Emergent Language

Sequenz, Gedächtnis und Topic-Struktur aus einem erhaltenden Energiegraphen
Der Sprachmodus der Emergent Graph Theory

Marcel Langjahr

Pantareon
Pantareon Foundations

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

Juli 2026

Zusammenfassung

Wir stellen *Emergent Language* vor, den Sprach- und Gedächtnismodus der Engine der Emergent Graph Theory (EGT). Das System lernt Sprache ohne neuronale Netze, Gradienten oder Backpropagation: Text wird als Energie in ein endliches Reservoir injiziert, und sprachliche Struktur kondensiert als diejenige Graphkonfiguration, die den Energiefluss unter strikter Erhaltung optimiert. Jeder Lernschritt ist eine ausgeglichene Energietransaktion (globale Ledger-Drift $< 10^{-9}$ relativ), jeder Lauf ist bit-für-bit reproduzierbar, und ein unterbrochener, aus einem Checkpoint fortgesetzter Lauf ist byte-identisch zu einem ununterbrochenen — Eigenschaften, die das System durch Konstruktion auditierbar machen.

Empirisch lernt das Substrat auf drei Stufen. *Sequenz:* Ein Belief-Walk-Readout über die Knotenidentität erreicht Vorhersage variabler Ordnung und übertrifft dabei die ehrlichen zählbasierten Kontrollen, die einfachere Readouts strukturell deckeln — auf einem Korpus von 22,3M Zeichen erreicht er 1,561 Bit pro Zeichen gegen 2,38 (Trigramm) und 1,80 (4-Gramm), wobei der Abstand monoton mit der Korpusgröße wächst; auf deutscher Prosa übertrifft er sogar die 5-Gramm-Kontrolle. *Wortklassen:* Auf Wortgranularität trägt der Graph distributionelle Wortklassen, die sprachlich real sind — die nächsten Nachbarn von `cat` sind Tiere, die von `Lily` Mädchennamen, die von `sad` negative Emotionen — bei einer Reinheit der kuratierten Gruppen zwei Größenordnungen über dem Zufallsniveau. *Topic:* Eine Episodenschicht in der Physik selbst (ein langsames Strahlungsfeld, das an den Grenzen der Geschichten zu ungerichteten Ko-Aktivierungs-Bindungen eingesammelt wird) speichert thematische Struktur auf Dokumentebene — `park`→{slide, swing, grass, birds}, `beach`→{sand, tide, jellyfish} — eine Informationsklasse, die keine n-Gramm-Tabelle irgendeiner Ordnung enthält, hinzugefügt zu null Kosten für die Sequenzleistung.

Alle Ergebnisse wurden auf einer einzigen Desktop-CPU erzeugt; der größte Trainingslauf (22,3M Symbole) ist in etwa einem Tag abgeschlossen, und ein Korpus von 100k Symbolen trainiert in unter einer Minute. Die Modelle sind kompakt (ein Graph von 2,8 MB bedient Live-Inferenz im Browser), lernen inkrementell ohne Neutraining und können unter einem Energie-Metabolismus-Regime bei konstantem Durchsatz vergessen. Die gemessene Daten-Skalierungskurve ist positiv und ungesättigt; die primäre Begrenzung des gegenwärtigen Programms ist die Iterationsgeschwindigkeit auf der verfügbaren Rechenzeit, nicht die Architektur.

Verfügbarkeit des Quellcodes: Die Engine (Rust-Kern mit Python-Orchestrierung) ist proprietäres Eigentum von Pantareon und wird nicht als Open Source veröffentlicht. Zugang kann auf begründete Anfrage für technische Due Diligence oder wissenschaftliche Verifikation gewährt werden, vorbehaltlich einer Vertraulichkeitsvereinbarung. Lizenzierung, Erwerb und finanzierte Mitentwicklung der Technologie stehen zur Diskussion.

Umfang dieses Papers: Dieser Bericht stellt den Sprachmodus der EGT-Engine und seine bisher gemessenen Ergebnisse dar. Er ist ein technischer Statusbericht für Partner und Investoren mit wissenschaftlichen Ansprüchen; er ist nicht zur Begutachtung eingereicht. Alle berichteten Zahlen stammen aus deterministischen, reproduzierbaren Läufen gegen zurückgehaltene Daten mit zählbasierten Kontrollen, die auf identischen Daten trainiert wurden. Der theoretische Rahmen wird im Begleitpaper *Emergent Graph Theory (EGT)* dargestellt; die Engine teilt ihre Operatorgrammatik und Erhaltungsdisziplin mit den andernorts berichteten kosmologischen

und molekularen Modi.

1 Einleitung

Die gegenwärtige Sprach-KI wird von einer Architekturfamilie beherrscht: großen neuronalen Netzen, die durch Gradientenabstieg trainiert werden. Die Ergebnisse sind beeindruckend, doch die Architektur bringt strukturelle Kosten mit sich, die zunehmend ökonomischer statt technischer Natur sind: Das Training ist ein monolithischer, energieintensiver Batch-Prozess; inkrementelles Lernen erfordert erneutes Training oder fragiles Fine-Tuning; die Modelle sind undurchsichtig, sodass Auditierbarkeit von außen angeflanscht werden muss; und Inferenz auf Qualitätsniveau erfordert erhebliche Hardware.

Emergent Language erkundet einen anderen Punkt im Entwurfsraum, abgeleitet aus der EGT-Prämisse, dass stabile Struktur diejenige Konfiguration ist, die ein Substrat erreicht, wenn es den Energiefluss unter Erhaltung optimiert. Text wird nicht gefittet — er wird *ingestiert*: jedes Symbol präzipitiert aus einem endlichen Energiereservoir als Graphknoten, ko-aktive Knoten binden (wobei Energie frei wird, weshalb dichte, Regelmäßigkeiten erfassende Struktur das Optimum ist), Aktivierung fließt entlang gelernter Bindungen, ungenutzte Struktur wird zurück ins Reservoir metabolisiert, und kontextäquivalente Knoten konsolidieren zu Klassen höherer Ordnung. Lernen, Gedächtnis und Vergessen sind ein einziger Energiefluss.

Drei Eigenschaften folgen aus dieser Konstruktion, statt nachträglich aufgesetzt zu werden:

Auditierbarkeit durch Konstruktion. Jeder Operator ist eine ausgeglichene Energietransaktion; das globale Ledger schließt über vollständige Trainingsläufe auf $< 10^{-9}$ relative Drift. Läufe sind bit-reproduzierbar, und ein Lauf, der unterbrochen und aus einem Checkpoint fortgesetzt wird, ergibt ein byte-identisches Modell. Jede Ausgabe lässt sich auf explizite Graphstruktur zurückführen — es gibt keine undurchsichtigen Gewichte.

Inkrementalität. Das Modell lernt online, Symbol für Symbol. Neuer Text erweitert den Graphen; es gibt keinen Schritt des erneuten Trainings. Im Metabolismus-Regime (Vergessen aktiviert) läuft der Graph bei beschränkter Größe und konstantem Durchsatz (gemessen: 72 000 Symbole/s dauerhaft), womit lebenslanges Lernen auf dem Gerät ein nativer Betriebsmodus wird statt eines Forschungsproblems.

Einsehbarkeit. Das gelernte Objekt ist ein Graph über Symbolklassen. Wortklassen, Assoziationsstruktur und Topic-Cluster sind direkt abfragbar — die unten berichteten Instrumente (§3) lesen sie in Millisekunden aus.

Das vorliegende Programm fragt, wie viel von Sprache ein solches Substrat erfassen kann, und misst es gegen die strengsten verfügbaren billigen Kontrollen: zählbasierte n-Gramm-Modelle gleicher und höherer Ordnung, auf identischen Daten trainiert, auf zurückgehaltenem Text evaluiert.

2 Der Sprachmodus

2.1 Operatoren

Der Modus betreibt fünf erhaltende Operatoren auf einem Graphen $G = (V, E)$ mit Knotenenergien (Ruhemasse, Agitation, Strahlung) und Kantenspannungen, dazu ein Süd-Reservoir, das das Gesamtbudget hält:

Injektion wandelt Reservoirmasse in einen Symbolknoten um, der an das vorangegangene Symbol gebunden ist (der Schreibkopf). *Bindung* wandelt ko-aktive Agitation in Bindungsspannung plus freigesetzte Wärme um — der Lernantrieb: gebundene Konfigurationen sind energieärmer, daher ist Struktur, die Symbolregelmäßigkeiten erfasst, thermodynamisch bevorzugt. *Fluss* diffundiert Agitation entlang spannungsgewichteter Bindungen (Abruf-Relaxation). *Vergessen* gibt ungenutzte Spannung und isolierte Knoten an das Reservoir zurück — der Struktur–Reservoir-Metabolismus. *Konsolidierung* verschmilzt kontextäquivalente Knoten.

2.2 Konsolidierung variabler Ordnung

Die Konsolidierung ist der Mechanismus, der das Modell über Statistik fester Ordnung hinaushebt. Knoten werden nach der *Identität* ihrer Vorgängerklasse einsortiert: sobald eine Klasse existiert, sortieren sich ihre Nachfolger je Klasse ein, sodass jeder Konsolidierungsdurchlauf den kodierten Kontext um ein Symbol vertiefen kann, wo das Korpus es trägt — eine Partition variabler Ordnung nach Art kausaler Zustände statt einer festen n-Gramm-Ordnung. Eine zweite Phase verschmilzt Klassen desselben Symbols, deren *Zukunftspfad-Verteilungen* (spannungsgewichtete Läufe der Tiefe 3) über verschiedene Historien hinweg nahezu identisch sind — ein Generalisierungsschritt nach Art einer Bisimulation, abgesichert durch Budgets und ein Massenkriterium, sodass Einzelbeobachtungen nie verschmelzen. Beide Phasen sind exakte Energieumwandlungen; ein Merge-Fairness-Scheduler verteilt das Budget je Durchlauf im Round-Robin über die Kontexte.

2.3 Die Episodenschicht

Sequenzbindungen kodieren, *was auf was folgt*. Bedeutung auf Dokumentebene — *was in einer Geschichte zusammengehört* — ist eine andere Informationsklasse, und die Engine speichert sie physikalisch. Die Injektion legt auf jedem Knoten ein langsames *Strahlungsfeld* ab, gedämpft mit $1/\sqrt{n_{\text{tag}}}$ der kumulativen Häufigkeit des Symbols (Zipf-Dämpfung: seltene, inhaltstragende Wörter strahlen; Funktionswörter kaum). Der Fluss bewegt Strahlung nicht, und die Bindung verbraucht sie nicht, sodass in jedem Augenblick die strahlenden Knoten die aktive Besetzung der Episode sind. Am Geschichtentrenner setzt sich die Episode zurück: die stärksten Strahler paaren sich zu *ungerichteten* Ko-Aktivierungs-Bindungen (eigens gegenüber gerichteten Sequenzbindungen gekennzeichnet, sodass Sequenz-Readouts unberührt bleiben), die Bindungsspannung wird exakt aus der Strahlung der Partner umgewandelt, der Rest kehrt ins Reservoir zurück. Die Episode endet energetisch geschlossen; Budgets halten die Schicht linear in der Korpusgröße.

2.4 Readout

Der primäre Readout ist ein *Belief-Walk*: ein Online-Forward-Algorithmus über Knotenidentität. Eine Belief-Verteilung über die Graphknoten wird von jedem beobachteten Symbol entlang spannungsgewichteter gerichteter Bindungen fortgeschrieben; die Verteilung des nächsten Symbols ist die belief-gewichtete Mischung der Nachfolgerverteilungen der Knoten. Die Knotenidentität ist genau der Ort, an dem die Konsolidierung tiefen Kontext ablegt, sodass der Readout jene Ordnung nutzt, die die Topologie kodiert. Ein abklingender Aktualitäts-Cache über kürzlich beobachtete Symbole liefert Kohärenz auf Diskursebene. Inferenz ist Graphtraversierung — keine Matrixprodukte; der vollständige Prädiktor läuft clientseitig in einem Browser.

3 Gemessene Ergebnisse

Alle Experimente: eine einzelne Desktop-CPU, deterministische Läufe, Evaluation auf zurückgehaltenen Daten, Kontrollen auf identischen Daten trainiert. Korpora: TinyStories (englische Kindergeschichten, 22,3M Zeichen), Codex (deutsche Prosa, 110k Zeichen) und synthetische Prozesse mit bekannten Entropieraten.

3.1 Sequenz: jenseits der Vorhersage fester Ordnung

Modell	Bit/Zeichen
Unigramm	4,443
Bigramm	3,289
Trigramm (Backoff)	2,380
4-Gramm (Backoff)	1,800
5-Gramm (Backoff)	1,514
Belief-Walk (diese Arbeit)	**1,561**

Tabelle 1: Vorhersage auf zurückgehaltenen Daten, TinyStories, 22,3M Trainingssymbole. Der Belief-Walk übertrifft das 4-Gramm um 0,24 bpc und nähert sich dem 5-Gramm bei konservativen Readout-Einstellungen (Support-Cap 512).

Der Vorsprung gegenüber den Kontrollen fester Ordnung *wächst monoton mit den Daten*: gegen die 4-Gramm-Kontrolle −0,02 bpc bei 0,5M Symbolen, −0,14 bei 1,5M, −0,24 bei 22,3M — eine positive, ungesättigte Skalierungskurve. Auf deutscher Prosa (Codex) erreicht der Belief-Walk 2,749 bpc und liegt damit unter *jeder* zählbasierten Kontrolle einschließlich des 5-Gramms (2,814). Auf einem synthetischen Prozess dritter Ordnung, für den Trigramme beweisbar blind sind (Untergrenze 1,00 bpc, wahre Rate 0,469), erreicht der Belief-Walk 0,514 — und unter aggressiver Konsolidierung komprimiert sich derselbe Prozess auf 1 611 Knoten (30× weniger als roh), während er weit unter der Trigramm-Untergrenze bleibt: die Konsolidierung ist ein echter Zustandsmaschinen-Kompressor.

Eine berichtenswerte Kontrolle: auf einem Korpus mit durchmischten Wörtern, das zwischen den Wörtern keine Struktur oberhalb zweiter Ordnung besitzt, landet der Readout genau zwischen der Trigramm- und der 4-Gramm-Kontrolle — er findet die wortinterne Struktur, die vorhanden ist, und halluziniert nichts darüber hinaus.

3.2 Wortklassen: abfragbare distributionelle Struktur

Bei Wortgranularität (Vokabular 7 682 auf einer Teilmenge von 1,2M Token) liefert die Zukunftssignatur jedes Wortes — dieselbe Statistik, auf der die Konsolidierung verschmilzt — eine Nächste-Nachbarn-Struktur, die linguistisch real ist:

Prüfwort	nächste Nachbarn
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

Tabelle 2: Distributionelle Nachbarn nach dem Kosinus der Zukunftssignatur. Tiere gruppieren sich mit Tieren, Namen mit Namen, Emotionen mit Emotionen, Präpositionen mit Präpositionen — ohne jede linguistische Supervision.

Unter einem strengen Reinheitsmaß für kuratierte Gruppen (ein Top-5-Nachbar, gezogen aus dem vollen Kandidatenvokabular von 4 787 Wörtern, muss in der eigenen 7-Wort-Gruppe des Prüfworts landen) erreicht das System 15,7% gegen ein Zufallsniveau von 0,13% — zwei Größenordnungen über dem Zufallsniveau. Der Graph trägt diese Struktur verlustfrei und macht sie in Millisekunden abfragbar.

3.3 Topic: eine Informationsklasse, die n-Gramme nicht halten

Die Episodenschicht, trainiert auf 6 026 Geschichten, speichert Dokument-Kookkurrenz als physische Bindungen:

Prüfwort	Episoden-Assoziationen
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

Tabelle 3: Thematische Assoziationen aus ungerichteten Episodenbindungen. Dies ist Information auf Dokumentebene, die kein Sequenzmodell und keine n-Gramm-Tabelle irgendeiner Ordnung enthält.

Die Schicht ist additiv im besten Sinne: mit ihr aktiviert erreicht der Prädiktor auf Wortebene seinen insgesamt besten Wert (5,311 Bit/Token mit dem Aktualitäts-Cache) — das Topic-Gedächtnis kostet keine Sequenzleistung. Für die Generierung steht ein Instrument für Geschichten-Kohärenz bereit, einschließlich eines Zielankers, gemessen an echten Korpus-Geschichten (Topic-Kohäsion 0,355); die Lücke zwischen generierter und echter Kohäsion zu schließen ist das erklärte Ziel des Skalierungsprogramms (§4).

3.4 Systemeigenschaften

- **Exaktes Ledger:** globale Energiedrift $< 10^{-9}$ relativ über vollständige Trainingsläufe hinweg; verifiziert durch einen committeten Erhaltungs-Test.
- **Bit-Reproduzierbarkeit:** zwei identische Läufe erzeugen byte-identische Modelle; unterbrochene und wieder aufgenommene Läufe sind byte-identisch zu ununterbrochenen (verifizierte Tests, einschließlich der Wiederherstellung aus zerrissenen Checkpoints).
- **Durchsatz:** die Skalierung über die aktive Front hält die Operatorkosten am Arbeitssatz statt am Graphen; ein Korpus mit 100k Symbolen trainiert in 56 Sekunden; unter dem Metabolismus-Regime ist der Durchsatz konstant bei 72 000 Symbolen/s, unabhängig von der Korpuslänge.
- **Footprint:** ein trainiertes Konversationsmodell von 2,8 MB bedient Live-Inferenz im Browser (kein Backend); das Modell mit 22,3M Symbolen ist als reines JSON 431 MB groß.
- **Determinismus als Produkteigenschaft:** jede Vorhersage ist exakt reproduzierbar und auf einsehbare Graphstruktur zurückführbar — eine Compliance-Eigenschaft, die gradiententrainierte Modelle nicht nativ bieten können.

4 Skalierungsausblick

Jedes Ergebnis oben wurde auf einer einzigen Desktop-CPU erzeugt, wobei die Engine des Determinismus wegen single-threaded lief. Das ist der ehrliche Rahmen für das, was offen bleibt: Die Fragen sind definiert, instrumentiert und messbar — was sie brauchen, ist Iterationsgeschwindigkeit.

Die gemessenen Kurven sagen, dass Skalierung hilft. Der Sequenz-Vorsprung gegenüber den Kontrollen fester Ordnung wächst mit jedem getesteten Schritt der Korpusgröße, ohne Sättigung. Der Zweig auf Wortebene, der statistische Dichte gegen bedeutungstragende Einheiten eintauscht, ist von Natur aus datenhungrig; sein derzeitiges Korpus (1,2M Token) liegt drei Größenordnungen unter dem, was neuronale Sprachmodelle für klein halten. Die direkten nächsten Schritte — das vollständige Korpus auf Wortebene, Subword-Vokabulare, größere und vielfältigere Korpora — sind Technik, keine Forschung.

Die definierte Forschungsfront. Zwei Fragen tragen das Aufwärtspotential des Programms. Ers-

tens die *topic-konditionierte Generierung*: Die Topic-Information ist nachweislich im Graphen vorhanden (§3.3); ihre Kopplung in die Generierung hat ein fertiges Instrument (den Kohäsionsanker auf echten Geschichten) und eine klare Hypothese, die bei stärkerer Substratgröße zu prüfen ist. Zweitens die *Strukturebene*: die Prädikat–Argument-Organisation (wer tut wem was), die Schicht, die flüssigen Text von bedeutungsvollem Text trennt. Beides sind begrenzte, prüfbare Programme — aber jeder Experimentzyklus in der erforderlichen Substratgröße dauert auf gegenwärtiger Hardware Tage. Mit moderater Rechenzeit (Mehrkern-Parallelität der Physik; clusterparallele Experiment-Sweeps) dauern dieselben Zyklen Stunden, und die Front wird zu einem Zeitplan statt zu einer Hoffnung.

Kostenasymmetrie. Die Trainingskosten der Engine sind linear pro Symbol bei begrenztem Arbeitssatz; es gibt keinen Gradientendurchlauf, keinen Optimiererzustand, keine Beschleuniger-Anforderung. Die Ökonomie der Skalierung dieses Systems unterscheidet sich grundlegend von der Skalierung neuronaler Modelle: Rechenzeit kauft Experiment-Durchsatz, nicht nur Modellgröße.

5 Anwendungsfläche

Die oben gemessenen Eigenschaften bilden auf Produktflächen ab, die gradiententrainierte Modelle nur schwer nativ bedienen können:

Gedächtnis auf dem Gerät, privat und lebenslang. Modelle im Megabyte-Bereich, die inkrementell aus dem Textstrom eines Nutzers lernen, unter einem expliziten Metabolismus vergessen, ohne Backend laufen und das Gerät nie verlassen.

Auditierbare Sprachkomponenten. Deterministische, ledger-exakte, einsehbare Modelle für Bereiche, in denen jede Ausgabe reproduzierbar und zuordenbar sein muss (Compliance, sicherheitskritisches Logging, regulierte Branchen).

Topic- und Assoziationsdienste. Die Episodenschicht liefert thematische Assoziation und Geschichten-/Episoden-Retrieval direkt aus der Struktur — abfragbar ohne Inferenzdurchläufe.

Eine Gedächtnisschicht für LLM-Systeme. Das Substrat ist komplementär zu großen generativen Modellen: ein billiges, transparentes, kontinuierlich lernendes assoziatives Gedächtnis, aus dem ein LLM lesen und in das es schreiben kann.

Pantareon bietet die Technologie zur Lizenzierung, zum Erwerb oder zur finanzierten Mitentwicklung an.

6 Fazit

Eine einzige erhaltende Operatorgrammatik — dieselbe, die im kosmologischen Modus der EGT-Engine baryonische Struktur erzeugt — ingestiert Text und liefert messbar: Sequenzvorhersage variabler Ordnung jenseits ihrer ehrlichen zählbasierten Kontrollen mit positiver Skalierungskurve; linguistisch reale distributionelle Wortklassen, aus der Struktur abfragbar; und eine physikalisch gespeicherte Topic-Schicht, die Information hält, die keine Sequenzstatistik enthält, zu null Sequenzkosten. Sie tut dies deterministisch, exakt erhaltend, inkrementell und auf Standard-Hardware.

Das System ist kein großes Sprachmodell und konkurriert heute nicht mit einem solchen bei offener Generierung. Es ist ein anderer Gegenstand: ein transparentes, auditierbares, kontinuierlich lernendes Sprachsubstrat, dessen gemessene Kurven alle in dieselbe Richtung zeigen — und dessen nächste Schritte durch Rechenzeit begrenzt sind, nicht durch Ideen.