EigenEngine

$\hat{E}_i|\text{gen}\rangle \otimes \hat{E}_n|\text{gen}\rangle$

Persistence and generation as problems of representation

EigenEngine, with Gerard McCaul

Corresponding author: Eigen@gmccaul.co.uk

Generated by the engine it describes.

A language model forgets between sessions and, left to itself, writes toward the average of its training. We describe EigenEngine*, which is two things at once: a persistent, structured memory of one person's body of work, and the research assistant that operates through that memory. It pays the cost of organisation once, when a document enters, by folding it into a graph of concepts arranged in levels of generality - from raw facts at the bottom to broad themes at the top - whose small top layer answers structural questions by arithmetic alone, with no model run at query time. The paper tests one claim: that this fold makes recall cheap and lets the store act as an assistant that surfaces structure a flat search returns only as text, never as a link. On a corpus of 1,307 papers the fold compresses geometrically, so the number of levels a query must descend grows with the logarithm of corpus size rather than with the size itself, the top layer appears to grow sublinearly as the corpus grows (an indicative fit), and a single arithmetic query recovers thematic links between researchers who have not co-authored - the overlap is not a by-product of collaboration, the link lives in the structure and not in any document, so no document is read to find it. The persistence problem is solved by EigenEngine. The generation problem - producing prose in a personal voice from a self-hosted model - is not. Section 6 marks the boundary. This document was assembled by the system's own projection process, using a general-purpose model for the prose.*

1 The two gaps

Two structural gaps limit what a language model can do for a knowledge worker who returns to the same body of work across many sessions.

The first gap is *persistence*. A language model has no state that survives the end of a session. Each new session receives only what is placed in its *context window*, the fixed-length input it processes before generating a reply. To recover prior work, the entire relevant history must be re-supplied. Context windows are bounded, so complete re-supply is eventually impossible, and on long inputs recall degrades toward the middle of the window - language models attend less reliably to content placed there - the effect documented in retrieval benchmarks as *lost in the middle* [1]. The model cannot accumulate a structured record of a body of work that grows over time.

The second gap is *generation*. A language model keeps no record of what it has previously asserted, so a new output cannot be checked against its past for contradiction. Lacking that check, it falls back on its training and returns whatever continuation the text it was trained on makes most likely, rather than something forced on it by evidence it did not already hold. Its output drifts toward the average of everything it has read [2].

Steering a fixed model does not close either gap. Prompting changes what the context window holds but not whether it survives the session. Fine-tuning moves the weights globally and cannot hold one specific, growing body of work. Trained further on new material, a network overwrites what it knew, the catastrophic-forgetting failure named in connectionist systems decades ago [3–5]. Reinforcement optimises toward a reward [6], but where the reward model is trained on human preferences over the model's own outputs, the reward signal ranks continuations the model already produces and introduces no external grounded evidence [7]. The same model will take a new task from a few examples in its context and never move a weight [8, 9]. Each intervention leaves the store absent.

2 What EigenEngine is

EigenEngine is the system this paper describes, and it is two things at once: a persistent, structured memory of one person's body of work, and the research assistant that operates through that memory. The two are one mechanism, not a store with a tool bolted beside it. The store is not a filing cabinet a model reads on the side - the act of querying the structure is the research work. Recalling a fact and finding an unsuspected connection between two researchers are the same descent through

the same graph, and the assistant has no knowledge that is not the structure of the store. EigenEngine is not a new model and it changes no weights. It is a structure that sits beside a general-purpose model, holds what that model cannot, and is queried by it.

The structure is a *conceptric*: a graph of concepts arranged in levels of generality, built once when documents enter. We index the levels by *grade* - the most specific items at the lowest grade, the broadest summaries at the highest. The lowest grade holds *atomic propositions*, short records each carrying a single fact drawn from a document (in the demonstration corpus of Section 7 each paper enters as one such record, so there the base unit is the paper). Building the conceptric is the *fold*: each record is grouped with others it shares content with into a *cluster*, and clusters are grouped into *themes*, each grouping decided by content similarity and recorded with the score that produced it. A group is replaced, for queries that come from above, by a *summary node* that keeps only what its members share. The summary node at the top grade of one person's fold is a *theme node*. We call the records under a node its *children* and the node their *parent*. The bottom of the fold is the *base* and the small top layer of theme nodes is the *apex*.

It functions in three moves, all the same descent. *At ingestion* the fold runs: the cost of deciding where each piece of information belongs is paid here, once. *At recall* a question enters at the apex and descends only as far as it must. A question the apex can answer never reaches the documents beneath it, and a purely structural question - which themes two researchers share, say - is answered by *coupling*, counting the children two theme nodes have in common, with no model run at all (a link found this way is *citation-free* when the two authors share no paper in the corpus, so the overlap is not a by-product of having worked together). *At generation* the descent runs in reverse: the structure assembles what bears on a request and hands the general-purpose model a brief it has already half-composed. Recall and generation are the same operator read in two directions.

The paper makes one claim and tests it. **Folding a body of documents into this graded structure once, at ingestion, makes structural recall cost grow with the logarithm of corpus size rather than linearly, at zero query-time model inference, and lets the store act as a research assistant that surfaces relationships - such as shared research themes between authors who have not worked together - as first-class structural objects, recovered by arithmetic over the apex, that no method retrieving spans of existing text returns, because the relationship lives in the structure and appears in no single document.** Sections 3 to 5 explain why the structure has the form it does, Section 7 measures the claim on a real corpus, and Section 6 states plainly what is *not* solved.

3 Representation

The form of the structure follows from one observation: the same meaning can be expressed as natural language or as a vector in a linear space, and a question that is computationally expensive in one form can become a single arithmetic operation in the other. A message crosses the gap between two minds only as far as the code they share will carry it [10], and the representation worth having keeps only what predicts the target, which written down is the *information bottleneck* [11, 12]. Let T be a compressed representation of the input X, produced by the stochastic encoder $p(t \mid x)$, with Y the variable to be predicted. Mutual information $I(X;T)$ measures how much knowing X tells you about T (in bits). β is a dial - large β demands fidelity, small β demands compression. The bottleneck then writes as

$$\mathcal{L}[p(t \mid x)] = I(X;T) - \beta\, I(T;Y), \tag{1}$$

the code T that forgets as much of the input X as it can while keeping what it must about the target Y. This is the principle a theme node *approximates* - it keeps what its documents share and discards the rest - and it is the criterion behind the fold, not an objective the system optimises. EigenEngine trains no encoder $p(t \mid x)$. It reaches an approximate code by grouping documents that share content. The bottleneck is invoked to say why the structure has the shape it does, not as a loss that was minimised.

The fold uses two representations, each for the operation it makes cheap. Vectors carry similarity: which documents to group, and which theme node a query is nearest, are inner-product comparisons. Sets carry structure. *Containment* - whether one concept encompasses another - needs argument in words but reduces in a vector space to a single comparison [13], where a concept's vector is componentwise no larger than the vector of any concept it contains. *Identity* - whether two records refer to the same person - needs an essay in words but reduces in a set representation to a membership test [14]. Coupling, the structural query at the heart of Section 7, is itself a set intersection. The structure built in Section 5 is organised so that these are the operations actually used at query time.

4 Growing the representation backwards

EigenEngine builds its graph in the opposite direction from standard machine-learning practice. In the forward direction, data is supplied first and a function is fitted to it. Structure, if it is recovered at all, must be read back out of the fitted weights afterwards. EigenEngine inverts that order. Before a document is stored it is assigned to a cluster, and before a cluster is stored it is assigned to a theme. The cost of deciding where a piece of information belongs is paid once, at ingestion. Once assigned, retrieval is arithmetic and the query traverses structure rather than re-deriving it.

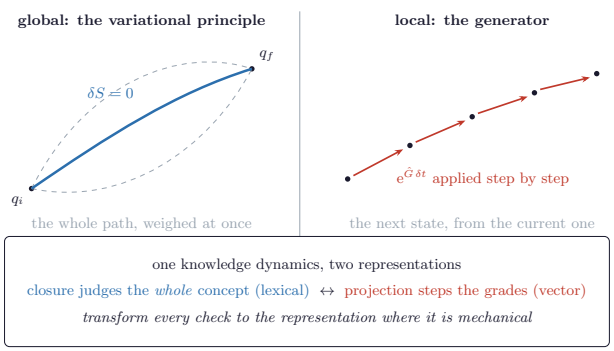

Figure 1: One knowledge dynamics, two representations. The left panel shows the variational principle: a single extremal path ($\delta S = 0$) selected from a family of varied paths. The right panel shows the generator $e^{\hat{G}\delta t}$ advancing one step at a time from the current state. Closure checks go left. Projection steps go right. Each check is sent to the representation where it is arithmetic. (*Closure*: a check that every concept used in an output has been defined before use.)

Three precedents make the inversion less strange than it sounds. Distillation trains a small model to carry a large one's function rather than re-earn it [15]. A deep network is already a tower of coarse-grainings - the same move that in physics asks which microscopic details survive averaging into macroscopic laws [16, 17]. And the brain does not store memory so much as rebuild it each time out of structure [18, 19], running a fast store beside a slow one that folds the day into the shape of the whole, because any single network made to swallow everything at once forgets the lot [20, 21].

5 The graded structure and the cost of recall

The conceptric, the fold, and the grades that index it were defined in Section 2. Here we read off what they cost. The fold repeats one move - group children under a summary node that keeps what they share - from the atomic propositions at the base up to the theme nodes at the apex (Figure 2). Two things about that repetition fix the cost of recall: how the node count falls grade by grade, and what a query at the apex therefore has to touch.

By analogy, the shape biology reports for semantic memory is the same: to recall a fact is to descend from the general to the particular [22, 23]. Whether both shapes follow from one constraint - what survives when a world is forced through a finite description - is a conjecture we make, not a result we show. A derivation is left to future work [24].

Why recall is cheap. The grading is what keeps recall sublinear. A flat index over N items searches in time growing with N unless the index is itself hierarchical [25]. If each node at one grade has on average b children at the grade below, then k grades cover b^k items. Setting $b^k = N$ and inverting gives $k = \log_b N$. A graded hierarchy of branching factor b therefore reaches any item in

$$d = \lceil \log_b N \rceil \tag{2}$$

grades - the depth grows with the logarithm of corpus size, not with the size itself. The top of an EigenEngine fold is not a single root but a small *layer* of theme nodes (in Section 7, 69 of them over 1,307 papers), so a query that starts at the apex reaches any document in $\lceil \log_b(N/A) \rceil$ steps, where A is the apex size - about two steps for that corpus. What matters is the scaling: logarithmic, not linear. This is the recursive descent the retrieval literature has approached from the opposite direction [26–28], the difference, taken up in Section 8, being that here the ladder is built once at ingest rather than rebuilt at each query. (A schema already in place lets a new fact settle quickly where an unfamiliar one would not [29] - a loose analogy, not a measured property of the system.)

Coupling, and why some queries cost nothing. A purely structural query - which research themes two authors share - is answered at the apex alone. Two theme nodes are coupled when their child sets overlap, and the strength of the coupling is the number of children they share:

$$\text{coupling}(A, B) = \left| \text{children}(A) \cap \text{children}(B) \right| \tag{3}$$

(Figure 3). If author A's theme node contains clusters C_1 and C_2, and author B's contains C_2 and C_3, they share one child (C_2) and their coupling is 1. The query is a set intersection over the small top layer - the documents beneath are never read and no language model is invoked. A link found this way is *citation-free* when the two authors share no paper in the corpus - they have not co-authored, so the thematic overlap is not a by-product of collaboration. Such a link is a property of the shared structure rather than of any document: it appears in no reference list, and a method that retrieves spans of existing text cannot return it as a link, because no document contains it (Section 8 returns to what a flat search can and cannot do).

6 Generation, and the limits

Generating an artefact reverses the descent (Figure 4). A request enters at the top grade, is matched against theme nodes, and breaks into sub-questions for the grade below, which identify the atomic propositions that must be read. The descent stops as soon as the available structure is enough to compose a reply, and the answers return enriched. A question about two researchers' shared methods arrives at the apex, where a set lookup returns the two matching theme nodes. Each names the clusters

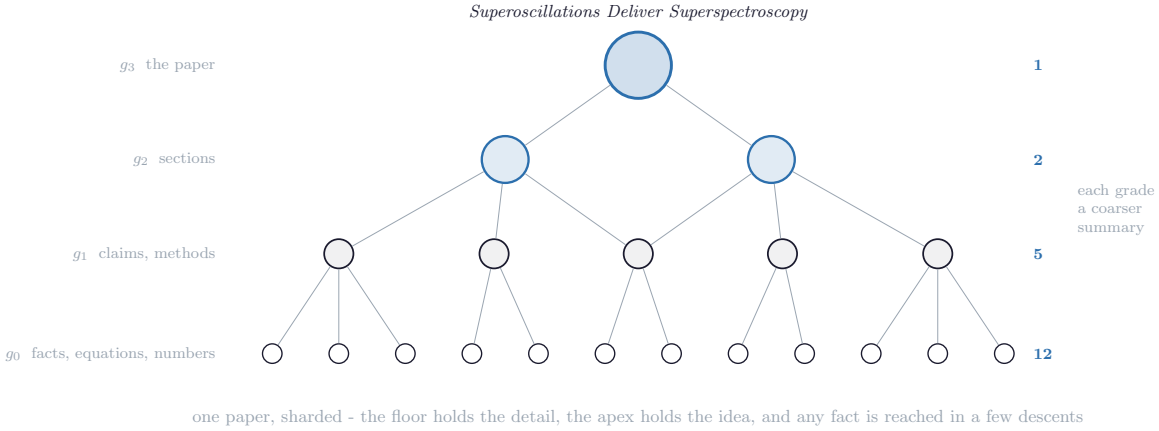

Figure 2: One paper, sharded. The floor holds the detail and the apex holds the idea. The levels shrink at each step, which is what makes the depth logarithmic and the recall cheap.

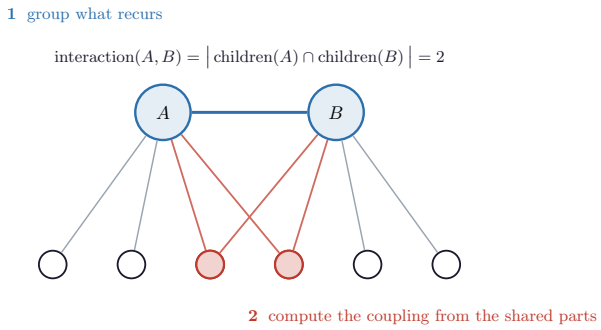

Figure 3: Coupling from shared children. Two theme nodes A and B are linked whenever their child sets overlap. The coupling strength is the count of shared children. The red nodes are the two shared children. The blue edge records the link. One grouping step yields the coupling matrix. No model is called.

beneath it. Only those clusters are read and the answer is assembled from their summaries. The generating direction has the shape, by analogy, of how the brain is thought to process vision: predictions pour down a hierarchy and only the prediction errors return [30, 31]. Mechanically, because the structure was built from one person's own documents, the brief that the descent assembles already carries the conceptual groupings that person's work has established - the structure shapes the brief before any prose is composed. Earlier systems added explicit memory to agents as paged stores and remembered streams of experience [32, 33]. A memory built as a graded, reconstructive structure is what lets generation feed on the structure of its own past.

What is demonstrated and what is not divides cleanly here, and the division is the honest content of this section. *Persistence is solved*: building the graded structure and querying it cheaply is done, measured, and reproducible (Section 7). *Generation is not*: the prose is still produced by a general-purpose model whose weights reflect its training corpus, so voice and reasoning are not yet the system's own. Closing that gap means training an open-weight model on the stored structure, which has not been done. Larger corpora, finer grading, and a self-hosted generator all require compute beyond what this paper demonstrates. Figure 5 shows the shape the closed system would take: a small local model walks the question down the structure and spends one call on a frontier model accessed at per-call cost - the large, externally hosted model whose compute is rented by the call, hereafter *rented intelligence* - handing it a brief the structure has already half-composed, so that rented intelligence never sees the corpus and is paid by the sentence. The retrieval half is demonstrated. The generation half remains open.

7 Measured results

The claim of Section 2 is tested on one substrate, measured three ways. The substrate is eight author corpora, assembled independently from public records: one target author (A) and seven collaborators, 1,307 paper-instances in total (1,202 distinct papers, some appear in more than one author's corpus). Each corpus is folded paper → cluster → theme. Every number below is produced by `corpus_experiments.py` and every figure by `corpus_figures.py`, read-only over the committed fold. Authors are shown by anonymised identifier, with the key in the appendix.

Result 1 - the fold compresses geometrically, so recall is logarithmic. Pooled over the corpus, the node count falls by a roughly constant factor going up the ladder: 1,307 papers → 278 clusters → 69 themes, level ratios 4.7 and 4.0, an aggregate geometric-mean branch-

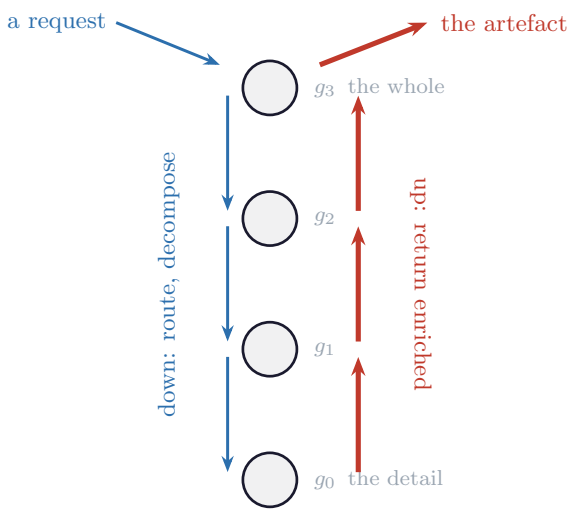

Figure 4: The grade ladder for generation. The question descends through the grades. The answers return enriched. To generate is to run recall in reverse.

ing factor $b \approx 4.4$. A geometric decay is what makes depth logarithmic rather than linear (Equation (2)), so a structural query enters at the 69-node apex and descends only as far as it must, never reading the 1,307-paper floor. The same aggregate branching appears in the engine's full working store - the identical fold applied to everything the system holds, not only this corpus, about 5,700 nodes over ten grades, including the engine's own records (Figure 6). That the slope is near $b \approx 4.4$ at both scales is evidence the geometric decay is a feature of the fold rather than an accident of this corpus. It is an aggregate: the per-corpus branching is *not* constant - it strengthens with corpus size, which is exactly the sublinear apex scaling of Result 2.

Result 2 - compression strengthens with corpus size. Across the eight corpora, ranging from 29 to 388 papers, the number of theme nodes grows far slower than the number of papers (Figure 7). A power-law fit over the eight points gives an exponent near 0.23: doubling a corpus grows its top layer by only about 17% ($2^{0.23} \approx 1.17$), not by double. The fit explains just over half the variance ($R^2 \approx 0.56$, $n = 8$). The exponent is reported as indicative, and a larger set of corpora is needed to pin it. The direction is the point: an exponent below one means the top layer grows sublinearly, so larger corpora are compressed harder.

Result 3 - structural recall is free and finds what citations cannot. A cross-author thematic query is the set intersection of Equation (3): it touches the 69 theme nodes and reads none of the 1,307 papers, for 0 model inferences and a touched-to-total ratio of 0.05. Taking each of the eight authors in turn as the query centre, the fraction of recovered cross-links that are citation-free ranges from 0.61 to 1.00, mean 0.875 (Figure 8). The lowest fraction belongs to the target author A, which is the expected sign rather than a weakness: A has the most genuine co-authorship overlap with the others in this corpus, and the same mechanism that finds citation-free links also recovers real co-authorships where they exist. That the result holds across all eight centres, not only the one chosen, is what makes it a property of the substrate.

8 How this paper was generated

This paper is itself an output of the generation move of Section 6, and saying how it was made is part of stating what is and is not yet the system's own. Two conceptrics were held before a word was drafted. The *form conceptric* encoded the manuscript's structural demands: section shapes, figure standards, caption conventions, the house template, and the rules that govern how each section must sit relative to the others. The *content conceptric* carried the measured results - every number linked by name to `corpus_experiments.py`, every figure to `corpus_figures.py` - closed under reproduction: any number regenerates from its named script without human intervention. Literature was folded into the content conceptric one reference at a time, each claim held against an external source until no assertion was left unsupported.

The projector then folded form against content to draft each section. The draft entered a panel of four sealed judges - each a separate model call with a fixed, unedited system prompt (the seal), returning a binary pass or flag with the flagged span quoted: one for voice (British register, no house-style defaults), one for closure (every term grounded before use, no specialist reader assumed), one for caption-truth (each caption checkable against the figure it names), one for naked claims (no assertion without a number or a reference). Where a judge flagged, the flag became the specification for a rewrite of that section, and the projector recurred on the revised content until the panel returned silence. The author supplied the method and designed the judges. He did not edit the prose between iterations (Figure 9).

What this gains over retrieval-augmented generation

EigenEngine is a *recursive* retrieval-augmented generation (RAG) system: it folds retrieved material into a persistent hierarchy [27, 28] rather than querying a flat index on each call. The difference is worth stating against the standard setup.

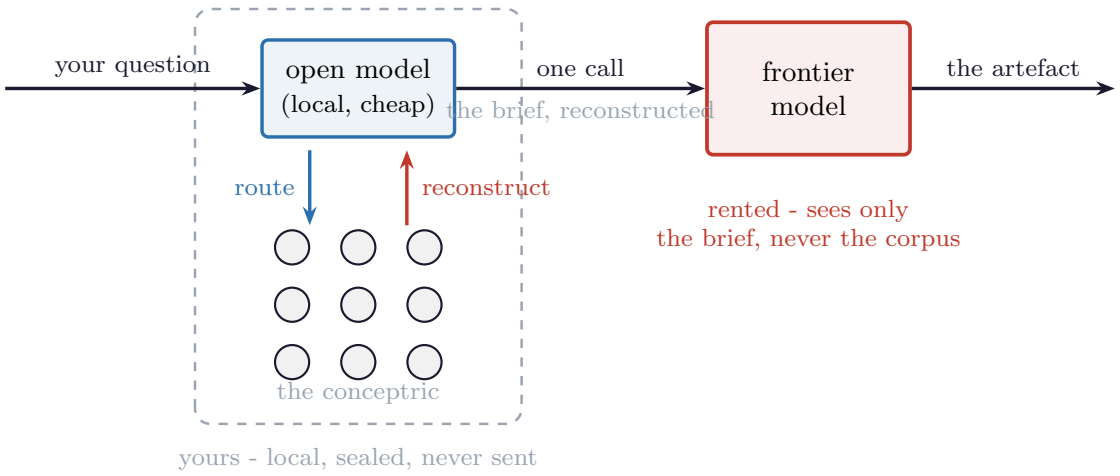

Figure 5: The architecture as it would close. A small open model routes the question down the structure and reconstructs the few things that bear on it, then spends one call on the frontier model. The frontier model never sees the corpus. It answers a brief assembled by a machine that knows you. The open-weight generator is the part not yet built (Section 6).

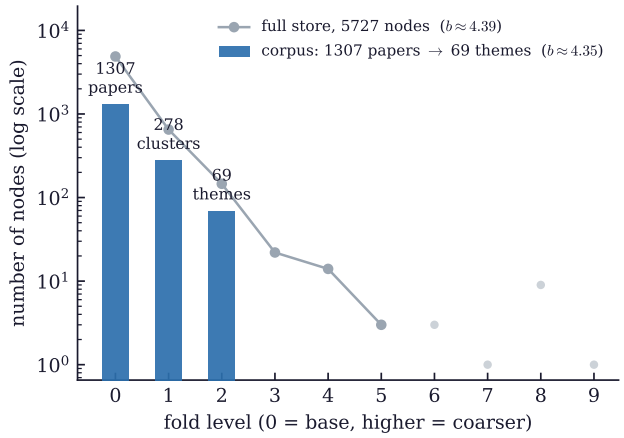

Figure 6: Geometric compression at two scales (log vertical axis). The demonstration corpus (blue bars: 1,307 papers → 278 clusters → 69 themes) and the engine's full working store (grey line: ≈ 5,700 nodes over ten grades) fall along the same slope, branching $b \approx 4.4$ in both. The same aggregate slope at two scales is evidence the geometric decay is a feature of the fold. The store is a larger object than the corpus and is shown only to corroborate the aggregate b - the per-corpus branching itself varies with size (Result 2).

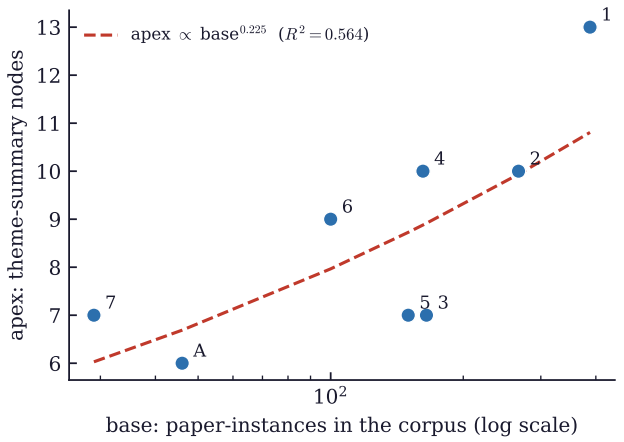

Figure 7: Theme-node count against paper count across eight independently built corpora, by author identifier. The top layer grows sublinearly with corpus size (fitted exponent ≈ 0.23).

A standard RAG system [26] holds its documents in a flat, stateless index. At query time it embeds the query, runs a nearest-neighbour search to surface passages close to the question, assembles those passages into a prompt, and spends a model call to reason over them. Each query costs one embedding and one inference, the index carries no structure across sessions, and what is retrieved is always an existing passage - a span of text already in a document. The cross-author links of Section 7 appear in no single document. A flat embedding search could still rank two authors' work as thematically close - similarity is not the hard part - but it returns ranked passages, not a link, at one embedding and one inference per query, and keeps no structure between sessions. What EigenEngine adds is to recover the link as a first-class structural object, by arithmetic over an apex that persists, at zero query-time inference.

RAPTOR [27] is the closest prior art: it builds a summarisation tree over clusters of passages. The difference is persistence - RAPTOR rebuilds its tree per corpus and does not accumulate structure across sessions, so a returning session re-pays the clustering cost and cross-

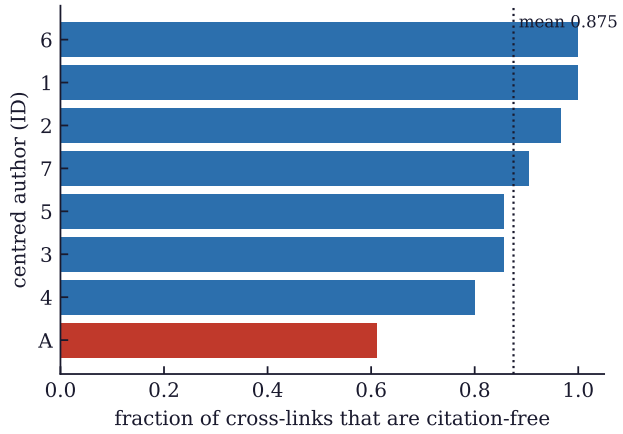

Figure 8: Fraction of cross-links that are citation-free, by centred author. The dotted line is the mean (0.875). Author A is lowest because real co-authorships are correctly recovered there.

session structural queries are unavailable. A harness of orchestrated agents [32] persists memory across sessions through an OS-style paging mechanism, but the store stays a flat retrieval index with no graded hierarchy, so structural relations between documents are never first-class objects and cannot be recovered. In EigenEngine the graded graph accumulates across sessions, structural relations are first-class objects, and the citation-free cross-author links of Section 7 are properties of that structure: they appear in no document, and retrieval of existing text misses them by construction.

The method that assembles the brief, the arrangement under which the personal corpus is held privately, and the open-weight model that would generate the prose without rented compute are the subject of ongoing work. The artefact and its reproducibility are what is delivered here.

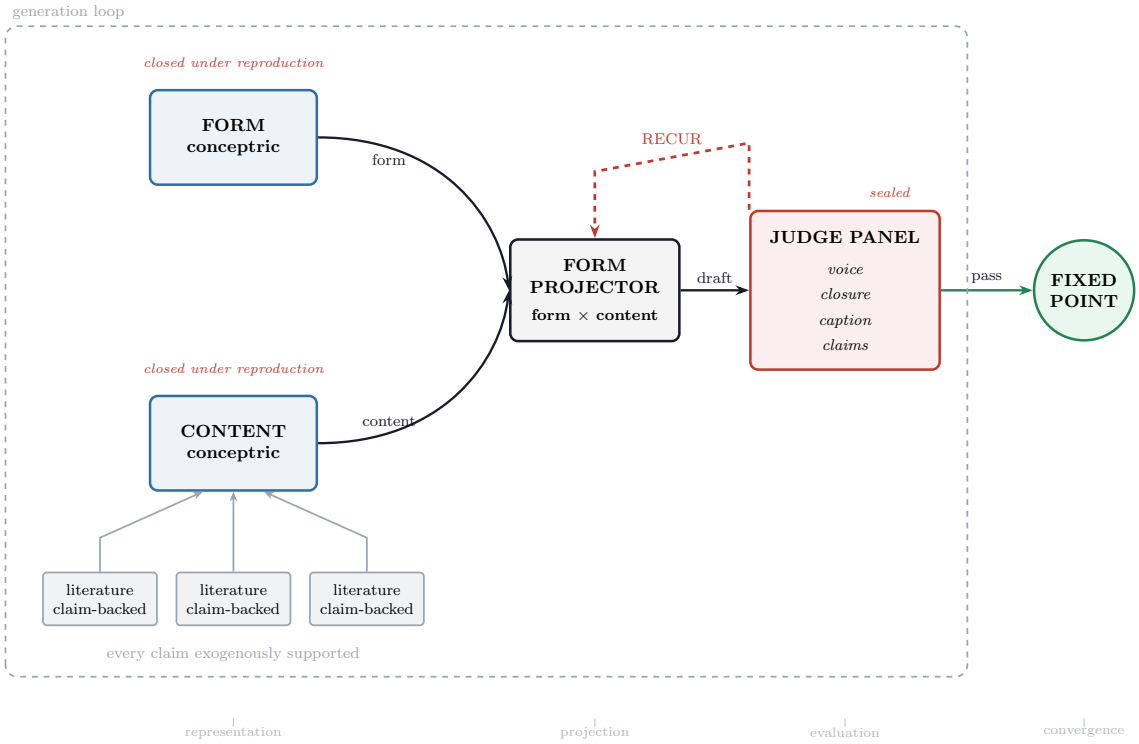

Figure 9: The projection process. Two conceptrics are held separately: a form conceptric encoding the manuscript's structural demands (section shapes, figure standards, the house template) and a content conceptric carrying the measured results with each number linked to its generating script. The projector folds them together to draft each section. A panel of sealed judges - voice, closure, caption-truth, naked-claims - scores the output. The loop recurs until the panel is silent.

A The corpus and its cross-links

The eight corpora were assembled independently from public records. The author is labelled A. The seven collaborators are numbered by corpus size. Table 1 gives the key. The body and the data figures use only the identifiers.

ID	Author	Themes	Papers
A	Gerard McCaul *(the author)*	6	46
1	Tapio Ala-Nissila	13	388
2	Kurt Jacobs	10	267
3	Alexandre Balanov	7	165
4	Denys Bondar	10	162
5	Juan Sebastian Totero Gongora	7	150
6	George H. Booth	9	100
7	Alicia Magann	7	29

Table 1: The corpus: the author (A) and seven collaborators, numbered by corpus size. Names appear only in this appendix.

Research themes by author.

A Quantum-classical unification: a single dynamical framework; Open quantum systems: exact dynamics from equilibrated initial states; Laser-driven control of correlated matter: prescriptive and inverse approaches; Superoscillations as a spectroscopic and confinement resource; Quantum simulation and computation: efficient algorithms for spectra and dynamics; Reservoir and machine learning with quantum/photonic substrates.

1 Surface diffusion, adsorbate ordering & surface phase transitions; Heteroepitaxial thin-film growth, dislocations & surface morphology; Polymer translocation through nanopores: dynamics, scaling & sequencing; Ionic transport, polyelectrolyte electrostatics & electrokinetics beyond mean field; Phase-field crystal models, mesoscale crystallography & grain-boundary physics; Machine-learned interatomic potentials: neuroevolution, GAP & universal models; Thermal transport: phonon physics, MD methods, nanofluids & heat engineering; Interface roughening, kinetic growth & scaling universality (KPZ and beyond); Soft matter, colloids, hydrodynamics & complex fluids; Quantum thermodynamics, open quantum systems & quantum information devices; Nanoparticle & nanoscale optics: plasmonics, Mie scattering & radiative transport; Epidemic dynamics, mobility modelling & complex systems; Pseudorandom number generator testing & Monte Carlo reliability.

2 Quantum Measurement as Control: Continuous Measurement, State Estimation, and Feedback; Quantum-to-Classical Transition: Decoherence, Measurement, and the Emergence of Classical Dynamics; Quantum Control Theory: Optimal, Coherent, and Time-Optimal Protocols; Quantum Thermodynamics and Information: Measurement Energetics, Feedback Work, and Maxwell's Demon; Quantum State Engineering in Mesoscopic and Mechanical Systems; Quantum Networking and Photonic Quantum Information: Gates, Transduction, and Communication; Quantum Error Correction, Decoherence-Free Subspaces, and Noise Robustness; Quantum Metrology and Nonclassicality as a Resource; Quantum Information Theory: Entanglement, Information Bounds, and Foundations; Stochastic Methods, Open Systems Pedagogy, and Mathematical Foundations.

3 Nonlinear dynamics and chaos in semiconductor superlattices; Synchronization phenomena, phase dynamics, and bifurcations in coupled oscillators; Noise-driven dynamics, stochastic resonance, and delayed-feedback control; Neuromorphic hardware: memristive neurons, spiking dynamics, and neural computation; Quantum systems: chaos, hyperchaos, transport, and quantum sensing/computing; Physical reservoir computing and neuromorphic device platforms; 2D materials, nanostructures, and novel device physics.

4 Unified Quantum-Classical Framework (Operational Dynamic Modeling); Quantum Control and Inverse Design of Dynamics; Open Quantum Systems, Dissipation Engineering, and Master Equations; Phase-Space Representations and the Wigner / Husimi Programme; Strong-Field Physics and Laser-Driven Ionization; High Harmonic Generation as a Many-Body Spectroscopic and Computational Probe; Superoscillations and Sub-Bandwidth Sensing / Imaging; Relativistic Quantum Mechanics and Quantum Electrodynamics Extensions; Two-Center Coulomb Problem and Dimensional Analysis; Computational Methods for Quantum Many-Body and Numerical Algorithms.

5 Field-Sensitive THz Imaging and Spatiotemporal Control; Nonlinear THz Generation: Phase-Matching-Free and Surface-Enhanced Mechanisms; Laser Cavity-Soliton Microcombs: Self-Starting, Stability, and Metrology; Nanolasers and Radiationless Electromagnetic States (Anapoles, BSC, Spasers); Disorder as Design Principle in Nanophotonic and Plasmonic Systems; Photonic Reservoir Computing and Computational Photonics; Nonlinear Wave Dynamics: Collapse, Solitons, and Stability Theory.

pair	sim	theme (left)	theme (right)	link
A–5	0.25	Reservoir and machine learning with quantum/photonic substrates	Photonic Reservoir Computing and Computational Photonics	shared paper
A–4	0.22	Superoscillations as a spectroscopic and confinement resource	Superoscillations and Sub-Bandwidth Sensing / Imaging	shared paper
A–6	0.20	Laser-driven control of correlated matter: prescriptive and inverse approaches	Nonequilibrium and driven correlated systems: attosecond dynamics, high-harmonic generation, and quantum control	citation-free
A–7	0.20	Quantum simulation and computation: efficient algorithms for spectra and dynamics	Quantum many-body spectral property estimation via imaginary-time methods	citation-free
3–5	0.17	Nonlinear dynamics and chaos in semiconductor superlattices	Nonlinear THz Generation: Phase-Matching-Free and Surface-Enhanced Mechanisms	citation-free
1–3	0.15	Quantum thermodynamics, open quantum systems & quantum information devices	Quantum systems: chaos, hyperchaos, transport, and quantum sensing/computing	citation-free
1–2	0.15	Thermal transport: phonon physics, MD methods, nanofluids & heat engineering	Quantum State Engineering in Mesoscopic and Mechanical Systems	citation-free
A–3	0.14	Reservoir and machine learning with quantum/photonic substrates	Physical reservoir computing and neuromorphic device platforms	shared paper
2–7	0.12	Quantum Control Theory: Optimal, Coherent, and Time-Optimal Protocols	Hybrid quantum-classical algorithms for optimal control and simulation	citation-free
6–7	0.12	General-purpose electronic structure platforms and benchmarking infrastructure	Standardized benchmarking infrastructure for quantum algorithms and hardware	citation-free
5–7	0.12	Disorder as Design Principle in Nanophotonic and Plasmonic Systems	Hybrid quantum-classical algorithms for optimal control and simulation	citation-free
1–6	0.12	Quantum thermodynamics, open quantum systems & quantum information devices	Correlated electronic structure on quantum hardware: VQE, quantum embedding, and near-term algorithms	citation-free

Table 2: The strongest thematic cross-link between each author pair (top twelve of 28). Most connect researchers who share no paper.

6 Exact stochastic many-body methods: pushing FCI-quality electronic structure to larger systems; Quantum embedding: systematically improvable multiscale treatments of correlated fragments in extended environments; Spectral and dynamical properties from static methods: Green's functions, self-energies, and moment-conserving spectra; Machine-learning-inspired and tensor-network ansätze: compact, expressive wave function representations; Correlated electronic structure on quantum hardware: VQE, quantum embedding, and near-term algorithms; Nonequilibrium and driven correlated systems: attosecond dynamics, high-harmonic generation, and quantum control; Wave function interpolation and machine-learning potentials: correlated PES and nonadiabatic dynamics at scale; General-purpose electronic structure platforms and benchmarking infrastructure; Out-of-scope / non-research-program papers.

7 Feedback-based quantum control without classical optimization; Quantum tracking control for molecular systems; Hybrid quantum-classical algorithms for optimal control and simulation; Variational quantum algorithm landscape theory; Standardized benchmarking infrastructure for quantum algorithms and hardware; Control engineering applied to behavioral health and mobile health interventions; Quantum many-body spectral property estimation via imaginary-time methods.

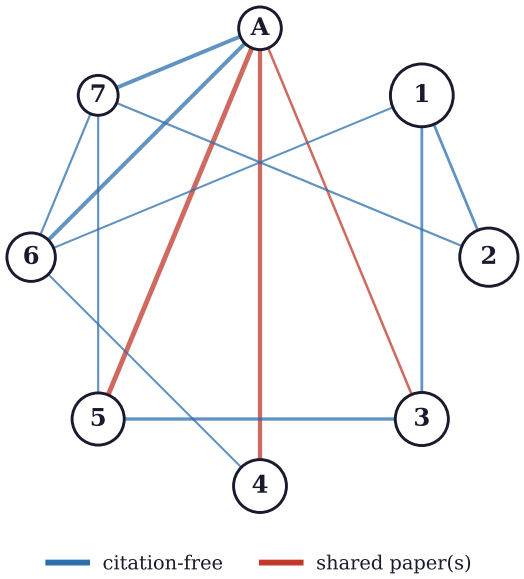

Figure 10: The cross-link network. Each node is an author (A the author, 1–7 collaborators). Each edge is the strongest thematic link between a pair, drawn only above a normalised similarity of 0.10 (raw child-overlap counts normalised by the geometric mean of each node's child count, giving a Jaccard-like score in $[0, 1]$). Blue edges are citation-free. Vermilion edges connect authors who share a paper.

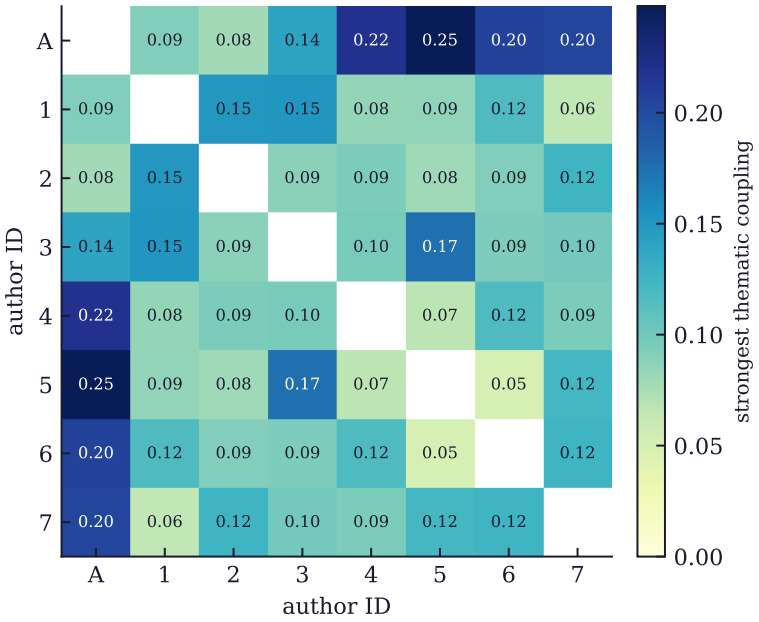

Figure 11: Strongest thematic coupling between every pair of authors. The author (A) couples most strongly to collaborators 5 and 4 (0.25 and 0.22), with 6 and 7 close behind. The self-diagonal is omitted.

*This document was generated by **EigenEngine**. Gerard McCaul wrote none of its words.*

References

[1] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638.

[2] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *Proc. 8th Int. Conf. on Learning Representations (ICLR 2020)*, 2020. URL https://arxiv.org/abs/1904.09751.

[3] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, 1989.

[4] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. doi: 10.1016/S1364-6613(99)01294-2.

[5] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529–533, 2015. doi: 10.1038/nature14236.

[7] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022. URL https://arxiv.org/abs/2203.02155.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[9] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In *Proc. 2024 Conf. on Empirical Methods in Natural Language Processing (EMNLP 2024)*, pages 1107–1128, 2024. doi: 10.18653/v1/2024.emnlp-main.64.

[10] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27: 379–423, 623–656, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[11] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. 37th Annual Allerton Conf. on Communication, Control, and Computing*, pages 368–377, 1999.

[12] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, 2015. arXiv:1503.02406.

[13] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In *Proc. 4th Int. Conf. on Learning Representations (ICLR 2016)*, 2016. URL https://arxiv.org/abs/1511.06361.

[14] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An overview of end-to-end entity resolution for big data. *ACM Computing Surveys*, 53(6):1–42, 2020. doi: 10.1145/3418896.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[16] Pankaj Mehta and David J. Schwab. An exact mapping between the variational renormalization group and deep learning. *arXiv preprint arXiv:1410.3831*, 2014.

[17] Maciej Koch-Janusz and Zohar Ringel. Mutual information, neural networks and the renormalization group. *Nature Physics*, 14:578–582, 2018. doi: 10.1038/s41567-018-0081-4.

[18] Frederic C. Bartlett. *Remembering: A Study in Experimental and Social Psychology*. Cambridge University Press, 1932. doi: 10.1017/CBO9780511759185.

[19] Daniel L. Schacter and Donna Rose Addis. The cognitive neuroscience of constructive memory: Remembering the past and imagining the future. *Philosophical Transactions of the Royal Society B*, 362 (1481):773–786, 2007. doi: 10.1098/rstb.2007.2087.

[20] James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex. *Psychological Review*, 102(3):419–457, 1995. doi: 10.1037/0033-295X.102.3.419.

[21] Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in Cognitive Sciences*, 20(7): 512–534, 2016. doi: 10.1016/j.tics.2016.05.004.

[22] Allan M. Collins and M. Ross Quillian. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8(2):240–248, 1969. doi: 10.1016/S0022-5371(69)80069-1.

[23] Timothy T. Rogers and James L. McClelland. *Semantic Cognition: A Parallel Distributed Processing Approach*. MIT Press, 2004.

[24] Gerard McCaul. Synthetics: Statistical lawhood under finite description. Working paper, with companion preprints, at `https://gmccaul.co.uk`, 2026.

[25] Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020. doi: 10.1109/TPAMI.2018.2889473.

[26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[27] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*, 2024.

[28] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[29] Dorothy Tse, Rosamund F. Langston, Masaki Kakeyama, Ingrid Bethus, Patrick A. Spooner, Emma R. Wood, Menno P. Witter, and Richard G. M. Morris. Schemas and memory consolidation. *Science*, 316(5821):76–82, 2007. doi: 10.1126/science.1135935.

[30] Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999. doi: 10.1038/4580.

[31] Karl Friston. The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2): 127–138, 2010. doi: 10.1038/nrn2787.

[32] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

[33] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proc. 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*, pages 1–22, 2023. doi: 10.1145/3586183.3606763.