clamped protocol used in the benchmark the equilibrium update descends the free energy rather than the exact backpropagation gradient (the two coincide only in the small-nudge limit of Theorem 2, Section 4), and it empirically inherits catastrophic forgetting unless an additional retention mechanism is added. Weight regularisation also failed in the class-incremental setting: EWC and SI stayed near chance, and the energy-based EWC variant destabilised relaxation by pulling weights away from the contractive regime required by Proposition 1. Noise replay failed for a different reason: random samples lie off

the data manifold and therefore do not recreate meaningful old-task decision boundaries. Finally, simple "free" fixes for depth, including metaplastic per-synapse learning rates and homeostatic norm control, did not remove the error-imbalance barrier.

Table 18. Efficiency of a $784 \to 256 \to 10$ network (per training sample). On von Neumann hardware LEL costs more; the energy advantage is a projection on in-memory substrates using the figures of [6]. † See qualifications below the table; both the memory and the in-memory energy figures require substantial caveats that we now state explicitly.

Metric	Backprop	LEL (This Study)
Relative compute (vs. forward)	∼3×	∼16× ($T = 15$)
Wall-clock (CPU, same epochs)	1×	∼7.5× slower
Activation/state memory †	O(depth × width)	O(depth × width)
Energy, von Neumann (DRAM)	∼390 µJ	∼2100 µJ (worse)
Energy, in-memory (projected) †	not applicable	∼0.3–3 µJ (partial, see below)

† **Memory.** Algorithm 1 retains the states $\boldsymbol{x}_\ell$ of *all* L layers simultaneously throughout the T-step relaxation, so total state memory is O(depth × width), the same asymptotic order as backpropagation's stored forward activations; we correct an earlier overstatement of this row as O(width). The O(width) quantity that is genuinely local is the memory needed to *compute one layer's weight update* (Lemma 1: only $\boldsymbol{e}_\ell$ and $\boldsymbol{x}_{\ell-1}$), not the network's total simultaneous state. The real structural difference from backpropagation is qualitative, not a reduction in peak activation memory: LEL does not need the additional bookkeeping of a reverse-mode automatic-differentiation graph kept alive from the forward pass until a backward sweep consumes it, and the weight update itself needs no backward sweep once the equilibrium states are available. **Energy.** The in-memory projection uses only the per-operation energy figures of [6] for arithmetic; it does *not* yet include state movement between memory and compute during the T-step relaxation, analogue–digital conversion overhead, write energy for the per-class generators, or the cost of generating and replaying samples during continual learning, all of which are required for a fair, same-substrate comparison and are not measured here. We report it as a directional, order-of-magnitude projection and flag a full-pipeline, same-substrate energy measurement, including relaxation, conversion, generator training and replay, as open work in Section 7.3.

Depth limitation.

The strongest algorithmic limitation is depth. In the current implementation, layers near the input receive equilibrium errors several orders of magnitude smaller than layers near the output, so adding depth can reduce rather than improve accuracy (Figure 18 and Table 17). Grafting helps by allowing early layers to learn while they are still close to the output, but it does not solve the underlying imbalance. The manuscript therefore does not claim that LEL is ready for very deep networks; it identifies a reproducible failure mode that future local-learning methods must overcome.

Computational-cost limitation.

LEL is not faster on standard von Neumann hardware. The iterative relaxation requires multiple state updates per training sample, making it roughly 5–7.5× more expensive in our measurements than a matched backpropagation baseline (Table 18). The possible advantage is structural rather than immediate: local updates reduce activation storage and may avoid costly data movement when executed in memory or on neuromorphic hardware. Until that full pipeline is implemented and measured on a device, the energy argument remains a projection.

CIFAR-10 limitation.

The CIFAR-10 result is deliberately modest. A final accuracy of 37.0% would be unacceptable if the paper were claiming competitiveness with modern backpropagation-trained CNNs or transformers. That is not the claim. The experiment tests whether the same local replay mechanism still recovers most of the available ceiling when the base model is a shallow raw-pixel classifier. It does (37.0% out of a 39.6% ceiling), but the low ceiling shows that useful deployment will require stronger local representations, feature-space replay or a more capable local generator. We state the generalisation claim narrowly, on purpose: the

experiment shows that local replay preserves most of what *this specific* shallow, raw-pixel classifier can achieve on CIFAR-10 under *this* protocol; it does not show, and should not be read as evidence, that the same mechanism scales to a stronger visual-recognition setting, a deeper network, or learned (rather than raw-pixel) features, each of which would need to be tested on its own terms before the claim could be extended.

Scope limitation.

The benchmark uses small networks, public image datasets and controlled task sequences. It does not address large-scale foundation models, naturalistic non-stationarity, continual reinforcement learning or measured hardware energy. These omissions are intentional: the purpose is to provide a clean reference point where the role of local credit assignment, replay, generator quality and memory budget can be inspected without hidden pretrained components.

Research programme.

Table 19 translates the open challenges above into testable tasks, tracing a path from the current proof of concept (local generative replay on image benchmarks) toward scalable, hardware-native continual learning without backpropagation.

Table 19. Open questions organised as testable research tasks. The table is intended as a starting point for new projects rather than as a complete forecast of the field.

Problem	Near-Term Test	Evidence of Progress
Depth and scaling	Train deeper LEL/PC networks while monitoring per-layer equilibrium errors	Error magnitudes remain balanced and accuracy improves with added depth
Generator quality	Swap local generators under a fixed replay protocol	Class-IL accuracy approaches the oracle without storing raw data
Feature-space replay	Run local continual learning on learned or frozen representations	Higher CIFAR-like ceilings without changing the continual mechanism
Attention	Formulate local updates for attention-like retrieval	Sequence or transformer-style benchmark trained without a global backward pass
Hardware	Implement relaxation and dreaming on neuromorphic or in-memory hardware	Measured joule advantage, not only projected operation counts
Memory accounting	Report raw data, feature data, generator and statistic budgets separately	Fair comparison between replay, regularisation and architecture methods
Evaluation culture	Publish failures and controls alongside positive numbers	Fewer irreproducible claims and clearer design boundaries

8. Conclusions

This survey set out to answer one concrete question: can a backpropagation-free network keep learning without forgetting? The answer is yes, but only with the right retention mechanism, and that finding requires common, reproducible ground rather than the scattered formulations, notations and benchmarks that have fragmented the local-learning literature. We organised the two halves of the problem (backpropagation-free credit assignment and continual learning) into one taxonomy, expressed the energy-based family through a single transparent formulation, re-derived and numerically re-validated its core theory, and ran a controlled empirical case study across three datasets and two independent implementations. The intent was never a new learning rule, but a synthesis,

reproducibility package and benchmark that make the field legible and give others a concrete baseline to extend or falsify.

Empirical synthesis.

Across three datasets and two independent implementations, the results converge on a clear hierarchy: generative dreaming is the only retention mechanism that works reliably under the strict Class-IL protocol. EWC and SI remain near chance; noise replay fails because random samples lie off the data manifold; only locally generated "dream" samples, drawn from the energy-based model's own learned distribution, recreate the decision boundaries needed to prevent catastrophic forgetting. The effect is quantitatively large and general: local dreaming reaches 86–88% retention on Split-MNIST (96% of the joint-training ceiling) and 96–98% of the ceiling on Fashion-MNIST and CIFAR-10, reproduced in a second independent implementation. The remaining gap to the oracle ceiling is governed by generator quality, not by the continual mechanism, a conclusion the generator-capacity ablation confirms directly (Figure 15).

What the evidence shows.

Four findings hold up across every dataset and both implementations. First, local energy-based learning *matches* backpropagation in supervised accuracy under the fully clamped, finite-T protocol used throughout the benchmark. This parity is empirical, not a direct instance of Theorem 2: that theorem proves exact gradient equivalence only in the small-nudge limit $\beta \to 0$ (Figure 8), a regime distinct from the full clamping used here, under which the local update descends the free energy E and reaches only partial cosine alignment ($\approx$0.62) with the backpropagation gradient; we confirm both regimes numerically rather than conflating them. Second, given that we have established that fully clamped local learning matches backpropagation's accuracy without any built-in protection for past tasks, it is unsurprising, and we confirm empirically, that vanilla local learning also *inherits* backpropagation's catastrophic forgetting; locality is not, by itself, a cure. Third, the cure is structural rather than regularisational: weight anchoring (EWC, SI) fails in the class-incremental regime, and when ported to the energy-based setting it even destabilises the inference relaxation. *Generative replay*, by contrast, succeeds, and energy-based networks support it natively because the same network that discriminates can dream on its own learned data manifold. Fourth, the resulting effect is large and general: local dreaming converts chance-level forgetting into 86–88% retention on Split-MNIST (96% of the joint-training ceiling) and reaches 96–98% of the ceiling on Fashion-MNIST and CIFAR-10, reproduced in a second independent implementation. The limit is as clear as the result: the absolute number is set by the quality of the base classifier and the generator, not by the continual mechanism, and we isolated this experimentally rather than just asserted it, showing retention rising monotonically with generator capacity toward the oracle (Figure 15).

Why it matters.

These results recast backpropagation as one credit-assignment choice among several rather than a law of learning. The case for the alternatives is not, today, that they are more accurate; they are not, a conclusion consistent with broader reviews of continual and biologically inspired learning [37–39]. It is that they are *structurally different* in ways that matter for where computing is heading: they are local, so they need neither a stored computation graph nor a transported copy of the weights; they are generative, so they can rehearse the past without storing it; and they update online, which is the native regime of systems that must keep learning after deployment. These are exactly the properties an energy-, memory- and data-movement-constrained future rewards, and they are the reason

the intersection of local learning and continual learning, the niche this survey occupies, is worth taking seriously.

What remains open.

We have been deliberate about not overclaiming. On conventional von Neumann hardware, the approach is several times slower and more energy-hungry than backpropagation; its energy advantage is a principled *projection* onto in-memory and neuromorphic substrates, not a measurement we made. Training degrades in very deep networks because the layerwise equilibrium errors become exponentially imbalanced. We reproduce this open problem honestly, and the community is actively attacking it through reparameterisation, energy re-balancing and better inference schedules (Section 7). And the extension of local credit assignment to the architectures that dominate contemporary AI, namely attention and transformers, remains almost entirely unexplored. We also reported, as carefully as the positive results, what did *not* work: naive energy sampling diverges, regularisation destabilises the relaxation, and the most natural "free" fixes for the depth barrier did not help. These negative results are part of the contribution, because they delimit the design space for those who follow.

Outlook.

The most important next step is to stop projecting and start *measuring*: to express the relaxation and the generative dreaming in a spiking or analogue form and report joules on real neuromorphic devices, converting the structural energy argument into evidence. In parallel, importing the depth remedies surveyed here into the continual-learning pipeline and probing whether an energy-based or forward-only formulation of self-attention can match backpropagation would tell us whether local learning scales to the problems that matter most; hybrid architectures that already pair a fixed-dynamics reservoir with a trainable attention module for structured, interpretable classification illustrate one concrete direction for such hybridisation [86]. We have tried to lower the barrier to all of this by releasing the unified formulation and every experiment as small, dependency-light, independently reproducible scripts. The primary contribution of this review is to establish a reproducible baseline and clarify the research agenda for local credit assignment in continual learning, not to resolve it.

Funding: This research received no external funding.

Data Availability Statement: All code and scripts used in this benchmark are publicly available at https://github.com/chvn00/LEL (accessed on 30 June 2026).

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LEL	Local Equilibrium Learning.
BP	Backpropagation.
PC	Predictive Coding.
EqProp	Equilibrium Propagation.
FA	Feedback Alignment.
TP	Target Propagation.
EWC	Elastic Weight Consolidation.
SI	Synaptic Intelligence.
CL	Continual Learning.
Class-IL	Class-Incremental Learning.

Task-IL	Task-Incremental Learning.
BIR	Brain-Inspired Replay.
MLP	Multilayer Perceptron.
STDP	Spike-Timing-Dependent Plasticity.
GPU	Graphics Processing Unit.

Appendix A. Terminology

Table A1 fixes how terms are used in this paper. These terms are included because the same words are sometimes used differently across the backpropagation-free and continual-learning literatures.

Table A1. Terminology used throughout the paper.

Term	Meaning in This Paper
Backpropagation-free	The credit-assignment update does not require a global backward pass through the computational graph. The term does not imply better accuracy, lower cost on CPUs/GPUs, or continual learning by itself.
Local update	A synaptic update uses quantities available at the connected units, typically presynaptic activity and postsynaptic prediction error or state.
Energy-based learning	Inference is cast as relaxation of neural states to reduce an energy or free-energy objective; learning updates parameters using the equilibrium states.
LEL	The particular local equilibrium formulation used in this paper: a transparent predictive-coding-style network with squared prediction errors and local Hebbian updates.
Dreaming	Storage-free generative replay: pseudo-samples of old classes are generated by a local generator and mixed with current-task data.
Oracle replay	Replay using stored real examples from past tasks. It is a bound, not a storage-free method.
Class-incremental learning	The model must classify among all classes seen so far without being told the task identity at test time.
Task-incremental learning	The model is told the task identity or evaluated within the correct task head at test time.
Generator bottleneck	The situation in which replay would work with perfect samples, but the learned generator does not cover the old data manifold well enough.
Depth/error imbalance	The decay of equilibrium errors across layers in deep local networks, which leaves early layers with too little learning signal.

Appendix B. Implementation Notes for LEL

Appendix B records several implementation details that are easy to miss when translating the equations into code. They are not new theory, but they are important for reproducing the behaviour reported in the paper.

Appendix B.1. Signs and Objectives

In the supervised, fully clamped version used for most experiments, the output state is fixed to the target during relaxation. The layer error is $\boldsymbol{e}_\ell = \boldsymbol{x}_\ell - f(\boldsymbol{W}_\ell \boldsymbol{x}_{\ell-1} + \boldsymbol{b}_\ell)$, and the local weight increment is proportional to $(f'(\boldsymbol{a}_\ell) \odot \boldsymbol{e}_\ell)\boldsymbol{x}_{\ell-1}^\top$. This sign is chosen so that the update descends the reduced free energy. It is tempting to compare this update directly with the backpropagation gradient of the supervised loss, but in the fully clamped regime the objectives are not identical. The exact comparison to backpropagation is in the nudged regime of Theorem 2, where the output is softly pulled toward the target, and the limit

$\beta \to 0$ is taken. This distinction explains why the theorem-validation script reports two different behaviours: full clamping saturates below perfect cosine alignment, while the small-nudge update approaches an alignment of one.

Appendix B.2. Relaxation Stability

The inference step size γ is a stability parameter as much as a speed one. If γ is too small, the network may not approach equilibrium within the chosen budget T; if it is too large, the activities oscillate or diverge. A simple diagnostic is to track the free energy or the norm of the state gradient across relaxation steps. In a stable run, the energy should decrease or at least settle after a small transient caused by finite-step discretisation. When the energy grows systematically, increasing epochs or replay samples will not fix the experiment; the state dynamics themselves are outside the contractive regime assumed by Proposition 1.

Appendix B.3. Finite Inference and Biased Updates

The envelope theorem gives the local gradient at equilibrium. In code, equilibrium is approximated by a finite number of steps. The resulting update is biased, and the bias is controlled by both the relaxation budget T and the conditioning of the energy landscape. This is why T should be reported even when accuracy appears insensitive to it. A shallow network may work with $T = 1$ or $T = 2$ because the forward initialisation is already close to equilibrium; a deeper network may require more steps, different precisions or residual dynamics before the same claim is meaningful.

Appendix B.4. Clamping, Nudging and Test-Time Inference

Training and testing use different operations. During training, the input and either the output target (clamped case) or a soft output penalty (nudged case) participate in the energy. During testing, the experiments here use the feedforward prediction for classification, because no target is available. One can also run free relaxation at test time, but then the protocol must specify which variables are clamped, which are free, and how the predicted label is read out. Mixing these conventions across methods can change the result, especially in recurrent or strongly energy-based models.

Appendix B.5. Generator Training and Labelling

The dreaming pipeline contains two models: a generator for samples and a source of labels for them. In the idealised, fully sequential design (Algorithm 2), each generator is trained only on the current class data when that class is first seen and is later sampled without access to old raw data, and dreamed samples are labelled by a frozen copy of the classifier, the general pseudo-rehearsal setting. We state plainly that the released scripts producing the headline numbers (Remark 2) do not implement this design: generators (and the global centring statistic) are pre-computed once over each class's full data before the task loop, and dreamed samples are labelled directly with the identity of the generator that produced them rather than by querying a frozen classifier. This is a controlled simplification, not the general pseudo-rehearsal setting, and it matters because self-labelling by a frozen classifier propagates that classifier's own errors into the rehearsal set, an effect the released experiments do not exercise. We tested whether this expectation holds by implementing the fully sequential, self-labelled variant directly (`strict_sequential_dreaming.py`): over two seeds it reaches 86.8%/97.4% (LEL) and 90.0%/98.1% (BP) class-/task-incremental accuracy, statistically indistinguishable from the 86.7%/97.4% and 90.3%/98.1% obtained under the simplified protocol. At this benchmark's scale, the frozen classifier's labelling errors are evidently too rare, or too easily absorbed by the generative replay mechanism,

to measurably harm retention; whether this holds at larger scale or with a weaker base classifier remains open.

Appendix B.6. Minimal Pseudocode for One Task Transition

The following task-transition view is often easier to implement than the full algorithmic listing. Let θ_{k-1} be the classifier after task $k-1$, and let $\{G_c : c \in \mathcal{S}_{k-1}\}$ be the stored local generators for old classes. For a new task with data D_k, create a training set

$$\widetilde{D}_k = D_k \cup \{(G_c(z_i),\ \theta_{k-1}(G_c(z_i))) : c \in \mathcal{S}_{k-1},\ i = 1, \ldots, n_d\}. \tag{A1}$$

Then train the classifier locally on $\widetilde{D}_k$, train new generators for the new classes, and freeze a copy of the classifier for the next labelling step. The important feature is that past raw data never enter $\widetilde{D}_k$; only generated samples do.

References

1. Baydin, A.G.; Pearlmutter, B.A.; Radul, A.A.; Siskind, J.M. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.* **2018**, *18*, 1–43.
2. Hooker, S. The hardware lottery. *Commun. ACM* **2021**, *64*, 58–65. [CrossRef]
3. Lillicrap, T.P.; Santoro, A.; Marris, L.; Akerman, C.J.; Hinton, G. Backpropagation and the brain. *Nat. Rev. Neurosci.* **2020**, *21*, 335–346. [CrossRef]
4. Lillicrap, T.P.; Cownden, D.; Tweed, D.B.; Akerman, C.J. Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.* **2016**, *7*, 13276. [CrossRef]
5. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
6. Sze, V.; Chen, Y.-H.; Yang, T.-J.; Emer, J.S. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* **2017**, *105*, 2295–2329. [CrossRef]
7. Millidge, B.; Tschantz, A.; Buckley, C.L. Predictive coding approximates backprop along arbitrary computation graphs. *Neural Comput.* **2022**, *34*, 1280–1317. [CrossRef]
8. Whittington, J.C.R.; Bogacz, R. An approximation of the error backpropagation algorithm in a predictive coding network with local Hebbian synaptic plasticity. *Neural Comput.* **2017**, *29*, 1229–1262. [CrossRef]
9. Scellier, B.; Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Front. Comput. Neurosci.* **2017**, *11*, 24. [CrossRef]
10. Song, Y.; Millidge, B.; Salvatori, T.; Lukasiewicz, T.; Xu, Z.; Bogacz, R. Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nat. Neurosci.* **2024**, *27*, 348–358. [CrossRef]
11. Bogacz, R. A tutorial on the free-energy framework for modelling perception and learning. *J. Math. Psychol.* **2017**, *76*, 198–211. [CrossRef]
12. Martin, E.; Ernoult, M.; Laydevant, J.; Li, S.; Querlioz, D.; Petrisor, T.; Grollier, J. EqSpike: Spike-driven equilibrium propagation for neuromorphic implementations. *iScience* **2021**, *24*, 102222. [CrossRef]
13. van de Ven, G.M.; Tuytelaars, T.; Tolias, A.S. Three types of incremental learning. *Nat. Mach. Intell.* **2022**, *4*, 1185–1197. [CrossRef]
14. Kumaran, D.; Hassabis, D.; McClelland, J.L. What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends Cogn. Sci.* **2016**, *20*, 512–534. [CrossRef]
15. Annabi, L.; Pitti, A.; Quoy, M. Continual sequence modeling with predictive coding. *Front. Neurorobot.* **2022**, *16*, 845955. [CrossRef]
16. Nechesov, A.; Ruponen, J. CALM: Continual associative learning model via sparse distributed memory. *Technologies* **2025**, *13*, 587. [CrossRef]
17. Papachristodoulou, A.; Kyrkou, C.; Timotheou, S.; Theocharides, T. Convolutional channel-wise competitive learning for the forward-forward algorithm. In Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI-24), Vancouver, BC, Canada, 20–27 February 2024.
18. Scodellaro, R.; Kulkarni, A.; Alves, F.; Schröter, M. Training convolutional neural networks with the forward-forward algorithm. *Sci. Rep.* **2025**, *15*, 38461. [CrossRef]
19. Kohan, A.; Rietman, E.A.; Siegelmann, H.T. Signal propagation: A framework for learning in a forward pass. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 8585–8596. [CrossRef]
20. Nøkland, A. Direct feedback alignment provides learning in deep neural networks. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.

21. Meulemans, A.; Carzaniga, F.S.; Suykens, J.A.K.; Sacramento, J.; Grewe, B.F. A theoretical framework for target propagation. In Proceedings of the 34th Conference on Neural Information Processing Systems (NIPS 2020), Vancouver, BC, Canada, 6–12 December 2020; Volume 33, pp. 20024–20036.
22. Ramsauer, H.; Schäfl, B.; Lehner, J.; Seidl, P.; Widrich, M.; Adler, T.; Gruber, L.; Holzleitner, M.; Pavlović, M.; Sandve, G.K.; et al. Hopfield networks is all you need. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
23. Krotov, D.; Hopfield, J.J. Dense associative memory for pattern recognition. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
24. Chen, H.; Yang, B.; He, F.; Zhou, F.; Chen, S.; Wu, C.; Li, F.; Chua, Y. Effective methods and framework for energy-based local learning of deep neural networks. *Front. Artif. Intell.* **2025**, *8*, 1605706. [CrossRef]
25. Salvatori, T.; Pinchetti, L.; Millidge, B.; Song, Y.; Bao, T.; Bogacz, R.; Lukasiewicz, T. Learning on arbitrary graph topologies via predictive coding. In Proceedings of the 36th Conference on Neural Information Processing Systems (NIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022.
26. Tang, M.; Salvatori, T.; Millidge, B.; Song, Y.; Lukasiewicz, T.; Bogacz, R. Recurrent predictive coding models for associative memory employing covariance learning. *PLoS Comput. Biol.* **2023**, *19*, e1010719. [CrossRef]
27. Millidge, B.; Salvatori, T.; Song, Y.; Lukasiewicz, T.; Bogacz, R. Predictive coding: Towards a future of deep learning beyond backpropagation? In Proceedings of the International Joint Conferences on Artificial Intelligence, Vienna, Austria, 22–29 July 2022.
28. Laborieux, A.; Ernoult, M.; Scellier, B.; Bengio, Y.; Grollier, J.; Querlioz, D. Scaling equilibrium propagation to deep ConvNets by drastically reducing its gradient estimator bias. *Front. Neurosci.* **2021**, *15*, 633674. [CrossRef]
29. Scellier, B.; Ernoult, M.; Kendall, J.; Kumar, S. Energy-based learning algorithms for analog computing: A comparative study. In Proceedings of the 37th Conference on Neural Information Processing Systems (NIPS 2023), New Orleans, LA, USA, 10–16 December 2023.
30. Laborieux, A.; Zenke, F. Holomorphic equilibrium propagation computes exact gradients through finite size oscillations. In In Proceedings of the 36th Conference on Neural Information Processing Systems (NIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022.
31. O'Shea, R.; Rajendran, B. Closed-form feedback-free learning with forward projection. *Nat. Commun.* **2026**, *17*, 2414. [CrossRef]
32. Nøkland, A.; Eidnes, L.H. Training neural networks with local error signals. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
33. Sebastian, A.; Le Gallo, M.; Khaddam-Aljameh, R.; Eleftheriou, E. Memory devices and applications for in-memory computing. *Nat. Nanotechnol.* **2020**, *15*, 529–544. [CrossRef]
34. Burr, G.W.; Shelby, R.M.; Sebastian, A.; Kim, S.; Kim, S.; Sidler, S.; Virwani, K.; Ishii, M.; Narayanan, P.; Fumarola, A.; et al. Neuromorphic computing using non-volatile memory. *Adv. Phys. X* **2017**, *2*, 89–124. [CrossRef]
35. Guerguiev, J.; Lillicrap, T.P.; Richards, B.A. Towards deep learning with segregated dendrites. *eLife* **2017**, *6*, e22901. [CrossRef]
36. Marblestone, A.H.; Wayne, G.; Kording, K.P. Toward an integration of deep learning and neuroscience. *Front. Comput. Neurosci.* **2016**, *10*, 94. [CrossRef]
37. De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3366–3385. [CrossRef]
38. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Netw.* **2019**, *113*, 54–71. [CrossRef]
39. Hadsell, R.; Rao, D.; Rusu, A.A.; Pascanu, R. Embracing change: Continual learning in deep neural networks. *Trends Cogn. Sci.* **2020**, *24*, 1028–1040. [CrossRef]
40. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef]
41. Zenke, F.; Poole, B.; Ganguli, S. Continual learning through synaptic intelligence. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
42. Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
43. Li, Z.; Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2935–2947. [CrossRef]
44. Shin, H.; Lee, J.K.; Kim, J.; Kim, J. Continual learning with deep generative replay. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
45. van de Ven, G.M.; Siegelmann, H.T.; Tolias, A.S. Brain-inspired replay for continual learning with artificial neural networks. *Nat. Commun.* **2020**, *11*, 4069. [CrossRef]

46. Graffieti, G.; Maltoni, D.; Pellegrini, L.; Lomonaco, V. Generative negative replay for continual learning. *Neural Netw.* **2023**, *162*, 369–383. [CrossRef]
47. Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. iCaRL: Incremental classifier and representation learning. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
48. Lopez-Paz, D.; Ranzato, M. Gradient episodic memory for continual learning. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 6467–6476.
49. Chaudhry, A.; Ranzato, M.; Rohrbach, M.; Elhoseiny, M. Efficient lifelong learning with A-GEM. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
50. Masse, N.Y.; Grant, G.D.; Freedman, D.J. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, E10467–E10475. [CrossRef]
51. Mallya, A.; Lazebnik, S. PackNet: Adding multiple tasks to a single network by iterative pruning. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7765–7773.
52. Benna, M.K.; Fusi, S. Computational principles of synaptic memory consolidation. *Nat. Neurosci.* **2016**, *19*, 1697–1706. [CrossRef]
53. Zenke, F.; Neftci, E.O. Brain-inspired learning on neuromorphic substrates. *Proc. IEEE* **2021**, *109*, 935–950. [CrossRef]
54. Aslam, M.A.; Hamza, M.; Zhu, S.; Hu, H.; Xu, W.; Irfan, M.; Zheng, J.; Aslam, S.; Khosravi, M. Continual learning inspired by brain functionality: A comprehensive survey. *Int. J. Intell. Syst.* **2025**, *2025*, 3145236. [CrossRef]
55. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
56. Pehlevan, C.; Mohan, S.; Chklovskii, D.B. Blind nonnegative source separation using biological neural networks. *Neural Comput.* **2017**, *29*, 2925–2954. [CrossRef]
57. Rodríguez-Torres, A.; Valencia-Niño, C.H.; Alvarez-Icaza, L. Application of classical and quantum-inspired methods through multi-objective optimization for parameter identification of a multi-story prototype building. *Buildings* **2025**, *15*, 3743. [CrossRef]
58. Yoon, J.; Yang, E.; Lee, J.; Hwang, S.J. Lifelong learning with dynamically expandable networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
59. Zhou, H.; Chen, J.; Li, J.; Yang, L.; Li, Y.; Miao, X. Bring memristive in-memory computing into general-purpose machine learning: A perspective. *APL Mach. Learn.* **2023**, *1*, 040901. [CrossRef]
60. Rasch, M.J.; Mackin, C.; Le Gallo, M.; Chen, A.; Fasoli, A.; Odermatt, F.; Li, N.; Nandakumar, S.R.; Narayanan, P.; Tsai, H.; et al. Hardware-aware training for large-scale and diverse deep learning inference workloads using in-memory computing-based accelerators. *Nat. Commun.* **2023**, *14*, 5282. [CrossRef]
61. Ambrogio, S.; Narayanan, P.; Tsai, H.; Shelby, R.M.; Boybat, I.; di Nolfo, C.; Sidler, S.; Giordano, M.; Bodini, M.; Farinha, N.C.P.; et al. Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* **2018**, *558*, 60–67. [CrossRef]
62. Yi, S.-i.; Kendall, J.D.; Williams, R.S.; Kumar, S. Activity-difference training of deep neural networks using memristor crossbars. *Nat. Electron.* **2023**, *6*, 45–51. [CrossRef]
63. Li, C.; Belkin, D.; Li, Y.; Yan, P.; Hu, M.; Ge, N.; Jiang, H.; Montgomery, E.; Lin, P.; Wang, Z.; et al. Efficient and self-adaptive in-situ learning in multilayer memristor neural networks. *Nat. Commun.* **2018**, *9*, 2385. [CrossRef]
64. Cramer, B.; Billaudelle, S.; Kanya, S.; Leibfried, A.; Grübl, A.; Karasenko, V.; Pehle, C.; Schreiber, K.; Stradmann, Y.; Weis, J.; et al. Surrogate gradients for analog neuromorphic computing. *Proc. Natl. Acad. Sci. USA* **2022**, *119*, e2109194119. [CrossRef]
65. Wang, W.; Li, Y.; Wang, M. Difficulties and approaches in enabling learning-in-memory using crossbar arrays of memristors. *Neuromorph. Comput. Eng.* **2024**, *4*, 032002. [CrossRef]
66. Davies, M.; Wild, A.; Orchard, G.; Sandamirskaya, Y.; Guerra, G.A.F.; Joshi, P.; Plank, P.; Risbud, S.R. Advancing neuromorphic computing with Loihi: A survey of results and outlook. *Proc. IEEE* **2021**, *109*, 911–934. [CrossRef]
67. Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **2018**, *38*, 82–99. [CrossRef]
68. Furber, S. Large-scale neuromorphic computing systems. *J. Neural Eng.* **2016**, *13*, 051001. [CrossRef]
69. Neftci, E.O.; Mostafa, H.; Zenke, F. Surrogate gradient learning in spiking neural networks. *IEEE Signal Process. Mag.* **2019**, *36*, 51–63. [CrossRef]
70. Lesort, T.; Lomonaco, V.; Stoian, A.; Maltoni, D.; Filliat, D.; Diaz-Rodriguez, N. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Inf. Fusion* **2020**, *58*, 52–68. [CrossRef]
71. Maltoni, D.; Lomonaco, V. Continuous learning in single-incremental-task scenarios. *Neural Netw.* **2019**, *116*, 56–73. [CrossRef]
72. Hassabis, D.; Kumaran, D.; Summerfield, C.; Botvinick, M. Neuroscience-inspired artificial intelligence. *Neuron* **2017**, *95*, 245–258. [CrossRef]
73. Richards, B.A.; Lillicrap, T.P.; Beaudoin, P.; Bengio, Y.; Bogacz, R.; Christensen, A.; Clopath, C.; Costa, R.P.; de Berker, A.; Ganguli, S.; et al. A deep learning framework for neuroscience. *Nat. Neurosci.* **2019**, *22*, 1761–1770. [CrossRef]
74. Roy, K.; Jaiswal, A.; Panda, P. Towards spike-based machine intelligence with neuromorphic computing. *Nature* **2019**, *575*, 607–617. [CrossRef]

75. Zenke, F.; Ganguli, S. SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Comput.* **2018**, *30*, 1514–1541. [CrossRef]
76. Bellec, G.; Scherr, F.; Subramoney, A.; Hajek, E.; Salaj, D.; Legenstein, R.; Maass, W. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat. Commun.* **2020**, *11*, 3625. [CrossRef]
77. Ornes, S. Can neuromorphic computing help reduce AI's high energy cost? *Proc. Natl. Acad. Sci. USA* **2025**, *122*, e2528654122. [CrossRef]
78. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Rome, Italy, 27–29 February 2016.
79. Santurkar, S.; Tsipras, D.; Ilyas, A.; Madry, A. How does batch normalization help optimization? In Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, QC, Canada, 2–8 December 2018; Volume 31, pp. 2483–2493.
80. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
81. Valencia, C.H.; Vellasco, M.M.B.R.; Figueiredo, K. Echo State Networks: Novel reservoir selection and hyperparameter optimization model for time series forecasting. *Neurocomputing* **2023**, *545*, 126317. [CrossRef]
82. Hoover, B.; Liang, Y.; Pham, B.; Panda, R.; Strobelt, H.; Chau, D.H.; Zaki, M.; Krotov, D. Energy transformer. In Proceedings of the 37th Conference on Neural Information Processing Systems (NIPS 2023), New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 27532–27559.
83. Malladi, S.; Gao, T.; Nichani, E.; Damian, A.; Lee, J.D.; Chen, D.; Arora, S. Fine-tuning language models with just forward passes. In Proceedings of the 37th Conference on Neural Information Processing Systems (NIPS 2023), New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 53038–53075.
84. Scialom, T.; Chakrabarty, T.; Muresan, S. Fine-tuned language models are continual learners. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 6107–6122.
85. Huang, J.; Cui, L.; Wang, A.; Yang, C.; Liao, X.; Song, L.; Yao, J.; Su, J. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 11–16 August, 2024; pp. 1416–1428.
86. Valencia-Niño, C.H.; Nuñez-Rodriguez, R.A.; Vellasco, M.M.B.R.; Marin, J. ESNformer: A hybrid reservoir–transformer architecture for interpretable, position-aware classification of structured assessment data, with a Braille-literacy case study. *Technologies* **2026**, *14*, 517. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.