The Open Alpha Registry: Enabling efficiently sourced information for agentic consumption

Ilan Strauss (AI Disclosures Project)*

June 2026 – brief proposal

1 The shift from human readers to agentic readers

Agentic consumption of internet material has now overtaken human consumption. Cloudflare's traffic data shows Claudebot catching up to Googlebot, with autonomous agents – not traditional search crawlers – driving the change. Training still accounts for the majority of bot traffic (about 53%), but agentic search and inference are the fastest-growing categories at roughly 10% and rising.

The web's signaling infrastructure was built for human readers. PageRank, social-feed ranking, ad-mediated traffic – all of these read off observable human behavior: links, clicks, dwell time. None of them produce the signal agentic readers need. When a chatbot or research agent retrieves a document to answer a query, it has to ask: *"what does this document teach the model that it does not already know?"*

2 What breaks: traffic, the user-preference signal, and token-efficient search

Three consequences follow from agents replacing human readers.

First, data providers lose monetizable traffic. Some of this is content being absorbed into model weights at training time, but a larger and growing share is inference-time retrieval: an agent summarizes a page so the user never visits it. The economic case for producing high-quality web content erodes with the ad impressions it no longer generates.

Second, the user-preference data that historically anchored search ranking is migrating out of search engines and into chatbots. Search ranking has long depended on signals that are noisy observations of a hidden variable – whether the document satisfied the user – including clicks, long clicks, query reformulations, and dwell time. These signals are now generated inside chatbot sessions, held by the AI provider, and no longer flow back to the public

*Fiscally sponsored by Code for Science & Society. Correspondence: ilan@aidisclosures.org.

ranking infrastructure that produced them. Search degrades on one side; chatbot providers accumulate the data on the other, with strong vertical-integration incentives to build their own retrieval indexes that monopolize the signal.

Third, agentic search loses the ability to retrieve efficiently. Token-efficient retrieval requires knowing which candidate document adds the most information relative to two things: what other candidates in the retrieval set already carry, and what the consuming model already knows. Without that signal, the agent over-fetches, pulling in paraphrases and content the model has memorized alongside whatever genuinely new material it needs. The cost shows up as latency, compute spend, and degraded answer quality, because the model's context window is finite and redundant content crowds out novel evidence.

3 Alpha: the missing signal

Ranking by human preference alone, without any model-relative scoring, does not distinguish documents by novelty to the machine. It can route an agent to the highest-clicked Wikipedia paragraph on photosynthesis when a frontier model already has that paragraph memorized verbatim, paying retrieval cost and attribution credit for information the model gains nothing from. An agentic-era ranker needs to know what the model already knows.

Borrowing the term from finance: alpha is the part of a thing's return that a benchmark does not already explain. Applied to web content read by AI systems, the benchmark is what a reference model already knows, and a document's alpha is the marginal information it adds.

Fix a reference language model M, a candidate document d, and a query q. Write $M \oplus d$ for the model when d is supplied as additional conditioning (in-context at inference time; included in fine-tuning for calibration audits; the chosen channel is recorded in the published score). The per-query information gain of d on q is the reduction in M's cross-entropy loss on the accepted completion:

$$\alpha(d \mid q;\, M) \;=\; \ell(q;\, M) \;-\; \ell(q;\, M \oplus d), \tag{1}$$

where $\ell(q;\, M) = -\log p_M(y_q^* \mid q)$ and y_q^* is the answer the user took to be correct. At the level of M's predictive distribution, this is a Kullback–Leibler movement of beliefs toward the truth on q:

$$\alpha(d \mid q;\, M) \;=\; D_{\mathrm{KL}}\big(p^*(\cdot \mid q) \,\big\|\, p_M(\cdot \mid q)\big) \;-\; D_{\mathrm{KL}}\big(p^*(\cdot \mid q) \,\big\|\, p_{M\oplus d}(\cdot \mid q)\big), \tag{2}$$

the portion of M's gap to the truth on q that d closes [Shannon, 1948, Kullback and Leibler, 1951]. Aggregating over a benchmark task distribution T:

$$\alpha(d;\, M, T) \;=\; \mathbb{E}_{q\sim T}\big[\,\alpha(d \mid q;\, M)\,\big]\,. \tag{3}$$

Three properties follow directly from the definition. Alpha is *benchmark-relative*: the same document scores differently against different reference models. A Wikipedia article on photosynthesis has near-zero alpha against a frontier model that has memorized it, but substantial

alpha against a smaller one. Alpha is also *task-relative*: a Tigrinya-language news document has high alpha for Tigrinya-language tasks and near-zero alpha for C++ tasks. And alpha *decays*: once a successor model has absorbed d, the gain against that model falls toward zero. The implication is that the same document carries different alpha against different model versions over time, and any ranking that uses alpha as a signal has to be recomputed against current reference models.

A registry paying for novelty has to condition on what it already holds. Writing R for the registered corpus, the operative quantity is the per-query marginal gain over the model *and* the corpus:

$$\alpha^*(d \mid q;\, M, R) \;=\; \ell\big(q;\, M \oplus R\big) \;-\; \ell\big(q;\, M \oplus R \oplus d\big). \tag{4}$$

A useful property of α^* is that it collapses paraphrase. The first document to register a fact gets the full informational gain in its score. A later document carrying the same fact registers against a corpus that already contains it, so its α^* shrinks to whatever residual it adds. The registry therefore differs from a flat list of documents because novelty is defined against the registered corpus itself. A competitor computing "novelty" against their own private corpus would get a different, non-comparable number. The registry's R becomes the canonical reference frame, the way a stock exchange's order book is the canonical price, because the reference state is large and is the shared standard.

4 Relevance is necessary, alpha is sufficient

Per-query alpha is query-conditional by construction. A document irrelevant to q contributes no informational gain (no reduction in the model's cross-entropy loss on q's accepted answer): if d concerns photosynthesis and q concerns Python type inference, conditioning M on d does not move its prediction on q, and $\alpha(d \mid q;\, M) \approx 0$. Relevance is therefore the necessary condition for $\alpha > 0$ – the definition enforces it; no separate filter is required.

Relevance is not sufficient. A highly relevant document the model has already memorized, for example a Wikipedia article on photosynthesis for a frontier model trained on it, also has $\alpha \approx 0$. The model already knows the answer, and conditioning on the source does not lower its loss. Among relevant documents, alpha discriminates: a document the model has not seen scores high, and a document the model has already absorbed scores near zero. Alpha refines relevance, measuring relevance to what the model does not already know.

This collapses what looks like two distinct jobs for alpha into one pipeline, ordered from cheap to expensive:

1. **A cheap precomputed estimate.** A query-independent score $g(d)$ approximating d's expected alpha over a benchmark task distribution T, computed once at admission. This is the cache that lets an agent prune the corpus before any per-query work runs.
2. **Relevance retrieval.** Standard semantic or lexical retrieval (sentence-embedding ANN, BM25) against the query q. Necessary by construction; without it the registry spends compute scoring documents that cannot move the model on q.

3. **Per-query alpha at serving time.** The expensive measurement: $\alpha^*(d \mid q; M, R)$ run on the small set of relevant candidates. This determines the ranking the agent uses for the query.

Without a model-relative novelty signal, agentic retrieval wastes tokens pulling in paraphrases of what the model already knows. The reader pays in latency and compute, and the answer is no better. A model-relative signal lets the agent route to content that actually moves the model on the query, cutting retrieval cost and improving answer quality.

The fingerprint graph supports the corpus-conditioning property of α^*. It is built from a cascade of byte-level hashing, shingle/MinHash near-duplicate signatures [Broder, 1997, Charikar, 2002], and approximate-nearest-neighbour search over learned sentence embeddings [Johnson et al., 2017]. The graph establishes priority among near-duplicates, so that when two documents carry the same fact, α^* attaches to whichever was registered first, and later submissions register only the residual they add.

The registry's distinctive contribution is the corpus conditioning. Relevance is the agent's job. Alpha against M alone can be computed by any model operator. The corpus-marginal α^*, which discounts a candidate against everything already admitted to R, requires knowledge of R. Only a venue sitting across federated operators has that knowledge.

5 User signals: a separate input

User behavior against deployed products – which documents users opened, which answers they accepted, which queries they reformulated – generates demand signals distinct from alpha. These should ideally help inform the registry over time, contributed by federated operators, and could form the basis for a public-interest search index that federates model signals.

6 What might a registry for alpha calculations look like?

R is a federated metadata registry modeled on Crossref and OpenAlex. It holds derivative records – identity codes, provenance fields, machine-readable rights, alpha vectors against the published reference models, and demand aggregates contributed by federated operators – over a body of source documents that stay with their original publishers or repositories. Each record's identity is an ISCC code: a cryptographic hash plus similarity-preserving fingerprints plus a semantic embedding. Provenance is a first-registration timestamp with pointers to upstream ancestors and downstream derivatives. Rights are machine-readable license terms, conformant with RSL or CoMP. The alpha vector carries one entry per published reference model, computable from the source text either by the registry at admission time or by the publisher locally and submitted as a non-invertible projection. The demand aggregate is a running per-(document, query-cluster) summary updated from federated operator submissions.

R is built in two passes. A one-time bootstrap ingests publicly indexable content with stable identifiers – Crossref-indexed scholarly works, OpenAlex, Wikipedia, Europe PMC,

the permissively-licensed slice of Common Crawl – running each item through the fingerprint cascade and computing initial alpha vectors against the reference models. After bootstrap, R extends continuously through voluntary submission. Publishers and federated operators submit either source text for the registry to fingerprint, or pre-computed fingerprints and alpha projections if they want to keep the source local. New submissions that the cascade finds no ancestor for become top-level records. Submissions that match existing content become derivative records pointing to their nearest ancestor, with their alpha credit reduced to whatever residual they add. The resulting object is shaped like Crossref or OpenAlex: gigabytes to low terabytes of metadata at 10^9 entries, indexing a body of source content many orders of magnitude larger that the registry itself never holds.

Appendix

A Estimation in practice

At registry scale (10^7 to 10^9 documents), producing $M \oplus d$ by retraining the model is infeasible – a frontier training run costs nine figures [Hoffmann et al., 2022]. The pipeline solves this two ways.

First, the fingerprinting cascade admits a candidate only after deduplication against R. Alpha is computed only on content already established as non-redundant.

Second, alpha is estimated by a tiered procedure. **Tier 1** is per-document surprisal under M, a single forward pass used as a quality input. A random key-smash has high surprisal and no task value, which is why surprisal alone is not the published alpha. **Tier 2** is gradient-based training-data attribution via TracIn [Pruthi et al., 2020] and TRAK [Park et al., 2023, Grosse et al., 2023], which estimates the change in test-task loss from gradients at saved checkpoints, without retraining. **Tier 3** is the per-query retrieval-augmented lift:

$$\alpha_{\mathrm{RAG}}(d \mid q;\, M, R_q) \;=\; \ell\big(q;\, M \oplus R_q\big) \;-\; \ell\big(q;\, M \oplus (R_q \cup \{d\})\big), \tag{5}$$

This is two forward passes per (query, document) pair, with no gradients. A consumer can replay both passes and verify the result. α_{RAG} falls as R grows to contain more of what d contains, so the registry's ranking signal accounts for the decay property of alpha without any explicit time-discount rule. **Tier 4** is a periodic counterfactual audit by leave-one-out retraining or datamodels-style subset regression [Ilyas et al., 2022], reserved for arbitration and for keeping Tier 2 honest.

A working cost estimate: scoring 10^8 documents against three reference models on three task families using TRAK runs in the low millions of dollars per scoring epoch, comparable to a single moderate pretraining run.

B Two candidate ways to store alpha

Because alpha is conditional on the model, the query, and the corpus, a single scalar per document is unlikely to be the right storage form. The proposal will evaluate two candidate architectures during the prototype phase. They are sketched here.

(1) Alpha as a vector. The registry could store the document's query-independent half: a TRAK-style projected gradient $\widehat{g}(d;\, M)$ (one per reference model), plus a semantic embedding $\varphi(d)$ and an ISCC identity key (ISO 24138). Per-query alpha would then be recovered at serving time as the inner product of that stored half with a query-side object. TRAK [Park et al., 2023] is this construction. Its random-projection step uses a Johnson–Lindenstrauss matrix $\Pi \in \mathbb{R}^{k \times p}$ to bring gradients from parameter dimension p (billions for

frontier models) down to projection dimension k:

$$\widehat{g}(d;\,\theta) \;=\; \Pi\,\nabla_\theta \ell(d;\,\theta). \tag{6}$$

Because the projected gradients are non-invertible, publishers could compute them locally and submit only the projection, so the registry could score content it never holds. Novelty against the corpus would then be the component of $\widehat{g}(d)$ orthogonal to its registered neighbours.

(2) Alpha as a sketch update. An alternative is for the registry to hold a small bank of frozen probe models and a running summary of their predictive state on a fixed public evaluation grid. The summary is a sufficient statistic for how each probe predicts across a canonical task lattice. When a candidate arrives, the registry runs it through the probe with one forward pass and records the delta to the sketch: the amount the model's predictive distribution over the grid moves toward the truth once d is conditioned on. Alpha would live in that recorded movement, stored as a sparse update against the sketch. This is the datamodels and influence-sketching idea [Ilyas et al., 2022] adapted to a registry data structure.

The two approaches differ in where the conditioning lives. The vector approach defers conditioning to a runtime inner product, expensive to populate but flexible against any query distribution. The probe-sketch approach bakes the corpus conditioning into the stored state at write time, cheaper to serve but well-defined only on the tasks the lattice covers. The prototype phase will compare them on a controlled benchmark and recommend a path.

References

[Broder, 1997] Broder, A. Z. (1997). On the Resemblance and Containment of Documents. *Compression and Complexity of Sequences*.

[Charikar, 2002] Charikar, M. S. (2002). Similarity Estimation Techniques from Rounding Algorithms. *Proc. STOC*.

[Grosse et al., 2023] Grosse, R. et al. (2023). Studying Large Language Model Generalization with Influence Functions. arXiv:2308.03296.

[Hagiu and Wright, 2023] Hagiu, A. and Wright, J. (2023). Data-enabled learning, network effects, and competitive advantage. *RAND Journal of Economics*.

[Hoffmann et al., 2022] Hoffmann, J. et al. (2022). Training Compute-Optimal Large Language Models. arXiv:2203.15556.

[Ilyas et al., 2022] Ilyas, A. et al. (2022). Datamodels: Predicting Predictions from Training Data. *Proc. ICML*.

[Johnson et al., 2017] Johnson, J., Douze, M., and Jégou, H. (2017). Billion-scale similarity search with GPUs. arXiv:1702.08734.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *Annals of Mathematical Statistics* 22(1).

[Park et al., 2023] Park, S. M. et al. (2023). TRAK: Attributing Model Behavior at Scale. *Proc. ICML*.

[Pruthi et al., 2020] Pruthi, G. et al. (2020). Estimating Training Data Influence by Tracing Gradient Descent. *NeurIPS*.

[Shannon, 1948] Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal* 27.