The Source of Language Capability in an Energy-Based Associative Substrate

Local rules, gradient descent, and a data-grown context graph on a common character-level harness

Marcel Langjahr

Pantareon GmbH
https://pantareon.io/
langjahr@pantareon.io

June 2026

Abstract

We ask a narrow but decisive question: when a decentralised, energy-based associative substrate is made to model language, does the capability come from the substrate's native dynamics, from the learning rule applied to it, or from neither? Using a common character-level harness—identical tokenisation, an explicit train/validation split, and a held-out bits-per-character metric—we sweep a family of mechanisms that differ in how, or whether, parameters are adjusted: purely local update rules on a shared associative architecture (Hebbian, Oja, delta), an energy-based local rule (equilibrium propagation), a gradient-trained recurrent substrate with relaxation dynamics and per-unit learned forgetting, a parameter-matched decoder-only Transformer, and two parameter-free context models grown directly from the data (context-tree weighting and a variable-order context graph).

The result is not the one the substrate programme expected. The local associative and energy-based rules do not learn language: Hebbian, Oja, and equilibrium propagation all sit at or above the letter-frequency bound, and the error-correcting delta rule only just dips below it. The gradient-trained substrate is competitive with the Transformer and in fact beats it at this scale. But the single best model on the 107k-character corpus is the parameter-free context graph at 2.44 bits/char, ahead of the gradient substrate (2.70), context-tree weighting (2.73), and the parameter-matched Transformer (3.36)—with no learned parameters at all. Two controlled experiments agree on the negative half of the picture: on a compositional-generalisation task a local predictive-coding rule reaches 45% where backpropagation reaches 83% under an identical architecture, and a continual-learning probe shows catastrophic forgetting in every model. The corrected conclusion is that small-scale language capability here comes from an explicit, variable-order *context* structure—reachable by gradient descent, but obtained most efficiently by a data-grown context graph—and not from the substrate's fixed associative dynamics. The architectural consequence is to keep the certified Hebbian engine as a distributed knowledge and consensus layer, while the sequence component points to a data-grown context graph, the direction we take up in further work.

Scope of this paper: The experiments below are deliberately small—character-level models of $\sim 10^5$ parameters (or none) trained on 10^5–10^6 characters on a single workstation. They are not claims about frontier-scale behaviour. Their purpose is comparative and mechanistic: to isolate the contribution of the *mechanism* by holding architecture, data, and evaluation fixed. Absolute bits-per-character figures should be read only relative to one another and to the stated baselines. Where a parameter-free method outperforms a learned one, the claim is specifically about this scale and this corpus; the Bitter Lesson [12] concerns the opposite, asymptotic regime, and we address the tension explicitly in the Discussion.

1 Introduction

The substrate studied here descends from the same commitment as Emergent Graph Theory: that function is carried by a *dynamic* graph of correlations among elements rather than by fixed pairwise weights, the view formalised in the correlation theory of brain function [1]. In the associative engine this appears as binding by phase, recall by relaxation into an attractor, and learning by Hebbian

covariance, with a resonance decay that lets unused structure fade. The engine is decentralised and deterministic by construction.

A natural hope follows: that such a substrate, exposed to enough text, would acquire language from its own dynamics—without the global, biologically implausible credit assignment of backpropagation. This paper tests that hope directly, and then widens the test. We separate two things that are easily conflated. One is the *architecture*: relaxation, forgetting, phase-binding. The other is the *learning rule*: how parameters are adjusted from prediction error. By running the same architecture under different learning rules on the same data, and by including a Transformer as an external reference, we can ask where the capability lives. We then add a third axis the original dichotomy leaves out—the *model class*—by including parameter-free context models that are grown from the data rather than trained at all. This lets us ask not only whether the gradient or the dynamics carries the capability, but whether either is necessary.

2 Experimental Setup

Common harness. All models share one interface. Text is tokenised at the character level with a vocabulary read from the corpus; no external tokeniser is used. The corpus is split into a training portion and a held-out validation tail. Performance is reported as validation bits-per-character (bpc), the cross-entropy of next-character prediction in bits, measured on text the model never trains on. The split makes the distinction between memorisation and generalisation visible: training loss alone is uninformative, as a sufficiently expressive model drives it to zero on a small corpus while validation loss rises. Three reference points anchor every run: a uniform code ($\log_2 V$), a unigram (letter-frequency) code, and a bigram code. We treat "below the bigram bound" as the threshold for genuine use of context, and "at or above the unigram bound" as the absence of any language signal.

Model A — gradient-trained substrate. A recurrent core in which the hidden state settles toward a tanh attractor by a leaky relaxation each step, and in which each unit carries a learned retention $\lambda \in (0, 1)$ acting as a resonance/forgetting time-constant. It is trained by truncated backpropagation through time. This model embodies the substrate's *dynamics* (relaxation, forgetting) under a *gradient* learning rule.

Model B — Transformer. A decoder-only Transformer with multi-head causal self-attention, pre-norm residual blocks, a position embedding, and next-token prediction [2]. Its hand-written gradients were verified against finite differences (worst relative error 1.2×10^{-5}) so that it is a correct, not a strawman, baseline.

Model C — native associative engine, and the local-rule family. The faithful mechanism of the certified substrate, using *no backpropagation*. State and concepts are complex unit-phasor vectors; similarity is the real part of the inner product (phase-binding, in the manner of holographic/vector-symbolic memories [3]). Recall is one relaxation—a single associative pass $W\mathbf{h}$ on the bounded energy shell. The associative memory W is built online by Hebbian outer-product accumulation of (next-code, context) pairs with a decay term (resonance forgetting). Only a single read-out temperature is calibrated; the memory itself is never trained by gradient. This is "forgetting and Hebbian covariance as the optimiser", the substrate's own learning rule. Around the same architecture and fixed random codes we also run two further local rules—*Oja*'s normalised Hebbian rule and the error-correcting *delta* (Widrow–Hoff) rule—and, separately, *equilibrium propagation* [6], an energy-based local learning rule in the same biologically-motivated, non-backprop family. These let us ask whether the substrate's failure, if it fails, is specific to pure Hebbian accumulation or general to local learning.

Parameter-free context models. Finally we include two methods with *no learned parameters*, grown directly from the data. *Context-tree weighting* (CTW) [7] maintains Krichevsky–Trofimov counts over all bounded-depth suffix contexts and mixes their predictions by Bayesian weighting; we run it both symbol-native (depth 8) and over the raw bitstream. *CTXGraph* is a variable-order context graph (depth ≤ 4): a generalisation of the context tree in which context states may be shared rather than strictly suffix-nested, with KT-style counts and an optional *conservation* constraint on the weighting. Like CTW it has no trained parameters; its structure is grown from the corpus, in the spirit of variable-order context modelling [8]. These methods sit on the same harness and are scored by the same held-out bpc.

Models A and B are parameter-matched at two operating points (a smaller ~ 7–9×10^4 configuration used for the broad sweep, and a larger ~ 2.2–2.4×10^5 configuration used for the two-corpus comparison)

and trained at the same learning rate. The local-rule family runs at $\sim 7 \times 10^4$ parameters; the context models have none. All language runs use early-stopping at the validation minimum, since every parametric model overfits small corpora.

3 Compositional Generalisation

Before language, we isolate the claim most often attached to local, energy-based learning: that it generalises *compositionally* where standard methods do not. We use a minimal, bounded-length SCAN "add-jump" task [4]: a primitive is seen during training only in isolation and must then be used in modifier compositions (TWICE, THRICE) it was never trained in. The decoder uses a shared output head with learned slot positions, so the held-out action is reachable in any position—otherwise the task is impossible by construction.

We compare a local predictive-coding rule against backpropagation under an *identical* architecture, loss, and data, with raw token inputs and learned embeddings for both; only the weight-update rule differs. Averaged over which primitive is held out and over seeds, both rules fit the training set completely, but their generalisation differs sharply: backpropagation reaches 83% exact-match on the held-out compositions, while the local predictive-coding rule reaches 45% and stays in the 17–45% range across inference settings. Full relaxation does not close the gap. On this task the local rule confers no compositional advantage; the compositional ability that appears is carried by the architecture (the shared decoder) and is obtained with ordinary backpropagation. This is consistent with the result that predictive coding approximates backpropagation only in a limit [5], and that systematic generalisation in this family is hard without either meta-learning or a built-in grammar.

4 Language Modelling

We run the full mechanism sweep on a 107k-character corpus in a constructed/philosophical register with vocabulary 103 (harder, locally irregular). Table 1 reports held-out bits-per-character at the validation minimum, ordered best to worst, with the three reference baselines below the rule. The parametric models in this sweep are at the smaller ~ 7–9×10^4 operating point; the context models carry no parameters. We then report a separate, parameter-matched two-corpus comparison of A, B, and C at the larger scale in Table 2.

Table 2: Parameter-matched two-corpus comparison at the larger ~ 2.2–2.4×10^5 scale (Models A, B) and the native engine (C). Dashes: not run.

Corpus (chars, vocab)	A: subst. (BP)	B: Transf.	C: Hebb.
Codex (107k, 103)	**2.40**	2.69	—
Prose (291k, 83)	**2.07**	2.20	5.13
uniform baseline	6.69/6.38		

Five observations follow.

First, and most consequential: the best model on this corpus has no learned parameters. The variable-order context graph (CTXGraph, conservation off) reaches 2.44 bpc, ahead of the gradient-trained substrate (2.70, $\sim$0.25 bits), symbol-native CTW (2.73), and the parameter-matched Transformer (3.36, $\sim$0.92 bits). A method that is *grown* from the data, not trained, is the most efficient learner in the comparison. This is the result that revises the substrate programme's expectation: the capability does not require gradient credit assignment, and at this scale it is not best served by it.

Second: local learning does not learn language. The Hebbian rule (4.81) and Oja's rule (4.90) sit at the unigram bound, and equilibrium propagation—an energy-based local rule from the same family the substrate belongs to—is worse still (5.56, above unigram). Only the error-correcting delta rule shows a faint signal (4.56, just below unigram), as expected for a rule that is a step toward, not away from, gradient descent. The negative result is therefore general to local learning here, not an artefact of pure Hebbian accumulation. This corroborates and extends the native-engine finding (Model C) in Table 2.

Third: the two negatives are not the same negative. CTXGraph and CTW are also non-gradient, also parameter-free—and they win. So the failure of the Hebbian family is not intrinsic to "no backpropagation" or to "no learned parameters". It is specific to the *mechanism*: a fixed-capacity associative outer-product memory under interference [9], in which the number of overlapping associations far exceeds what the substrate can hold without mutual erasure. The productive non-gradient structure is an explicit, variable-order record of context, not an associative matrix.

Fourth: the conservation constraint is costly. Turning it on collapses CTXGraph from 2.44 to 3.40 bpc—roughly the bigram bound—a penalty of $\sim$0.96 bits, the difference between genuine context use and almost none. Whatever invariant the constraint enforces on the weighting, it trades away most of the model's ability to fit local context statistics on this corpus. We report this as a measured

Table 1: Held-out bits/char at the validation minimum on the 107k corpus (vocab 103), lower is better. "Reads context?" is judged against the bigram bound (3.41): *yes* = below it (genuine context use), *faint* = below unigram only, *no* = at or above the unigram bound. Parametric models are at the $\sim$7–9 $\times 10^4$ operating point; context models have no learned parameters and are grown from the data.

Method	Val. bpc	Learned params	Reads context?
CTXGraph (D= 4, conserve off)	**2.44**	0 (grown)	**yes (best)**
Substrate (BPTT, Model A)	2.70	93.9k	yes
CTW (symbol-native, D= 8)	2.73	0 (grown)	yes
CTW (binary bitstream, D= 16 bits)	3.11	0 (grown)	yes
Transformer (Model B)	3.36	80.1k	yes
CTXGraph (D= 4, conserve on)	3.40	0 (grown)	yes ($\approx$ bigram)
Delta rule (Widrow–Hoff)	4.56	70.7k	faint
Hebbian (Model C family)	4.81	70.7k	no
Oja	4.90	70.7k	no
Equilibrium propagation	5.56	79.3k	no
baselines: uniform 6.69 unigram (letter freq.) 4.63 bigram 3.41			

property of the estimator and do not extrapolate it to the conservation principles of the underlying physical framework, which concern a different object; the relationship is a question for the deeper work, not a claim of this paper.

Fifth: the gradient substrate still edges the Transformer, and the edge shrinks with scale. The substrate beats the Transformer by +0.66 bpc at $\sim$90k parameters (Table 1), by +0.29 at $\sim$220k (Table 2, Codex), and by +0.13 at $\sim$220k on the larger 291k corpus (Prose). Two of these vary model size and one varies data size, so this is a consistent *direction* across both kinds of scale rather than a single clean curve—the direction expected if the Transformer's long-range advantage is largely unused at short context and small data and grows as scale grows. The two tables are at different operating points; at the larger scale both A and B improve (on Codex, substrate $2.70 \to 2.40$, Transformer $3.36 \to 2.69$), and the substrate retains its lead. We make no claim beyond the measured trend; whether the substrate's margin stabilises at a small positive value or crosses zero requires a larger corpus, and whether CTXGraph's lead survives the same scaling is the more important open question.

Sample continuations are consistent with the figures: the gradient substrate and the Transformer emit orthographically valid words and local fragments of the source register without coherent meaning, while the Hebbian engine produces near-random character strings. None of the parametric models is fluent; $\sim$2 bits/char is a modest character model, and a parameter-free context graph reaching it is a statement about efficiency, not about fluency.

5 Continual Learning and Forgetting

A second motivation attached to the substrate is that, unlike a frozen Transformer, it reaches "no final state" and learns from every input. We test this with a prequential protocol—predict, then update—on a stream that alternates between two styles (A and B, where B is a letter-permutation of A with the same alphabet but disjoint statistics), in the order $A\,B\,A\,B$. Fixed held-out probes for each style are measured throughout.

Two facts emerge. First, continual online learning is *not* distinctive to the substrate: a Transformer trained on the same stream also learns with every input and also reaches no final state. "Frozen weights" in deployed systems is an operational choice, not an architectural limit. Second, every model forgets catastrophically [10]: when the stream switches to B, the held-out loss on A rises in all cases; the substrate's resonance forgetting does not solve this. The substrate does exhibit *smaller* swings—its probe-A loss moves in a narrower band than the Transformer's, which learns each style more deeply but loses it more sharply—but this is confounded with weaker plasticity: the substrate also reaches a higher (worse) best loss on each style, so it has less to lose. All models show "savings" (faster re-learning on return). Whether selective forgetting yields a genuine stability advantage—low forgetting *without* sacrificing plasticity—is not established by these runs, and would require matching plasticity before comparing forgetting [11].

6 Discussion

The experiments cohere on the negative result and revise the positive one. On a compositional task the local rule underperforms backpropagation under identical architecture; on language the substrate's native rule, and local learning generally, fails to compress beyond letter frequency; and continual learning, the substrate's most plausible distinctive capability, is shared by the Transformer and leaves catastrophic forgetting unsolved. Removing the gradient from the substrate while keeping its dynamics (Model A $\rightarrow$ Model C) collapses performance from competitive to near-trivial. So far this matches the substrate programme's worst case.

But the sweep refutes the tempting inference that the capability therefore lives *in the gradient*. The best model on the corpus, CTXGraph, uses no gradient and no learned parameters, and beats both the gradient substrate and the parameter-matched Transformer. The operative variable is not the presence of a gradient; it is the inductive structure. An explicit, variable-order record of context—whether realised as a grown context graph or reached by gradient descent on a recurrent substrate—reads the corpus; a fixed associative outer-product memory does not, no matter how it is updated. Gradient descent is one route to the right structure; a data-grown context graph is another, and at this scale a markedly more parameter-efficient one.

This sharpens, rather than illustrates, the Bitter Lesson [12]. The standard reading—generic computation under search and learning beats hand-designed structure—describes the asymptotic, large-scale regime. Our regime is the opposite one, explicitly: $\sim 10^5$ characters and $\sim 10^5$ parameters or fewer. Here a well-matched structural prior (a context graph) beats a generic gradient model, exactly as the Bitter Lesson's own scale-caveat would predict, and consistent with the substrate-vs-Transformer margin shrinking as scale grows. The lesson for the programme is not "structure loses" but "the structure must match the task": the substrate's dynamics are a structural prior *for associative recall*, which is the wrong prior for sequence prediction; the context graph is a structural prior *for sequence prediction*, which is the right one. The original commitment—that function lives in a dynamic graph of correlations grown from data rather than in fixed pairwise weights [1]—is, read this way, vindicated by CTXGraph and falsified by the fixed Hebbian matrix, which is precisely the "fixed pairwise weights" the commitment opposed. We state this as a consistent reading, not as a derivation: CTXGraph is a classical context model, not something the framework predicted, and the connection is one we test in the deeper work rather than assert here.

It does not follow that the certified associative substrate is without value. Its established strengths—decentralisation, deterministic and certifiable state, content-addressable associative recall—are precisely the properties of a *memory and coordination* layer, not of a language model. The same Hebbian mechanism that fails to compress language is a natural fit for a distributed, verifiable knowledge store that competent sequence models query and over which they reach consensus. The architectural consequence is to separate concerns that the original programme fused: sequence prediction and reasoning from models with the right context structure; knowledge from a retrieval layer; decentralisation and certification from the associative engine acting as the shared substrate between nodes. Within the parameter-free, graph-grown family the programme prefers, the result also names the sequence component: a data-grown context graph, not the associative matrix. This is the direction the next line of work takes up.

7 Conclusion

Holding architecture, data, and evaluation fixed and sweeping the mechanism, we find that the best small-scale language model on this corpus is a parameter-free, data-grown context graph (2.44 bpc), ahead of a gradient-trained substrate (2.70), context-tree weighting (2.73), and a parameter-matched Transformer (3.36). The substrate's native Hebbian rule, and local learning generally—Hebbian, Oja, and equilibrium propagation—do not learn language, plateauing at or just below the letter-frequency bound; only the error-correcting delta rule shows a faint signal. The corrected conclusion is that small-scale language capability here comes from an explicit, variable-order context structure—reachable by gradient descent, obtained most efficiently by a data-grown context graph—and not from the substrate's fixed associative dynamics, nor from the presence of a gradient per se. Two further controlled experiments—compositional generalisation and continual learning—agree on the negative half. The result is negative for the strong thesis that intelligence emerges from the substrate's native dynamics, and constructive for an architecture in which the certified decentralised engine serves as a knowledge and consensus layer beneath a data-grown context model. The natural next measurements are whether the context graph's small-scale lead survives at larger corpus sizes and against a long-context Transformer,

and what invariant the conservation constraint is buying at the cost it charges.

References

[1] von der Malsburg, C., "The Correlation Theory of Brain Function", in *Models of Neural Networks II*, E. Domany, J. L. van Hemmen, K. Schulten (eds.), Springer, New York, pp. 95–119 (1994); orig. MPI für Biophysikalische Chemie, Internal Report 81-2 (1981).

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., "Attention Is All You Need", *Advances in Neural Information Processing Systems* **30** (2017).

[3] Plate, T. A., "Holographic Reduced Representations", *IEEE Transactions on Neural Networks* **6**(3), 623–641 (1995).

[4] Lake, B. M., Baroni, M., "Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks", *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 2873–2882 (2018).

[5] Whittington, J. C. R., Bogacz, R., "An Approximation of the Error Backpropagation Algorithm in a Predictive Coding Network with Local Hebbian Synaptic Plasticity", *Neural Computation* **29**(5), 1229–1262 (2017).

[6] Scellier, B., Bengio, Y., "Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation", *Frontiers in Computational Neuroscience* **11**, 24 (2017).

[7] Willems, F. M. J., Shtarkov, Y. M., Tjalkens, T. J., "The Context-Tree Weighting Method: Basic Properties", *IEEE Transactions on Information Theory* **41**(3), 653–664 (1995).

[8] Cleary, J. G., Witten, I. H., "Data Compression Using Adaptive Coding and Partial String Matching", *IEEE Transactions on Communications* **32**(4), 396–402 (1984).

[9] Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. Natl. Acad. Sci. USA* **79**(8), 2554–2558 (1982).

[10] McCloskey, M., Cohen, N. J., "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem", *Psychology of Learning and Motivation* **24**, 109–165 (1989).

[11] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., et al., "Overcoming Catastrophic Forgetting in Neural Networks", *Proc. Natl. Acad. Sci. USA* **114**(13), 3521–3526 (2017).

[12] Sutton, R. S., "The Bitter Lesson", *incompleteideas.net/IncIdeas/BitterLesson.html* (2019).