Emergent Language

Sequence, Memory, and Topic Structure from a Conserving Energy Graph
The Language Mode of Emergent Graph Theory

Marcel Langjahr

Pantareon
Pantareon Foundations

langjahr@pantareon.io · https://pantareon.io/

July 2026

Abstract

We present *Emergent Language*, the language and memory mode of the Emergent Graph Theory (EGT) engine. The system learns language without neural networks, gradients, or backpropagation: text is injected as energy into a finite reservoir, and linguistic structure condenses as the graph configuration that optimises energy flow under strict conservation. Every learning step is a balanced energy transaction (global ledger drift $< 10^{-9}$ relative), every run is bit-for-bit reproducible, and an interrupted run resumed from a checkpoint is byte-identical to an uninterrupted one — properties that make the system auditable by construction.

Empirically, the substrate learns on three levels. *Sequence:* a belief-walk readout over node identity achieves variable-order prediction, surpassing the honest count-based controls that structurally cap simpler readouts — on a 22.3M-character corpus it reaches 1.561 bits per character against 2.38 (trigram) and 1.80 (4-gram), with the margin growing monotonically with corpus size; on German prose it surpasses even the 5-gram control. *Word classes:* at word granularity, the graph carries distributional word classes that are linguistically real — nearest neighbours of `cat` are animals, of `Lily` girls' names, of `sad` negative emotions — with curated-group purity two orders of magnitude above chance. *Topic:* an episode layer in the physics itself (a slow radiation field collected at story boundaries into undirected co-activation bonds) stores document-level thematic structure — `park`→{slide, swing, grass, birds}, `beach`→{sand, tide, jellyfish} — an information class that no n-gram table of any order contains, added at zero cost to sequence performance.

All results were produced on a single desktop CPU; the largest training run (22.3M symbols) completes in roughly a day, and a 100k-symbol corpus trains in under a minute. Models are compact (a 2.8 MB graph serves live in-browser inference), learn incrementally without retraining, and can forget under an energy-metabolism regime at constant throughput. The measured data-scaling curve is positive and unsaturated; the primary limitation of the present programme is iteration speed on available compute, not architecture.

Availability of Source Code: The engine (Rust core with Python orchestration) is proprietary to Pantareon and is not released as open source. Access can be granted upon reasonable request for technical due diligence or scientific verification, subject to a non-disclosure agreement. Licensing, acquisition, and funded co-development of the technology are open to discussion.

Scope of this Paper: This report presents the language mode of the EGT engine and its measured results to date. It is a technical status report intended for partners and investors with scientific standards; it is not submitted for peer review. All numbers reported are from deterministic, reproducible runs against held-out data with count-based controls trained on identical data. The theoretical framework is presented in the companion paper *Emergent Graph Theory (EGT)*; the engine shares its operator grammar and conservation discipline with the cosmological and molecular modes reported elsewhere.

1 Introduction

Contemporary language AI is dominated by one architecture family: large neural networks trained by gradient descent. The results are impressive, but the architecture brings structural costs that are

increasingly economic rather than technical: training is a monolithic, energy-intensive batch process; incremental learning requires retraining or fragile fine-tuning; the models are opaque, so auditability must be bolted on from the outside; and inference at quality requires substantial hardware.

Emergent Language explores a different point in the design space, derived from the EGT premise that stable structure is the configuration a substrate reaches when optimising energy flow under conservation. Text is not fitted — it is *ingested*: each symbol precipitates from a finite energy reservoir as a graph node, co-active nodes bind (releasing energy, which is why dense, regularity-capturing structure is the optimum), activation flows along learned bonds, idle structure is metabolised back into the reservoir, and context-equivalent nodes consolidate into higher-order classes. Learning, memory, and forgetting are one energy flux.

Three properties follow from this construction rather than being engineered in afterwards:

Auditability by construction. Every operator is a balanced energy transaction; the global ledger closes to $< 10^{-9}$ relative drift over full training runs. Runs are bit-for-bit reproducible, and a run interrupted and resumed from a checkpoint produces a byte-identical model. Any output can be traced to explicit graph structure — there are no opaque weights.

Incrementality. The model learns online, symbol by symbol. New text extends the graph; there is no retraining step. Under the metabolism regime (forgetting enabled), the graph runs at a bounded size and constant throughput (measured: 72,000 symbols/s sustained), making lifelong on-device learning a native operating mode rather than a research problem.

Inspectability. The learned object is a graph over symbol classes. Word classes, association structure, and topic clusters are directly queryable — the instruments reported below (§3) read them out in milliseconds.

The present programme asks how much of language such a substrate can capture, and measures it against the strictest cheap controls available: count-based n-gram models of matching and higher order, trained on identical data, evaluated on held-out text.

2 The Language Mode

2.1 Operators

The mode runs five conserving operators on a graph $G = (V, E)$ with node energies (rest mass, agitation, radiation) and edge tensions, plus a south reservoir holding the total budget:

Injection converts reservoir mass into a symbol node bonded to the previous symbol (the write head). *Binding* converts co-active agitation into bond tension plus released heat — the learning drive: bound configurations are lower-energy, so structure that captures symbol regularities is thermodynamically preferred. *Flow* diffuses agitation along tension-weighted bonds (recall relaxation). *Forgetting* returns idle tension and isolated nodes to the reservoir — the structure–reservoir metabolism. *Consolidation* merges context-equivalent nodes.

2.2 Variable-order consolidation

Consolidation is the mechanism that lifts the model above fixed-order statistics. Nodes bucket by the *identity* of their predecessor class: once a class exists, its successors bucket per class, so each consolidation pass can deepen the encoded context by one symbol where the corpus supports it — a variable-order, causal-state-style partition rather than a fixed n-gram order. A second phase fuses classes of the same symbol whose *future path distributions* (depth-3, tension-weighted walks) are near-identical across different histories — a bisimulation-style generalisation step, guarded by budgets and a mass criterion so single observations never fuse. Both phases are exact energy conversions; a merge-fairness scheduler distributes the per-pass budget round-robin across contexts.

2.3 The episode layer

Sequence bonds encode *what follows what*. Document-level meaning — *what belongs together in one story* — is a different information class, and the engine stores it physically. Injection deposits a slow *radiation* field on each node, damped by $1/\sqrt{n_{\text{tag}}}$ of the symbol's cumulative frequency (Zipf damping: rare, content-bearing words radiate; function words barely do). Flow does not move radiation and binding does not consume it, so at any instant the radiating nodes are the episode's active cast. At the story separator the episode resets: the strongest radiators pair into *undirected* co-activation bonds (marked distinctly from directed sequence bonds, so sequence readouts are untouched), bond tension converted exactly from the partners' radiation, the remainder returning to the reservoir. The episode ends energetically closed; budgets keep the layer linear in corpus size.

2.4 Readout

The primary readout is a *belief walk*: an online forward algorithm over node identity. A belief distribution over graph nodes is advanced along tension-weighted directed bonds by each observed symbol; the next-symbol distribution is the belief-weighted mixture of node successor distributions. Node identity is exactly where consolidation stores deep context, so the readout uses whatever order the topology encodes. A decaying recency cache over recently observed symbols supplies discourse-scale coherence. Inference is graph traversal — no matrix products; the full predictor runs client-side in a browser.

3 Measured Results

All experiments: single desktop CPU, deterministic runs, held-out evaluation, controls trained on identical data. Corpora: TinyStories (English children's stories, 22.3M characters), Codex (German prose, 110k characters), and synthetic processes with known entropy rates.

3.1 Sequence: beyond fixed-order prediction

Model	bits/char
unigram	4.443
bigram	3.289
trigram (backoff)	2.380
4-gram (backoff)	1.800
5-gram (backoff)	1.514
belief walk (this work)	**1.561**

Table 1: Held-out prediction, TinyStories, 22.3M training symbols. The belief walk surpasses the 4-gram by 0.24 bpc and approaches the 5-gram at conservative readout settings (support cap 512).

The margin over fixed-order controls *grows monotonically with data*: against the 4-gram control, −0.02 bpc at 0.5M symbols, −0.14 at 1.5M, −0.24 at 22.3M — a positive, unsaturated scaling curve. On German prose (Codex), the belief walk reaches 2.749 bpc, below *every* count control including the 5-gram (2.814). On a synthetic third-order process on which trigrams are provably blind (1.00 bpc floor, true rate 0.469), the belief walk reaches 0.514 — and under aggressive consolidation the same process compresses into 1,611 nodes (30× fewer than raw) while remaining far below the trigram floor: the consolidation is a genuine state-machine compressor.

A control worth reporting: on a shuffled-word corpus with no structure above second order between words, the readout lands exactly between the trigram and 4-gram controls — it finds the intra-word structure that exists and hallucinates nothing beyond it.

3.2 Word classes: queryable distributional structure

At word granularity (vocabulary 7,682 on a 1.2M-token subset), each word's future signature — the same statistic the consolidation fuses on — yields nearest-neighbour structure that is linguistically real:

probe	nearest neighbours
cat	rabbit, bird, frog, bear
Lily	Lucy, Amy, Mia, Lila, Sara
sad	scared, upset, embarrassed, guilty
smiled	nodded, shrugged, sighed, giggled
wanted	wants, want, decided, liked
in	through, near, at, into

Table 2: Distributional neighbours by future-signature cosine. Animals cluster with animals, names with names, emotions with emotions, prepositions with prepositions — with no linguistic supervision of any kind.

Under a strict curated-group purity metric (a top-5 neighbour drawn from the full 4,787-word candidate vocabulary must land in the probe's own 7-word group), the system scores 15.7% against a chance level of 0.13% — two orders of magnitude above chance. The graph carries this structure losslessly and makes it queryable in milliseconds.

3.3 Topic: an information class n-grams cannot hold

The episode layer, trained on 6,026 stories, stores document co-occurrence as physical bonds:

probe	episode associates
park	slide, swing, grass, birds, trees
kitchen	stove, faucet, plate, apron
beach	sand, tide, jellyfish, dolphin, wave
snow	igloo, snowball, freeze, winter
boat	sail, port, anchor, ocean

Table 3: Thematic associates from undirected episode bonds. This is document-level information that no sequence model or n-gram table of any order contains.

The layer is additive in the best sense: with it enabled, the word-level predictor reaches its best value

overall (5.311 bits/token with the recency cache) — the topic memory costs sequence performance nothing. For generation, a story-coherence instrument is in place, including a target anchor measured on real corpus stories (topic-cohesion 0.355); closing the gap between generated and real cohesion is the declared objective of the scaling programme (§4).

3.4 Systems properties

- **Exact ledger:** global energy drift $< 10^{-9}$ relative across full training runs; verified by a committed conservation gate.
- **Bit-reproducibility:** two identical runs produce byte-identical models; interrupted-and-resumed runs are byte-identical to uninterrupted ones (verified gates, including torn-checkpoint recovery).
- **Throughput:** active-front scaling keeps operator cost on the working set rather than the graph; a 100k-symbol corpus trains in 56 seconds; under the metabolism regime throughput is constant at 72,000 symbols/s regardless of corpus length.
- **Footprint:** a trained conversational model of 2.8 MB serves live in-browser inference (no backend); the 22.3M-symbol model is 431 MB as plain JSON.
- **Determinism as a product feature:** every prediction is exactly reproducible and attributable to inspectable graph structure — a compliance property gradient-trained models cannot offer natively.

4 Scaling Outlook

Every result above was produced on one desktop CPU, with the engine single-threaded for determinism. This is the honest frame for what remains open: the questions are defined, instrumented, and measurable — what they require is iteration speed.

The measured curves say scale helps. The sequence margin over fixed-order controls grows with every corpus-size step tested, without saturation. The word-level track, which trades statistical density for meaning-bearing units, is data-hungry by nature; its current corpus (1.2M tokens) is three orders of magnitude below what neural language models consider small. The direct next steps — the full word-level corpus, subword vocabularies, larger and more diverse corpora — are engineering, not research.

The defined research frontier. Two questions carry the programme's upside. First, *topic-conditioned generation*: the topic information demonstrably exists in the graph (§3.3); coupling it into generation has a ready instrument (the real-story cohesion anchor) and a clear hypothesis to test at stronger substrate scale. Second, the *structure level*: predicate–argument organisation (who does what to whom), the layer that separates fluent text from meaningful text. Both are bounded, testable programmes — but each experiment cycle at the required substrate size takes days on present hardware. With moderate compute (multi-core parallelism of the physics; cluster-parallel experiment sweeps), the same cycles take hours, and the frontier becomes a schedule rather than a hope.

Cost asymmetry. The engine's training cost is linear per symbol with a bounded working set; there is no gradient pass, no optimiser state, no accelerator requirement. The economics of scaling this system differ fundamentally from scaling neural models: compute buys experiment throughput, not just model size.

5 Application Surface

The properties measured above map onto product surfaces that are difficult for gradient-trained models to serve natively:

On-device, private, lifelong memory. Megabyte-scale models that learn incrementally from a user's text stream, forget under an explicit metabolism, run without a backend, and never leave the device.

Auditable language components. Deterministic, ledger-exact, inspectable models for domains where every output must be reproducible and attributable (compliance, safety-critical logging, regulated industries).

Topic and association services. The episode layer yields thematic association and story/episode retrieval directly from structure — queryable without inference passes.

A memory layer for LLM systems. The substrate is complementary to large generative models: a cheap, transparent, continuously learning associative memory that an LLM can read from and write to.

Pantareon offers the technology for licensing, acquisition, or funded co-development.

6 Conclusion

A single conserving operator grammar — the same one that produces baryonic structure in the cosmological mode of the EGT engine — ingests text and yields, measurably: variable-order sequence prediction beyond its honest count controls with a positive scaling curve; linguistically real distributional word classes, queryable from structure; and a physically stored topic layer holding information no sequence statistic contains, at zero sequence cost. It does so deterministically, exactly conserving, incrementally, and on commodity hardware.

The system is not a large language model and does not compete with one on open-ended generation today. It is a different object: a transparent, auditable, continuously learning language substrate whose measured curves all point the same direction — and whose next steps are limited by compute, not by ideas.