# HUBS OR FRINGES: PRETRAINING DATA SELECTION VIA WEB GRAPH CENTRALITY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The performance of modern language models depends critically on pretraining data composition. Yet existing data selection methods rely on auxiliary classifiers for document scoring or mixture optimization, adding computational overhead and dependence on labeled data. We propose WEBGRAPHMIX, a lightweight data selection framework that computes structural centrality scores over the Common Crawl host-level web graph and uses them to vary the proportion of central versus peripheral documents in the pretraining mixture. We hypothesize that central hosts expose models to reusable abstractions, while peripheral hosts encode specialized, long-tail knowledge. WEBGRAPHMIX computes centrality scores efficiently at web scale, requiring no model training, labeled data, or downstream supervision. We integrate WEBGRAPHMIX into the DataComp-LM pipeline and train models at 400M and 1B parameter scales with 8B and 28B tokens respectively, evaluating on 23 tasks ranging from factual knowledge to symbolic reasoning. Our experiments show that central and peripheral web regions encode complementary capabilities. Mixture combining both at a ratio of 1:1 achieves 41.4% on average, compared to 39.8% for uniform sampling. Combining structural scores with document-level quality classifier scores further improves performance to 43.8%. These findings demonstrate that web graph topology is a meaningful axis for pretraining data curation, capturing information that is largely orthogonal to existing content-based approaches.

## 1 INTRODUCTION

The performance of modern language models (LMs) depends critically on the composition of their pretraining data. While neural scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022) characterize how data size affects performance, far less is understood about how the structure of large-scale web corpora should influence data selection.

In practice, modern pretraining pipelines rely on massive web dumps that are filtered, deduplicated, and sampled at the document level (Albalak et al., 2024). These pipelines implicitly treat documents as independent units, applying heuristic quality filters or domain classifiers without considering relationships between documents (Soldaini et al., 2024). As a result, existing approaches largely ignore how information is organized across the web.




Figure 1: Subgraph of the Common Crawl host-level web graph. Node size is proportional to their Betweenness centrality score.

However, the web is fundamentally a graph. Webpages and hosts are connected through hyperlinks, forming a large-scale network that encodes topical structure, citation patterns, and information flow. We hypothesize that the structural



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

position of a document's *host* in this graph may correlate with the type and transferability of knowledge the document provides during pretraining. Structurally central hosts—those that lie on many shortest paths or connect diverse regions—act as hubs or bridges between otherwise weakly connected communities, and their content is more likely to co-occur with heterogeneous contexts and expose models to reusable abstractions. In contrast, peripheral hosts may carry specialized or long-tail content that is less broadly shared. From a language modeling perspective, this suggests that graph structure may influence the diversity and overlap of token-level learning signals, and therefore shape the capabilities learned during pretraining.

In this work, we introduce WEBGRAPHMIX, a graph-based data selection framework that operates directly on the hyperlink graph and is fully unsupervised. We compute centrality over the Common Crawl host-level graph, mainly betweenness (Freeman, 1977) and Katz (Katz, 1953) centrality, and construct training mixtures that emphasize structurally *central* data, *peripheral* data, or combinations of the two. Eigenvector centrality, the undirected analogue of PageRank (Page et al., 1999), did not improve over uniform sampling, consistent with DCLM (Li et al., 2024).

Unlike domain-based methods, which construct semantic taxonomies or optimize domain mixtures through regression or proxy training (Wettig et al., 2025; Xie et al., 2023a; Liu et al., 2025), or quality-based methods, which score documents with learned classifiers (Penedo et al., 2024; Li et al., 2024; Wettig et al., 2024), WEBGRAPHMIX needs no taxonomy, classifier, or regression model, only structural signals intrinsic to the web graph.

We integrate WEBGRAPHMIX into the DataComp-LM (DCLM) pipeline (Li et al., 2024), train 400M and 1B parameter models, and evaluate on the 23 tasks of DCLM CORE v2. Centrality scores for the full host graph (13.9M nodes, 439.6M edges) take under 9 hours to compute on at most four H100 GPUs and are reused across all experiments.

Our main findings at 1B scale are:

- **Combining structural extremes helps.** A 50/50 mix of central and peripheral hosts reaches 41.4% average accuracy versus 39.8% for uniform sampling, outperforming either extreme alone as well as the middle of the distribution, and raises *Reasoning* from 19.6% to 21.4%.
- **Centrality complements content quality.** Combined with the DCLM-fasttext quality classifier, centrality reaches 43.8%, 4.0 points above uniform sampling and 1.5 above the classifier alone, and every combined configuration we ran beats the classifier.
- **The useful signal is not link counting.** Performance falls as the selection signal becomes more degree-like (betweenness 41.4%, Katz 40.8%, in-degree 40.2%).

## 2 RELATED WORK

**Document-level curation.** Existing data curation largely treats documents as independent units. Rule-based filters remove boilerplate, spam, and malformed text (Raffel et al., 2020; Rae et al., 2021; Penedo et al., 2023), and deduplication with MinHash (Broder, 1997; Lee et al., 2022) or Bloom filters (Soldaini et al., 2024) removes near-duplicates; DataComp-LM (Li et al., 2024) standardizes these steps for compute-controlled comparisons. Quality-based methods such as FineWeb-Edu (Penedo et al., 2024), DCLM-fasttext (Li et al., 2024), QuRating (Wettig et al., 2024), and Ask-LLM (Sachdeva et al., 2026) score documents by properties such as educational value, BETR (Mizrahi et al., 2025) selects documents similar to benchmark examples, and other methods use perplexity (Wenzek et al., 2020), n-gram overlap (Xie et al., 2023b), or attention-based signals (Hua et al., 2025). All of these treat selection as ranking independently scored documents.

**Domain mixture optimization.** DoReMi (Xie et al., 2023a), RegMix (Liu et al., 2025), TiK-MiX (Wang et al., 2025), DoGE (Fan et al., 2024), and Aioli (Chen et al., 2025) optimize weights over predefined domains using proxy models, regression, or influence estimates, while Skill-it (Chen et al., 2023), WebOrganizer (Wettig et al., 2025), Nemotron-CLIMB (Diao et al., 2026), and Group-MATES (Yu et al., 2026) first construct their own domains by clustering or taxonomy design. These approaches yield strong gains but typically require substantial computation, model training, or downstream supervision, and they derive structure from content similarity rather than from the native connectivity of the web.

**Web graph structure.** Hyperlinks encode citation, topical proximity, and information flow, and link-analysis methods such as PageRank (Page et al., 1999) and HITS (Kleinberg, 1999) have long exploited them for ranking. Craw4LLM (Yu et al., 2025) uses quality signals to prioritize pages *during* crawling, MeCo (Gao et al., 2025) conditions on URL metadata during training with gains that persist under URL anonymization, and LinkBERT (Yasunaga et al., 2022) uses links to construct training contexts by placing linked documents in the same window. In contrast, we use a host's position in the link graph to select and weight documents within an already-crawled corpus, which to our knowledge prior work has not explored.

# 3 Our Method: WebGraphMix

We introduce WebGraphMix, a lightweight pretraining data selection framework that leverages structural signals from the web graph. Rather than scoring documents independently based on content, our method assigns each document the centrality score of its host in the Common Crawl host graph and uses these scores to guide sampling.

## 3.1 Web Graph Construction

We operate on the Common Crawl host-level graph[1], where each node is a web host (e.g., `wikipedia.org`). An edge connects two hosts if any page on one links to any page on the other; edges are unweighted and deduplicated, so repeated hyperlinks do not create multi-edges. The graph has 13.9M nodes and 439.6M edges. Although hyperlinks are directed, we compute all centrality scores on the symmetrized (undirected) graph $G = (V, E)$: in the directed host graph most node pairs are mutually unreachable, so directed betweenness collapses onto the strongly connected core and assigns zero to nearly every other host. Consistent with this, betweenness correlates almost equally with a host's in-links and out-links, i.e., the number of hosts linking to it and the number it links to in the directed graph (Spearman $\rho = 0.735$ and $0.746$). We refer to the in-link count as in-degree. Shortest paths and walks are computed over host nodes only, so the number of documents a host contains plays no role in its score. The symmetrized graph has a single giant connected component containing 99.95% of hosts and 99.98% of tokens in our corpus, so component membership does not confound our comparisons (Appendix D.1).

The raw pretraining corpus we use, Corpus-200B[2] from Wettig et al. (2025), is a pre-processed version of the `1b-1x` CommonCrawl pool from DataComp-LM (Li et al., 2024) cleaned with RefinedWeb filters (Penedo et al., 2023) and BFF deduplication (Groeneveld, 2024). Each document in the preprocessed corpus is mapped to its corresponding host via its URL. We discard about 5% of the documents in the corpus without a host in the web graph. Centrality scores are computed at the host level and inherited by all associated documents. Specifically, if a host $v$ has centrality score $c(v)$, then each document $d_i$ from that host is assigned score $s_i = c(v)$. Common Crawl itself prioritizes URLs by harmonic centrality, so this pool is already denser at the center of the web than the open web is.

## 3.2 Centrality Score

We quantify structural importance using classical graph centrality measures that capture complementary aspects of connectivity.

**Betweenness centrality** (Freeman, 1977) measures how frequently a node lies on shortest paths between other nodes:

$$c_B(v) = \sum_{s \neq v \neq t} \frac{\sigma(s, t \mid v)}{\sigma(s, t)}, \tag{1}$$

where $s, t, v \in V$, $\sigma(s, t)$ is the number of shortest paths from $s$ to $t$, and $\sigma(s, t \mid v)$ counts those passing through $v$. Nodes with high betweenness act as bridges between otherwise weakly connected regions. Exact betweenness costs $O(|V||E|) \approx 6 \times 10^{15}$ operations on this graph, so we use

[1] We use `cc-main-2023-24-sep-nov-feb-host` from https://commoncrawl.org/web-graphs.

[2] https://huggingface.co/datasets/WebOrganizer/Corpus-200B

cuGraph's sampling approximation of Brandes' algorithm with 1.4M source vertices. Appendix C shows example content from the highest- and lowest-betweenness hosts.

**Katz centrality** (Katz, 1953) aggregates contributions from all walks in the graph:

$$c_K(v_i) = \eta \sum_j A_{ij} c_K(v_j) + \tau, \tag{2}$$

where $A$ is the adjacency matrix, $v_i, v_j \in V$, $i$ and $j$ both index all nodes in the graph, $\lambda_{\max}$ is the largest eigenvalue of $A$, $0 < \eta < 1/\lambda_{\max}$ ensures convergence, and $\tau$ is a bias term. This assigns higher scores to nodes connected to other influential nodes, while attenuating longer paths. We use cuGraph's default $\eta = 1/d_{\max} = 9.80 \times 10^{-8}$, which satisfies the convergence condition since $d_{\max} \geq \lambda_{\max}$, and $\tau = 1$. This attenuation is strong, so as parameterized our Katz scores are dominated by the first-order degree term (Spearman $\rho = 0.831$ with in-links). These measures capture complementary notions of structural importance: Betweenness emphasizes cross-community connectivity, while Katz centrality, as parameterized here, mainly reflects a host's degree.

**Eigenvector centrality.** PageRank (Page et al., 1999) is a variant of eigenvector centrality for directed graphs. Because we work on the symmetrized graph, we test eigenvector centrality directly. It does not improve over uniform sampling (39.6 vs. 39.8; Table 6), matching DCLM's finding that selecting top quantiles by PageRank does not outperform uniform sampling (Li et al., 2024). We therefore focus on betweenness and Katz and report eigenvector centrality as a baseline.

**Efficiency and scalability.** A key advantage of WEBGRAPHMIX is that centrality scores can be computed efficiently at web scale using distributed graph algorithms. We implement centrality computation over the host graph using GPU-parallelized primitives and graph partitioning with the `cuGraph` library[3]. In practice, computing Katz centrality (Foster et al., 2001) for the full Common Crawl host graph took us < 3 hours on one H100 GPU and computing Betweenness centrality (Brandes, 2001) took us < 6 hours on 4 H100 GPUs, after which the scores can be reused across all downstream experiments.

Unlike prior data selection methods that require repeated model training, gradient computation, or proxy evaluation, this is a *compute-efficient one-time preprocessing step*. Once computed, centrality scores incur negligible overhead during data sampling.

### 3.3 CENTRALITY-GUIDED DATA SAMPLING

Each host can be viewed as a subdomain embedded within the global web graph. Hosts differ substantially in their structural roles: some connect multiple regions of the graph and act as hubs or bridges, while others lie in sparsely connected or peripheral regions. We hypothesize that these structural differences correspond to differences in the type of knowledge encoded: Structurally central hosts are more likely to expose models to broadly reusable and cross-domain patterns, whereas peripheral hosts tend to contain specialized or long-tail information. Appendix C shows example crawled content from central and peripheral hosts. To study this effect, we construct data mixtures that vary systematically across the centrality spectrum.

Given host-level centrality scores $c(v)$, each document inherits a score $s_i = c(v_i)$ based on its host $v_i$. We then construct training datasets under a fixed token budget using the following sampling strategies.

**Top-$K$ (Central) sampling**: We select documents whose hosts fall within the top percentile of the centrality distribution (e.g., top 25%, or 75%), emphasizing structurally central regions of the web.

**Bottom-$K$ (Peripheral) sampling**: We select documents from the lowest percentile of the centrality distribution, focusing on peripheral or long-tail regions. Approximate betweenness assigns exactly zero to a large set of hosts, so the lowest percentiles contain ties. Zero-score hosts account for less than 25% of our token budget, so Bottom-$K$ always includes all of them and fills the remainder in ascending score order.

**Middle-$K$ sampling**: As a control for mixed sampling, we select documents from hosts in the middle 50% of the betweenness distribution (between the 25th and 75th percentiles), excluding both tails.

[3] https://github.com/rapidsai/cugraph

**Mixed sampling**: To test whether central and peripheral regions provide complementary signals, we construct mixtures combining both strata:

$$\alpha \cdot \text{Top-}K + (1-\alpha) \cdot \text{Bottom-}K, \tag{3}$$

where the mixture ratio $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$. Documents are sampled proportionally until the token budget is reached. Here Top-$K$ and Bottom-$K$ are percentile strata of a single centrality score; the combined method in §3.4 constructs its two strata differently.

### 3.4 Combining Structural and Quality Signals

In addition to pure structural selection, we explore combining centrality scores with document-level quality scores. We use the quality scores produced by DCLM-fasttext (Li et al., 2024), a bigram model trained to classify high quality text sampled from different sources and low quality text sampled from RefinedWeb (Penedo et al., 2023) reproduction. The classifier is never applied as a threshold: no document is removed from the pool before sampling. We normalize both the centrality scores and the quality scores by:

$$\hat{s}_i = \exp(s_i - \max_j s_j), \tag{4}$$

where $i$ and $j$ both index all documents in the pool, each document inheriting its host's centrality score. This gives us a score within $(0, 1]$. After normalizing both signals, we combine graph centrality and document quality in two complementary ways. For Top-$K$ selection, we use additive and multiplicative scores,

$$\hat{s}_i^{\text{add}} = \hat{s}_i^{\text{centrality}} + \hat{s}_i^{\text{quality}}, \qquad \hat{s}_i^{\text{mult}} = \hat{s}_i^{\text{centrality}} \cdot \hat{s}_i^{\text{quality}},$$

which favor documents that are both central in the web graph and high quality. For Bottom-$K$ selection, we instead use contrastive scores,

$$\hat{s}_i^{\text{sub}} = \hat{s}_i^{\text{centrality}} - \hat{s}_i^{\text{quality}}, \qquad \hat{s}_i^{\text{div}} = \hat{s}_i^{\text{centrality}} / \hat{s}_i^{\text{quality}},$$

and select documents with the lowest scores, thereby prioritizing high-quality documents that are less central. Both rankings are computed over the same full document pool. For a mixture ratio $\alpha$, the Top-$K$ portion fills $\alpha$ of the token budget by taking documents greedily in descending order of the additive or multiplicative score, and the Bottom-$K$ portion fills the remaining $1-\alpha$ in ascending order of the subtractive or divisive score. The two selections are disjoint in every configuration we report, so the realized mixture ratio equals $\alpha$ and no document is duplicated. Unlike the pure-centrality setting of §3.3, where Top-$K$ and Bottom-$K$ are percentile strata of a single score, here $\alpha$ divides the budget between two different rankings. These strategies allow us to test whether graph structure provides a signal complementary to document quality.

## 4 Experiments

### 4.1 Experimental Setup

All experiments use the official DataComp-LM (DCLM) framework, with its standardized data pools, model architectures, compute-optimal token budgets, and training and evaluation pipeline, at two scales: `400m-1x` (412M parameters, about 8.2B tokens) and `1b-1x` (1.4B parameters, about 28B tokens). A `1b-1x` run takes about 240 H100-hours (Appendix A), and `7b-1x` is about an order of magnitude beyond our budget; contemporaneous data-selection work operates in the same regime (Magnusson et al., 2025; Yu et al., 2026). We report 1B results in the main text and full 400M results in Appendix B. We foreground 1B because it is the only setting in which WebOrganizer and WebOrganizer+ report results, and because DCLM CORE v2 has much more measurable range at 1B: summed over the 22 tasks with well-defined chance levels, the above-chance margin is 4.09 at 1B against 2.43 at 400M, and at 400M ARC Challenge sits below chance while SQuAD, CoQA, and Jeopardy score only 10.7%, 12.4%, and 1.6%.

We report task-level and average accuracy on the 23 tasks of DCLM CORE v2 (Li et al., 2024), grouped into five categories[4] (Table 3): Commonsense & Reasoning, QA & Comprehension, Sci-

[4]As marked in the DCLM metadata: https://github.com/mlfoundations/dclm/blob/main/eval/eval_meta_data.csv

ence & Factual Knowledge, Symbolic & Algorithmic Reasoning, and Language Understanding, abbreviated as *Commonsense*, *Comprehension*, *Knowledge*, *Reasoning*, and *Language*.

**Uncertainty.** Bootstrapping evaluation examples (resampling each task's test examples with replacement) gives about $\pm 0.36$ on the 1B 23-task average (Random $39.84 \pm 0.36$, WEBGRAPHMIX $41.42 \pm 0.37$, Quality $42.30 \pm 0.33$). Retraining the 400M uniform baseline with three seeds gives $\sigma \approx 0.15$ (32.50, 32.79, 32.57). We treat average differences below roughly 0.4 points as unresolved.

We consider the following baselines: **Random**: uniformly randomly select from the data pool; **Quality**: select documents with top-$K$ quality score produced by DCLM-fasttext; **WebOrganizer** (Wettig et al., 2025): topic and format domain pairs mixture predicted from RegMix (Liu et al., 2025) pipelines; **WebOrganizer+** (Wettig et al., 2025): the WebOrganizer domain mixture combined with DCLM-fasttext quality filter; **Eigenvector**: select documents from the top-$K$ hosts by eigenvector centrality on the symmetrized graph (§3.2); **In-degree mixture**: the same 50/50 Top-$K$/Bottom-$K$ mixture as WEBGRAPHMIX, but ranking hosts by raw in-degree, which tests whether simple link counting reproduces the effect; **Middle-$K$**: documents from the middle 50% of the betweenness distribution (§3.3).

### 4.2 MAIN RESULTS

Recall that WEBGRAPHMIX selects pretraining data by computing host-level centrality scores over the Common Crawl web graph and constructing a *mixed* dataset that combines documents from the top (central) and bottom (peripheral) quantiles of the centrality distribution, with mixture ratio $\alpha$ controlling the balance between the two strata (see Equation (3)). Among the two centrality measures explored, we find *betweenness centrality* most effective; it identifies hosts that bridge otherwise weakly connected regions of the web, which we hypothesize yields documents with broadly reusable, cross-domain patterns. WEBGRAPHMIX+ further incorporates document-level quality scores from DCLM-fasttext: for the Top-$K$ stratum, documents are ranked by the *multiplicative* combination of centrality and quality scores; for the Bottom-$K$ stratum, documents are ranked by the *division* of centrality by quality, thereby surfacing high-quality documents that are structurally peripheral.

Table 1 compares our best configurations with the baselines, averaged over task categories. We report two results separately. *Centrality alone*: mixing Top-$K$ and Bottom-$K$ documents ranked by betweenness at $\alpha = 0.5$ reaches 41.4%, 1.6 points above random selection and about four times the bootstrap interval, and improves over random selection on all task categories. It uses no quality signal and remains below the Quality baseline (42.3%). *Centrality combined with quality* (WEBGRAPHMIX+): adding the DCLM-fasttext score as in §3.4 reaches 43.8%, 1.5 points above the Quality baseline and 4.0 above random selection, and improves over the Quality baseline on 4 out of 5 categories.

The WebOrganizer baseline requires significant human effort for proposing the domain taxonomies. It also substantial computation: training 512 proxy models of 50M parameters and fitting a gradient-boosted regression model to optimize domain weights toward specific target tasks, namely MMLU (Hendrycks et al., 2021) and HellaSwag (Zellers et al., 2019). This explains WebOrganizer+'s strong performance on Commonsense—HellaSwag is a commonsense sentence-completion benchmark—but limits the method's generalizability to other capability categories. WEBGRAPHMIX+ matches/exceeds WebOrganizer+ on overall average (43.8% vs. 43.4%) while requiring no proxy training, no labeled targets, and no benchmark-specific tuning.

**Results at 400M.** The 400M results are weaker, and the best configuration differs between scales (Table 5 in Appendix B). Pure centrality selection does not help at 400M: the 1B-best mixture (betweenness 50/50) scores 1.0 point below uniform sampling, and no pure configuration beats uniform by more than 0.1 points (Tables 9 and 10). Combined with quality, the best 400M configuration (additive Katz Top-$K$) gains 2.6 points over uniform sampling and 0.6 over the Quality baseline, while the multiplicative betweenness configurations that perform best at 1B only match the Quality baseline.

Table 1: Accuracy on DCLM CORE v2 benchmark at 1B scale, averaged by task category. WEBGRAPHMIX uses betweenness centrality with $\alpha = 0.5$ Top-$K$/Bottom-$K$ mixture; WEBGRAPHMIX+ additionally combines centrality with the DCLM-fasttext quality score via multiplication and division. Note that while our WEBGRAPHMIX is close to WebOrganizer baseline, our method is significantly cheaper and more transferable. Per-task results are reported in Appendix B.

| Method | Commons. | Compreh. | Knowl. | Reason. | Lang. | Average |
|---|---|---|---|---|---|---|
| Random | 57.3 | 37.9 | 34.2 | 19.6 | 39.9 | 39.8 |
| Quality | 59.8 | 38.1 | 38.9 | 20.7 | **42.8** | 42.3 |
| WebOrganizer | 59.6 | 39.2 | 38.0 | 22.5 | 38.3 | 42.1 |
| WebOrganizer+ | **61.9** | 41.4 | 39.1 | 21.9 | 38.8 | 43.4 |
| Eigenvector | 56.9 | 37.4 | 34.8 | 19.3 | 38.1 | 39.6 |
| WEBGRAPHMIX | 59.5 | 39.4 | 35.4 | 21.4 | 40.2 | 41.4 |
| WEBGRAPHMIX+ | 60.8 | **42.6** | **39.7** | **22.6** | 41.9 | **43.8** |

# 5 ANALYSIS

## 5.1 STRUCTURAL POSITION DIFFERENTIALLY AFFECTS CAPABILITY CATEGORIES

Table 2 breaks down performance by capability category at the 1B scale for Top-$K$ (central) and Bottom-$K$ (peripheral) sampling and for betweenness mixtures. The effect of structural position depends on the capability, and the clearest effects come from mixing rather than from either stratum alone.

**Bottom-$K$ sampling consistently improves factual and commonsense knowledge.** The most consistent pure-stratum pattern appears in *Knowledge* and *Commonsense* task categories. In *Knowledge*, both Bottom-$K$ strategies outperform the random baseline: Betweenness Bottom-$K$ improves from 34.2% to 35.5% (+1.3%), while Katz Bottom-$K$ reaches 35.3% (+1.1%). In contrast, Betweenness Top-$K$ slightly hurts performance (-0.4%).

A similar trend appears for *Commonsense*. Katz Bottom-$K$ achieves the best score among pure strata (57.8%, +0.5%), while Katz Top-$K$ substantially underperforms the baseline (56.1%, -1.2%). Betweenness Bottom-$K$ is roughly neutral (+0.1%), while Betweenness Top-$K$ again slightly decreases performance (-0.5%).

These results suggest that peripheral regions of the web contain useful long-tail and diverse knowledge signals that are beneficial for factual recall and commonsense reasoning tasks.

**Structured reasoning benefits most from mixing Top-$K$ and Bottom-$K$ sampling.** Unlike the knowledge-oriented categories, *Reasoning* improves only slightly under pure centrality-based sampling. Katz Top-$K$ achieves the strongest pure-stratum result (20.4%, +0.8%), followed by Katz Bottom-$K$ (+0.6%), while Betweenness Top-$K$ and Bottom-$K$ produce small gains (+0.3% and +0.2% respectively). Mixing the two strata yields much larger gains: the betweenness 50/50 and 75/25 mixtures reach 21.4% (+1.8%) and 22.0% (+2.4%), the largest *Reasoning* improvements in the table.

This indicates that reasoning tasks benefit most when central and peripheral content are combined rather than drawn from either stratum alone, consistent with the two regions contributing complementary training signal.

**Comprehension and language understanding exhibit asymmetric behavior.** For *Comprehension*, Bottom-$K$ and Top-$K$ behave very differently. Katz Bottom-$K$ improves over baseline (+0.7%), while Katz Top-$K$ substantially hurts performance (-2.4%), the largest degradation in the table. A similar but weaker pattern appears for *Language Understanding*, with only Katz Bottom-$K$ improving.

One explanation would be that central hosts are less linguistically diverse. We tested this on 4,000 documents per group and found no such difference: central documents have a slightly *higher* type–token ratio (0.615 vs. 0.604) and TF-IDF dispersion (0.767 vs. 0.751), and identical mean sentence length (18.2), with the same pattern for Katz (Appendix D). We therefore leave the Comprehension drop under Katz Top-$K$ unexplained.

Table 2: Average accuracy across task categories at 1B scale for Top-$K$ and Bottom-$K$ sampling and for betweenness mixtures. Changes are relative to Random. Mixing Top-$K$ and Bottom-$K$ yields the largest gains, most notably on Reasoning. **Betw.** denotes Betweenness centrality scores. **Katz** denotes Katz centrality scores.

| Method | Commonsense | Comprehension | Knowledge | Reasoning | Language | Avg |
|---|---|---|---|---|---|---|
| Random | 57.3 | 37.9 | 34.2 | 19.6 | 39.9 | 39.8 |
| Betw. Top-$K$ | 56.8 -0.5 | 37.1 -0.8 | 33.8 -0.4 | 19.9 +0.3 | 39.3 -0.6 | 39.4 |
| Betw. Bottom-$K$ | 57.4 +0.1 | 36.9 -1.0 | 35.5 +1.3 | 19.8 +0.2 | 39.6 -0.3 | 40.0 |
| Betw. 50/50 | 59.5 +2.2 | 39.4 +1.5 | 35.4 +1.2 | 21.4 +1.8 | 40.2 +0.3 | 41.4 |
| Betw. 75/25 | 59.3 +2.0 | 38.6 +0.7 | 34.6 +0.4 | 22.0 +2.4 | 39.3 -0.6 | 41.0 |
| Katz Top-$K$ | 56.1 -1.2 | 35.5 -2.4 | 34.5 +0.3 | 20.4 +0.8 | 39.1 -0.8 | 39.2 |
| Katz Bottom-$K$ | 57.8 +0.5 | 38.6 +0.7 | 35.3 +1.1 | 20.2 +0.6 | 40.2 +0.3 | 40.5 |

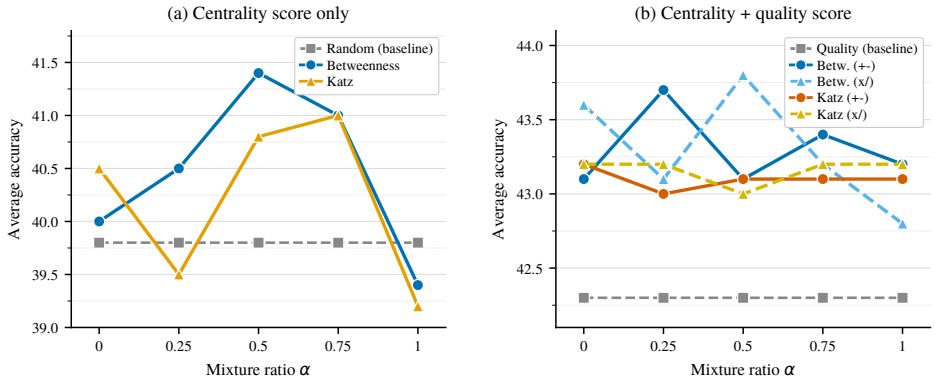


Figure 2: Average accuracy of mixture sampling with Betweenness centrality score and Katz centrality score at 1B scale, varying ratio of Top-$K$ and Bottom-$K$ tokens. (a) is centrality score only. (b) is centrality score combined with quality score. $+-$ means combining with addition for Top-$K$ and subtraction for Bottom-$K$. $\star/$ means combining with multiplication for Top-$K$ and division for Bottom-$K$.

**Centrality metric matters.** The two centrality measures also behave differently. Katz produces larger swings than betweenness in both directions; for Top-$K$ these range from +0.8% on *Reasoning* to -2.4% on *Comprehension*. Because our Katz scores closely track degree (§3.2), we read this as a difference between selecting well-linked hosts and selecting hosts that bridge regions of the graph, rather than as evidence that recursive influence is a stronger signal. §5.3 tests this directly.

### 5.2 Mixture Sampling

Figure 2(a) shows 23-task averages at 1B as the share of Top-$K$ tokens varies.

**Mixing outperforms pure sampling.** Neither pure Top-$K$ nor pure Bottom-$K$ achieves the gain of the $\alpha = 0.5$ mixture. It also beats the middle of the distribution. Middle-$K$ sampling scores 40.3%, the best single betweenness band, above uniform (39.8%), Top-$K$ (39.4%), and Bottom-$K$ (40.0%), so mid-centrality content is not harmful to include. Yet the Top-$K$/Bottom-$K$ mixture exceeds Middle-$K$ by 1.1 points, about three times the bootstrap interval. No single betweenness band beats uniform sampling by more than 0.5 points, while combining the two extremes gains 1.6. The gain therefore comes from combining central and peripheral content, not from excluding the middle. This is consistent with our hypothesis that the two ends of the distribution supply complementary training signal. Phrase-recurrence measurements are consistent with this: 10-word phrases from central documents recur elsewhere in the corpus 1.6–1.9× as often as those from peripheral documents (Appendix D.2).

**The optimal ratio is roughly balanced.** Across betweenness mixtures, $\alpha = 0.5$ outperforms both $\alpha = 0.25$ (40.5%) and $\alpha = 0.75$ (41.0%): performance peaks at a balanced mixture and declines

on either side. This inverted-U shape is consistent with the complementarity hypothesis: neither central nor peripheral documents should dominate. At every mixed ratio, betweenness matches or exceeds Katz, with the gap largest at $\alpha = 0.5$ (+0.6%). Katz mixtures show a similar trend, improving as the central share grows (39.5% to 40.8% to 41.0%) but dropping to 39.2% at pure Top-$K$. This is consistent with our Katz scores closely tracking degree (§3.2), so Katz mixtures behave more like selection by link count, which we test directly in §5.3.

At 400M, mixture improvements are small and less consistent. The best mixture, Katz $\alpha = 0.25$, improves 0.1% over the baseline, and Betweenness $\alpha = 0.75$ matches the baseline on average while making gains on specific tasks like ARC Easy (2.6%), Winogrande (2.4%), and ARC Challenge (2.5%). Other mixtures, including the 1B-best Betweenness $\alpha = 0.5$, fall below the baseline at this scale (§4.2).

### 5.3 Is Centrality a Proxy for Host Size or Link Count?

Because every document inherits its host's score, centrality could simply be selecting large or heavily linked sites. Across all 13.9M hosts, centrality correlates only weakly with host size (Spearman $\rho = 0.25$–$0.35$ with documents or tokens per host) but substantially with raw link counts ($\rho = 0.73$–$0.84$; Appendix D.3). We therefore test link counting directly, building the same 50/50 Top-$K$/Bottom-$K$ mixture but ranking hosts by raw in-degree. It scores 40.2%, only 0.4 points above uniform sampling, while betweenness exceeds it by 1.2 points, about three times the bootstrap interval. Performance falls as the signal becomes more degree-like (betweenness 41.4%, Katz 40.8%, in-degree 40.2%), the reverse of what a popularity account predicts, and eigenvector centrality, the most popularity-like measure we tested, does not beat uniform sampling (39.6%). Betweenness and Katz also share only 35% of their Top-$K$ hosts yet produce different capability profiles (Table 2). These results rule out host size, component membership, and raw link count as sufficient explanations, although host-level scores remain correlated with host identity and domain in ways we cannot fully separate (Appendix E).

### 5.4 Combining with Quality Scores

Figure 2(b) reports results for combining centrality with quality scores at 1B scale. The headline finding is that centrality extracts robustly positive value *on top of* the quality filter: every configuration in Figure 2(b) exceeds the quality-only baseline of 42.3%, with gains ranging from 0.5% to 1.5%. The strongest configuration, *Multiply Betweenness 50% Top, achieves an average of 43.8%*—a 1.5% improvement over quality-only and a 4.0% improvement over random sampling. This indicates that web graph centrality is not merely competitive with content-based quality scoring but consistently complementary to it. The signal centrality captures (structural position in the hyperlink graph) appears largely orthogonal to what DCLM-fasttext captures, so combining the two yields broadly compounding gains. Centrality and quality are also nearly independent at the document level: the Spearman correlation between host betweenness and DCLM-fasttext score is $\rho = +0.018$, and a top-10% quality threshold would keep 12.4% of the most peripheral decile against a 10% base rate, so the quality filter does not disproportionately remove peripheral documents (Appendix D). This near-independence is consistent with the additive gains in Figure 2(b).

## 6 Conclusion

We introduced **WebGraphMix**, a lightweight pretraining data selection framework that uses a host's position in the Common Crawl web graph as a sampling signal. Mixing central and peripheral hosts outperforms either extreme alone and the middle of the distribution, with strong gains on reasoning, and combining centrality with quality filtering improves further over the filter alone. These gains are not explained by host size or raw link counts, as performance declines when the selection signal becomes more degree-like. We hypothesize that central hosts supply broadly reused text and peripheral hosts supply long-tail knowledge. Unlike methods that require auxiliary model training, influence estimation, or taxonomy construction, **WebGraphMix** computes centrality once over a public web graph in under 9 hours on at most four GPUs, offering a lightweight signal complementary to content-based approaches and a promising new axis for pretraining data curation.

## AI DISCLOSURE

The core methodology (graph centrality computation and data selection) does not use LLMs as a component. LLMs are only used as the models being trained and evaluated, which is the standard subject of study rather than a methodological tool. We used LLMs only for writing assistance when drafting the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REPRODUCIBILITY

The paper fully discloses all the information needed to reproduce the main experimental results of the paper. Section 4.1 and Appendix A describe the full experimental setup: model scales, token budgets, the DCLM framework version, tokenizer, training configurations, and evaluation benchmark. The data pool, web graph source, and centrality computation library (cuGraph) are all specified. All training runs use identical preprocessing to isolate the effect of data selection.

## REFERENCES

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024. doi: 10.48550/arXiv.2402.16827. URL https://arxiv.org/abs/2402.16827.

Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25 (2):163–177, 2001. doi: 10.1080/0022250X.2001.9990249.

Andrei Z. Broder. On the resemblance and containment of documents. *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pp. 21–29, 1997. URL https://api.semanticscholar.org/CorpusID:11748509.

Mayee F Chen, Nicholas Roberts, Kush Bhatia, Jue WANG, Ce Zhang, Frederic Sala, and Christopher Re. Skill-it! a data-driven skills framework for understanding and training language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=IoizwO1NLf.

Mayee F Chen, Michael Y. Hu, Nicholas Lourie, Kyunghyun Cho, and Christopher Re. Aioli: A unified optimization framework for language model data mixing. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sZGZJhaNSe.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan SU, Markus Kliegl, ZIJIA CHEN, Peter Belcak, Yoshi Suhara, Hongxu Yin, Mostofa Patwary, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. Nemotron-CLIMB: Clustering-based iterative data mixture bootstrapping for language model pre-training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=aBlqKPkc4a.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: Domain reweighting with generalization estimation. In *International Conference on Machine Learning*, pp. 12895–12915. PMLR, 2024.

Kurt C. Foster, Stephen Q. Muth, John J. Potterat, and Richard B. Rothenberg. A faster Katz status score algorithm. *Computational & Mathematical Organization Theory*, 7(4):275–285, 2001. ISSN 1572-9346. doi: 10.1023/A:1013470632383. URL https://doi.org/10.1023/A:1013470632383.

Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977. ISSN 00380431, 23257938. URL http://www.jstor.org/stable/3033543.

Tianyu Gao, Alexander Wettig, Luxi He, Yihe Dong, Sadhika Malladi, and Danqi Chen. Metadata conditioning accelerates language model pre-training. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DdMMzlI5YE.

Dirk Groeneveld. BFF: The big friendly filter. https://github.com/allenai/bff, 2024. Bloom filter-based n-gram deduplication tool for language model pretraining data.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. doi: 10.48550/arXiv.2203.15556. URL https://arxiv.org/abs/2203.15556.

Kai Hua, Steven Wu, Ge Zhang, and Ke Shen. Attentioninfluence: Adopting attention head influence for weak-to-strong pretraining data selection. *arXiv preprint arXiv:2505.07293*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. doi: 10.48550/arXiv.2001.08361. URL https://arxiv.org/abs/2001.08361.

Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953. doi: 10.1007/BF02289026.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5): 604–632, 1999. doi: 10.1145/324133.324140.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.577. URL https://aclanthology.org/2022.acl-long.577/.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5BjQOUXq7i.

Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. DataDecide: How to predict best pretraining data with small experiments. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 42487–42502, 2025.

David Mizrahi, Anders Boesen Lindbo Larsen, Jesse Allardice, Suzie Petryk, Yuri Gorokhov, Jeffrey Li, Alex Fang, Josh Gardner, Tom Gunter, and Afshin Dehghan. Language models improve when pretraining data matches target tasks. *arXiv preprint arXiv:2507.12466*, 2025.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon LLM: Outperforming curated corpora with web data only. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=kM5eGcdCzq.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H. Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. How to train data-efficient LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yKUbw7q1IA.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. doi: 10.48550/arXiv.2402.00159. URL https://arxiv.org/abs/2402.00159.

Yifan Wang, Binbin Liu, Fengze Liu, Yuanfan Guo, Jiyao Deng, Xuecheng Wu, Weidong Zhou, Xiaohuan Zhou, and Taifeng Wang. Tikmix: Take data influence into dynamic mixture for language model pre-training. *arXiv preprint arXiv:2508.17677*, 2025.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4003–4012, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.494/.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. QuRating: Selecting high-quality data for training language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arXiv.2402.09739. URL https://arxiv.org/abs/2402.09739.

Alexander Wettig, Kyle Lo, Sewon Min, Hannaneh Hajishirzi, Danqi Chen, and Luca Soldaini. Organize the web: Constructing domains enhances pre-training data curation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. doi: 10.48550/arXiv.2502.10341. URL https://arxiv.org/abs/2502.10341.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=lXuByUeHhd.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://arxiv.org/abs/2302.03169.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. LinkBERT: Pretraining language models with document links. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8003–8016, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.551. URL https://aclanthology.org/2022.acl-long.551/.

Shi Yu, Zhiyuan Liu, and Chenyan Xiong. Craw4LLM: Efficient web crawling for LLM pretraining. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13843–13851, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.712. URL https://aclanthology.org/2025.findings-acl.712/.

Zichun Yu, Fei Peng, Jie Lei, Arnold Overwijk, Wen tau Yih, and Chenyan Xiong. Group-level data selection for efficient pretraining. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=uX4dyc7Z5Z.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

# A Experiment Details

After selecting documents according to our graph-based sampling strategy, we construct an untokenized dataset in JSONL format consistent with DCLM specifications. Tokenization and shuffling are performed using DCLM's official Rust-based tokshuf pipeline. Specifically, we tokenize with the GPT-NeoX tokenizer at sequence length 2049, following DCLM's standard configuration. The Rust pipeline produces WebDataset shards and generates the corresponding manifest file required by the DCLM training script. For each experiment, we create a dataset reference JSON to integrate seamlessly with the DCLM workflow. We do not modify tokenizer settings, sequence length, sharding configuration, or preprocessing logic. By using the official tokenize-and-shuffle implementation, we maintain identical preprocessing behavior to prior DCLM submissions and eliminate potential implementation-induced variation.

Model training is executed using DCLM's training.train entrypoint, which builds upon the OpenLM framework. All experiments follow fixed DCLM scale-specific recipes. We evaluate two compute scales: 400M-1x, which trains a 412M-parameter model on approximately 8.2B tokens, and 1B-1x, which trains a 1.4B-parameter model on approximately 28B tokens. For each scale, DCLM specifies the model architecture, number of layers, hidden size, attention heads, learning rate schedule, warmup steps, batch size, weight decay, gradient accumulation, and total number of training tokens. We use these configurations exactly as provided, without modification. In practice, we train with slightly more raw tokens than the nominal DCLM target to account for token loss during shuffling and padding, ensuring the effective training token count matches the intended compute budget. The 400M model takes around 26 H100-hours and the 1B model around 240 H100-hours.

Table 3: DCLM CORE v2 evaluation tasks used in our experiments, along with their categories.

| Task | Category | Few-shot | Description |
|---|---|---|---|
| HellaSwag | Commonsense & Reasoning | 0/10 | Sentence completion, grounded commonsense |
| CommonsenseQA | Commonsense & Reasoning | 10 | 5-choice commonsense QA |
| COPA | Commonsense & Reasoning | 0 | Causal reasoning, cause/effect |
| PIQA | Commonsense & Reasoning | 10 | Physical commonsense (2-choice) |
| Winograd | Commonsense & Reasoning | 0 | Pronoun resolution, commonsense |
| Winogrande | Commonsense & Reasoning | 0 | Large-scale Winograd-style |
| BoolQ | QA & Comprehension | 10 | Binary yes/no QA from passages |
| SQuAD (v2) | QA & Comprehension | 10 | Extractive QA; may be unanswerable |
| CoQA | QA & Comprehension | 0 | Conversational QA |
| OpenBookQA | QA & Comprehension | 0 | Multi-step reasoning + commonsense |
| ARC Easy | Science & Factual Knowledge | 10 | Grade-school science (easy), 4-choice |
| ARC Challenge | Science & Factual Knowledge | 10 | Grade-school science (hard), 4-choice |
| Jeopardy | Science & Factual Knowledge | 10 | Diverse trivia, generative |
| QA Wikidata | Science & Factual Knowledge | 10 | Big-Bench factual completions |
| MMLU | Science & Factual Knowledge | 5 | 57-subject academic QA (aggregate) |
| LSAT-AR | Science & Factual Knowledge | 3 | Analytical reasoning from LSAT |
| CS Algorithms | Symbolic & Algo Reasoning | 10 | Big-Bench: recursion, DP execution |
| Dyck Languages | Symbolic & Algo Reasoning | 10 | Big-Bench: balanced bracket completion |
| Operators | Symbolic & Algo Reasoning | 10 | Big-Bench: novel operator definitions |
| Repeat Copy Logic | Symbolic & Algo Reasoning | 10 | Big-Bench: words repeating and ordering |
| LAMBADA | Language Understanding | 0 | Last-word prediction, long context |
| Language Identification | Language Understanding | 10 | Big-Bench: identify written language |

Table 4: Licenses for existing assets used in this paper.

| Asset | Type | License | URL |
|---|---|---|---|
| Common Crawl Web Graph | Data | Custom ToU[†] | https://commoncrawl.org/web-graphs |
| cc-webgraph tools | Code | Apache 2.0 | https://github.com/commoncrawl/cc-webgraph |
| DCLM framework | Code | MIT | https://github.com/mlfoundations/dclm |
| DCLM data pool | Data | MIT[‡] | https://github.com/mlfoundations/dclm |
| WebOrganizer Corpus-200B | Data | Not specified[§] | https://huggingface.co/datasets/WebOrganizer/Corpus-200B |
| RefinedWeb filters | Code | Apache 2.0 | https://huggingface.co/datasets/tiiuae/falcon-refinedweb |
| BFF deduplication | Code | Apache 2.0[¶] | https://github.com/allenai/bff |
| cuGraph (RAPIDS) | Code | Apache 2.0 | https://github.com/rapidsai/cugraph |

## B FULL RESULTS

Pure centrality sampling at 1B scale tells a nuanced story. Bottom-$K$ outperforms Top-$K$ on average: Katz Bottom-$K$ achieves 0.405 (+0.7pp over baseline), while Katz Top-$K$ achieves only 0.392 ($-0.6$pp). This pattern holds across both centrality metrics.

At the smaller 400M scale, the pattern is weaker. The baseline (0.325) ties with Katz Bottom-$K$ (0.325) as the best average, and the other pure centrality strategies fall below it. However, the same task-level asymmetry exists: Katz Top-$K$ achieves 0.601 on BoolQ (+7.3pp over baseline) while the baseline wins on bigbench_qa_wikidata (0.444 vs. 0.387, +5.7pp).

Table 5: Best average accuracy by strategy category at 400M and 1B scale. Gains are relative to Random (top block) or Quality (bottom block). The best variant is chosen separately at each scale and differs between them; Appendix B gives every configuration at both scales. $+-$ means combining with addition for Top-$K$ and subtraction for Bottom-$K$. $\star/$ means combining with multiplication for Top-$K$ and division for Bottom-$K$.

| Strategy | Best variant | 400M | 1B |
|---|---|---|---|
| Random | — | 32.5 | 39.8 |
| Best pure centrality | Katz Bottom-$K$ | 32.5 +0.0 | 40.5 +0.7 |
| Best mixture | Katz $\alpha = 0.25$ / Betw. $\alpha = 0.5$ | 32.6 +0.1 | 41.4 +1.6 |
| Quality | — | 34.5 | 42.3 |
| Best additive | Katz $(+-)$ $\alpha = 1$ / Betw. $(+-)$ $\alpha = 0.25$ | 35.1 +0.6 | 43.7 +1.4 |
| Best multiplicative | Betw. $(\star/)$ $\alpha = 1$ / Betw. $(\star/)$ $\alpha = 0.5$ | 34.5 +0.0 | 43.8 +1.5 |

Table 6: Accuracy on 23 tasks from DCLM CORE v2 benchmark. All 1.4B models are trained on 28B tokens selected by baseline methods and our WEBGRAPHMIX method. Our methods use the betweenness centrality scores with a Top-$K$ and Bottom-$K$ mixture of 1:1. We use multiplication for combining with the quality scores, denoted as **Ours+**. Note that while our WEBGRAPHMIX is close to WebOrganizer baseline, our method is significantly cheaper and more transferable.

| Task | Random | Quality | WebOrg | WebOrg+ | Eigenvector | Ours | Ours+ |
|---|---|---|---|---|---|---|---|
| MMLU | 24.6 | 23.7 | 25.0 | 24.4 | 25.1 | **25.4** | 24.0 |
| HellaSwag (zero-shot) | 55.1 | 57.3 | 59.2 | **61.2** | 51.9 | 55.4 | 57.0 |
| Jeopardy | 15.3 | 26.8 | 19.4 | 24.6 | 15.0 | 13.0 | **27.1** |
| QA Wikidata | 58.9 | **60.2** | 58.9 | 59.0 | 58.4 | **60.2** | 59.8 |
| ARC Easy | 56.9 | 66.2 | 64.9 | **67.0** | 57.3 | 57.6 | 66.2 |
| ARC Challenge | 28.3 | **36.7** | 34.5 | **36.7** | 28.8 | 29.4 | 35.9 |
| COPA | 68.0 | **73.0** | 69.0 | 69.0 | 67.0 | 70.0 | 70.0 |
| CommonsenseQA | 24.1 | 21.9 | 22.5 | 32.3 | 29.2 | 31.8 | **32.5** |
| PIQA | 72.6 | 73.1 | 75.0 | **75.5** | 71.7 | 73.4 | 74.2 |
| OpenBookQA | 35.0 | 38.2 | 35.8 | 39.4 | 35.8 | 37.8 | **39.6** |
| LAMBADA | 55.3 | **60.3** | 51.4 | 52.2 | 51.5 | 55.0 | 58.9 |
| HellaSwag | 55.9 | 58.0 | 59.7 | **61.9** | 52.4 | 56.4 | 57.5 |
| Winograd | 71.4 | 76.9 | 73.6 | **77.3** | 70.7 | 72.2 | 76.6 |
| Winogrande | 54.1 | **58.4** | 57.9 | 56.3 | 55.6 | 57.4 | 58.1 |
| Dyck Languages | 15.8 | 21.2 | 19.1 | 22.5 | 16.8 | 23.1 | **25.4** |
| LSAT-AR | 21.3 | 20.0 | 25.2 | 22.6 | 23.9 | **27.0** | 25.2 |
| CS Algorithms | 42.3 | 42.7 | **44.9** | 42.9 | 41.3 | 41.2 | **44.9** |
| Operators | 17.1 | 19.0 | 19.5 | 19.0 | 19.0 | 18.1 | **20.0** |
| Repeat Copy Logic | 3.1 | 0 | **6.3** | 3.1 | 0 | 3.1 | 0 |
| SQuAD | 32.5 | 33.7 | 36.4 | 36.4 | 29.3 | 33.3 | **37.4** |
| CoQA | 26.0 | 29.5 | 27.8 | 29.7 | 24.2 | 26.3 | **30.8** |
| BoolQ | 58.0 | 50.8 | 56.9 | 59.9 | 60.2 | 60.2 | **62.4** |
| Language Identification | 24.5 | 25.2 | 25.2 | 25.3 | 24.7 | **25.4** | 24.8 |
| **Average** | 39.8 | 42.3 | 42.1 | 43.4 | 39.6 | 41.4 | **43.8** |

Table 7: Pure centrality sampling at 1B scale (1.4B parameters, 28B tokens). Each column selects documents whose hosts fall in the highest (Top-$K$) or lowest (Bottom-$K$) centrality stratum. Katz Bottom-$K$ achieves the highest average, suggesting peripheral web regions encode complementary capabilities at this scale.

| Task | Baseline | Betw. Top-$K$ | Betw. Bottom-$K$ | Katz Top-$K$ | Katz Bottom-$K$ |
|---|---|---|---|---|---|
| mmlu_fewshot | 0.246 | 0.245 | **0.257** | 0.248 | 0.250 |
| hellaswag_zeroshot | **0.551** | 0.535 | 0.549 | 0.524 | 0.548 |
| jeopardy | 0.153 | 0.101 | 0.145 | 0.120 | **0.165** |
| bigbench_qa_wikidata | 0.589 | 0.596 | 0.595 | 0.608 | **0.612** |
| arc_easy | 0.569 | 0.574 | **0.584** | 0.573 | 0.583 |
| arc_challenge | 0.283 | 0.288 | 0.298 | **0.305** | 0.301 |
| copa | **0.680** | 0.660 | 0.670 | 0.660 | 0.650 |
| commonsense_qa | 0.241 | 0.242 | 0.243 | 0.207 | **0.284** |
| piqa | 0.726 | 0.728 | **0.737** | 0.719 | 0.726 |
| openbook_qa | 0.350 | 0.338 | 0.358 | 0.346 | **0.370** |
| lambada_openai | **0.553** | 0.538 | 0.544 | 0.534 | 0.549 |
| hellaswag | **0.559** | 0.539 | 0.557 | 0.531 | 0.552 |
| winograd | **0.714** | 0.696 | 0.714 | 0.733 | 0.725 |
| winogrande | 0.541 | **0.578** | 0.551 | 0.554 | 0.560 |
| bigbench_dyck_languages | 0.158 | **0.213** | 0.195 | 0.165 | 0.181 |
| agi_eval_lsat_ar | 0.213 | 0.226 | **0.248** | 0.213 | 0.204 |
| bigbench_cs_algorithms | 0.423 | 0.380 | 0.426 | **0.456** | 0.414 |
| bigbench_operators | 0.171 | 0.171 | 0.171 | **0.195** | 0.181 |
| bigbench_repeat_copy_logic | **0.031** | **0.031** | 0 | 0 | **0.031** |
| squad | **0.325** | 0.294 | 0.316 | 0.286 | 0.321 |
| coqa | **0.260** | 0.249 | 0.256 | 0.240 | 0.236 |
| boolq | 0.580 | 0.602 | 0.547 | 0.548 | **0.616** |
| bigbench_language_id | 0.245 | 0.247 | 0.247 | 0.247 | **0.254** |
| **Average** | 0.398 | 0.394 | 0.400 | 0.392 | **0.405** |

Table 8: Mixture sampling at 1B scale (1.4B parameters, 28B tokens). Each column combines a specified percentage of Top-$K$ (central) documents with the remainder drawn from Bottom-$K$ (peripheral) documents. Betweenness 50% Top achieves the highest average (0.414), outperforming both the uniform baseline (0.398) and all pure sampling strategies from Table 7.

| Task | Baseline | Betw. 25% | Betw. 50% | Betw. 75% | Katz 25% | Katz 50% | Katz 75% |
|---|---|---|---|---|---|---|---|
| mmlu_fewshot | 0.246 | 0.251 | **0.254** | 0.252 | 0.233 | 0.252 | 0.241 |
| hellaswag_zeroshot | 0.551 | **0.567** | 0.554 | 0.543 | 0.542 | 0.557 | 0.547 |
| jeopardy | 0.153 | 0.153 | 0.130 | 0.137 | **0.175** | 0.156 | 0.149 |
| bigbench_qa_wikidata | 0.589 | 0.599 | 0.602 | **0.614** | 0.604 | 0.608 | 0.613 |
| arc_easy | 0.569 | 0.585 | 0.576 | 0.574 | **0.600** | 0.585 | 0.578 |
| arc_challenge | 0.283 | 0.295 | 0.294 | 0.294 | 0.303 | 0.288 | **0.304** |
| copa | 0.680 | 0.670 | 0.700 | 0.660 | 0.670 | **0.720** | 0.700 |
| commonsense_qa | 0.241 | 0.201 | 0.318 | **0.342** | 0.201 | 0.229 | 0.279 |
| piqa | 0.726 | 0.734 | 0.734 | 0.733 | 0.731 | **0.738** | 0.736 |
| openbook_qa | 0.350 | 0.352 | **0.378** | 0.358 | 0.340 | 0.336 | 0.360 |
| lambada_openai | 0.553 | **0.555** | 0.550 | 0.530 | 0.552 | 0.554 | 0.553 |
| hellaswag | 0.559 | **0.577** | 0.564 | 0.551 | 0.557 | 0.561 | 0.554 |
| winograd | 0.714 | **0.769** | 0.722 | 0.751 | 0.751 | 0.736 | 0.733 |
| winogrande | 0.541 | 0.561 | **0.574** | **0.574** | 0.557 | 0.566 | 0.571 |
| bigbench_dyck_languages | 0.158 | 0.159 | 0.231 | **0.272** | 0.183 | 0.158 | 0.174 |
| agi_eval_lsat_ar | 0.213 | 0.243 | 0.270 | 0.204 | 0.196 | 0.230 | **0.287** |
| bigbench_cs_algorithms | 0.423 | 0.448 | 0.412 | 0.417 | **0.457** | 0.442 | 0.446 |
| bigbench_operators | 0.171 | 0.157 | 0.181 | **0.190** | 0.167 | 0.176 | 0.162 |
| bigbench_repeat_copy_logic | 0.031 | 0.031 | 0.031 | 0 | 0.031 | **0.063** | 0.031 |
| squad | 0.325 | 0.328 | 0.333 | 0.322 | 0.307 | 0.329 | **0.351** |
| coqa | 0.260 | 0.268 | 0.263 | 0.255 | 0.259 | 0.244 | **0.272** |
| boolq | 0.580 | 0.563 | 0.602 | **0.610** | 0.430 | 0.593 | 0.554 |
| bigbench_language_id | 0.245 | 0.250 | 0.254 | **0.255** | 0.251 | 0.253 | 0.243 |
| **Average** | 0.398 | 0.405 | **0.414** | 0.410 | 0.395 | 0.408 | 0.410 |

Table 9: Pure centrality sampling at 400M scale (412M parameters, 8.2B tokens). The baseline (uniform sampling) ties with Katz Bottom-$K$ for the highest average (0.325); the other pure strategies fall 0.9–1.4 points below it. At this smaller scale, centrality alone does not outperform uniform sampling.

| Task | Baseline | Betw. Top-$K$ | Betw. Bottom-$K$ | Katz Top-$K$ | Katz Bottom-$K$ |
|---|---|---|---|---|---|
| mmlu_fewshot | 0.247 | **0.255** | 0.242 | 0.248 | 0.255 |
| hellaswag_zeroshot | 0.366 | 0.344 | **0.381** | 0.342 | 0.377 |
| jeopardy | 0.016 | **0.020** | 0.014 | 0.008 | 0.008 |
| bigbench_qa_wikidata | **0.444** | 0.421 | 0.392 | 0.387 | 0.380 |
| arc_easy | 0.449 | 0.452 | **0.455** | 0.448 | 0.443 |
| arc_challenge | 0.232 | 0.222 | 0.240 | **0.244** | 0.235 |
| copa | 0.620 | 0.580 | 0.570 | 0.620 | **0.670** |
| commonsense_qa | 0.268 | 0.267 | 0.224 | 0.252 | **0.367** |
| piqa | 0.670 | 0.670 | **0.680** | 0.659 | 0.675 |
| openbook_qa | 0.312 | 0.314 | 0.302 | 0.314 | **0.330** |
| lambada_openai | 0.384 | 0.354 | **0.378** | 0.357 | 0.386 |
| hellaswag | 0.368 | 0.344 | **0.379** | 0.344 | 0.381 |
| winograd | 0.612 | 0.586 | 0.608 | 0.593 | **0.626** |
| winogrande | 0.519 | **0.521** | 0.517 | 0.507 | 0.502 |
| bigbench_dyck_languages | 0.123 | 0.137 | 0.126 | **0.147** | 0.092 |
| agi_eval_lsat_ar | 0.222 | **0.252** | 0.239 | 0.222 | **0.252** |
| bigbench_cs_algorithms | 0.396 | **0.430** | 0.355 | 0.352 | 0.355 |
| bigbench_operators | **0.148** | 0.105 | 0.110 | 0.110 | 0.110 |
| bigbench_repeat_copy_logic | **0.063** | 0.031 | 0.031 | 0 | 0.031 |
| squad | **0.107** | 0.057 | 0.091 | 0.047 | 0.082 |
| coqa | 0.124 | 0.120 | **0.133** | 0.117 | 0.131 |
| boolq | 0.528 | 0.439 | **0.563** | **0.601** | 0.529 |
| bigbench_language_id | 0.253 | **0.254** | 0.250 | 0.244 | 0.246 |
| **Average** | **0.325** | 0.311 | 0.316 | 0.312 | 0.325 |

Table 10: Mixture sampling at 400M scale (412M parameters, 8.2B tokens). Katz 25% Top achieves the highest average (0.326), within seed noise of the uniform baseline (0.325), while the 1B-best Betweenness 50% mixture falls 1.0 point below it. Betweenness 75% Top shows gains on several individual tasks.

| Task | Baseline | Betw. 25% | Betw. 50% | Betw. 75% | Katz 25% | Katz 50% | Katz 75% |
|---|---|---|---|---|---|---|---|
| mmlu_fewshot | 0.247 | 0.238 | 0.244 | 0.242 | 0.241 | **0.259** | 0.235 |
| hellaswag_zeroshot | 0.366 | 0.368 | 0.359 | 0.358 | **0.370** | 0.361 | 0.352 |
| jeopardy | 0.016 | **0.021** | 0.012 | 0.021 | 0.016 | 0.016 | 0.018 |
| bigbench_qa_wikidata | **0.444** | 0.433 | **0.447** | 0.421 | 0.415 | 0.428 | 0.421 |
| arc_easy | 0.449 | 0.461 | 0.463 | **0.475** | 0.450 | 0.458 | 0.453 |
| arc_challenge | 0.232 | 0.230 | 0.235 | **0.257** | 0.255 | 0.233 | 0.244 |
| copa | **0.620** | 0.660 | 0.630 | 0.650 | **0.680** | 0.580 | 0.610 |
| commonsense_qa | 0.268 | 0.239 | 0.215 | **0.292** | 0.259 | 0.272 | 0.270 |
| piqa | 0.670 | 0.667 | **0.678** | 0.665 | 0.669 | 0.660 | 0.666 |
| openbook_qa | 0.312 | 0.306 | 0.296 | 0.302 | 0.308 | 0.310 | **0.320** |
| lambada_openai | 0.384 | 0.387 | 0.376 | 0.383 | **0.396** | 0.380 | 0.380 |
| hellaswag | 0.368 | **0.369** | 0.362 | 0.356 | 0.368 | 0.359 | 0.355 |
| winograd | 0.612 | 0.582 | 0.608 | 0.582 | **0.630** | 0.623 | 0.601 |
| winogrande | 0.519 | 0.494 | 0.502 | **0.543** | 0.500 | 0.501 | 0.515 |
| bigbench_dyck_languages | 0.123 | 0.136 | 0.152 | **0.165** | 0.149 | 0.108 | 0.161 |
| agi_eval_lsat_ar | 0.222 | 0.235 | 0.170 | 0.213 | **0.265** | 0.222 | 0.226 |
| bigbench_cs_algorithms | 0.396 | 0.429 | 0.361 | 0.393 | 0.389 | **0.436** | 0.394 |
| bigbench_operators | **0.148** | 0.105 | 0.133 | 0.124 | 0.143 | 0.124 | 0.110 |
| bigbench_repeat_copy_logic | **0.063** | 0.031 | 0 | 0.031 | 0 | 0 | 0 |
| squad | 0.107 | **0.112** | 0.071 | 0.062 | 0.089 | 0.081 | 0.083 |
| coqa | 0.124 | 0.133 | 0.127 | **0.133** | **0.137** | 0.123 | 0.120 |
| boolq | 0.528 | 0.528 | 0.559 | **0.554** | 0.524 | **0.594** | 0.546 |
| bigbench_language_id | **0.253** | **0.253** | 0.247 | 0.245 | 0.247 | 0.249 | 0.247 |
| **Average** | 0.325 | 0.323 | 0.315 | 0.325 | **0.326** | 0.321 | 0.319 |

Table 11: Additive quality–centrality combination at 400M scale (412M parameters, 8.2B tokens). Normalized quality and centrality scores are summed, and documents are ranked by the combined score. Add Katz Top-$K$ achieves the highest average (0.351), substantially outperforming both the uniform baseline (0.325) and the quality-only filter (0.345), demonstrating that structural centrality provides an additive signal on top of content-based quality scoring.

| Task | Baseline | Quality | Add Betw. Top | Add Betw. Bot. | Add Katz Top | Add Katz Bot. |
|---|---|---|---|---|---|---|
| mmlu_fewshot | 0.247 | **0.253** | 0.244 | 0.252 | 0.250 | 0.245 |
| hellaswag_zeroshot | 0.366 | 0.372 | 0.369 | 0.344 | **0.373** | 0.342 |
| jeopardy | 0.016 | **0.060** | 0.050 | 0.007 | 0.050 | 0.004 |
| bigbench_qa_wikidata | **0.444** | 0.400 | 0.396 | 0.391 | 0.399 | 0.394 |
| arc_easy | 0.449 | **0.562** | 0.546 | 0.398 | 0.546 | 0.389 |
| arc_challenge | 0.232 | 0.285 | 0.285 | 0.225 | **0.289** | 0.216 |
| copa | **0.620** | 0.600 | 0.600 | 0.560 | 0.600 | 0.610 |
| commonsense_qa | 0.268 | 0.360 | 0.202 | 0.280 | **0.368** | 0.251 |
| piqa | 0.670 | 0.663 | **0.672** | 0.653 | 0.664 | 0.668 |
| openbook_qa | 0.312 | 0.318 | 0.324 | 0.292 | **0.332** | 0.272 |
| lambada_openai | 0.384 | **0.419** | 0.413 | 0.291 | 0.417 | 0.294 |
| hellaswag | 0.368 | 0.374 | 0.369 | 0.342 | **0.376** | 0.338 |
| winograd | 0.612 | 0.619 | 0.615 | 0.560 | **0.637** | 0.549 |
| winogrande | 0.519 | 0.515 | 0.507 | **0.523** | 0.507 | **0.540** |
| bigbench_dyck_languages | 0.123 | 0.259 | 0.226 | 0.132 | **0.278** | 0.104 |
| agi_eval_lsat_ar | 0.222 | 0.248 | 0.183 | 0.191 | **0.274** | 0.191 |
| bigbench_cs_algorithms | 0.396 | 0.366 | 0.361 | 0.396 | **0.412** | 0.364 |
| bigbench_operators | **0.148** | **0.171** | 0.167 | 0.133 | 0.157 | 0.148 |
| bigbench_repeat_copy_logic | **0.063** | 0 | 0.031 | 0 | 0.031 | 0 |
| squad | 0.107 | 0.126 | 0.136 | 0.066 | **0.148** | 0.047 |
| coqa | 0.124 | 0.163 | 0.156 | 0.106 | **0.165** | 0.092 |
| boolq | 0.528 | 0.560 | 0.565 | **0.581** | 0.554 | 0.580 |
| bigbench_language_id | 0.253 | 0.247 | **0.261** | 0.250 | 0.254 | 0.251 |
| **Average** | 0.325 | 0.345 | 0.334 | 0.303 | **0.351** | 0.300 |

Table 12: Multiplicative quality–centrality combination at 400M scale (412M parameters, 8.2B tokens). Normalized quality and centrality scores are multiplied, and documents are ranked by the product. Both Multiply Betweenness Top-$K$ and Multiply Katz Top-$K$ achieve a tied best average of 0.345, matching the quality-only baseline. Bottom-$K$ variants underperform, confirming that multiplicative combination is most effective when selecting structurally central documents.

| Task | Baseline | Quality | Mult. Betw. Top | Mult. Betw. Bot. | Mult. Katz Top | Mult. Katz Bot. |
|---|---|---|---|---|---|---|
| mmlu_fewshot | 0.247 | 0.253 | 0.242 | 0.251 | **0.258** | 0.248 |
| hellaswag_zeroshot | 0.366 | **0.372** | 0.370 | 0.340 | 0.370 | 0.337 |
| jeopardy | 0.016 | **0.060** | 0.055 | 0.008 | 0.058 | 0.002 |
| bigbench_qa_wikidata | **0.444** | 0.400 | 0.420 | 0.359 | 0.402 | 0.371 |
| arc_easy | 0.449 | **0.562** | 0.544 | 0.386 | **0.567** | 0.388 |
| arc_challenge | 0.232 | **0.285** | 0.270 | 0.228 | 0.276 | 0.206 |
| copa | **0.620** | 0.600 | 0.590 | 0.560 | 0.600 | 0.590 |
| commonsense_qa | 0.268 | **0.360** | 0.229 | 0.276 | 0.304 | 0.268 |
| piqa | 0.670 | 0.663 | 0.667 | 0.647 | **0.671** | 0.657 |
| openbook_qa | 0.312 | 0.318 | 0.314 | 0.286 | **0.340** | 0.278 |
| lambada_openai | 0.384 | **0.419** | **0.425** | 0.297 | 0.419 | 0.294 |
| hellaswag | 0.368 | **0.374** | 0.370 | 0.342 | 0.368 | 0.338 |
| winograd | 0.612 | 0.619 | 0.630 | 0.590 | **0.634** | 0.601 |
| winogrande | 0.519 | 0.515 | 0.519 | **0.530** | 0.520 | 0.523 |
| bigbench_dyck_languages | 0.123 | **0.259** | 0.251 | 0.124 | 0.238 | 0.136 |
| agi_eval_lsat_ar | 0.222 | 0.248 | 0.265 | 0.204 | **0.283** | 0.178 |
| bigbench_cs_algorithms | 0.396 | 0.366 | **0.465** | 0.362 | 0.418 | 0.364 |
| bigbench_operators | 0.148 | **0.171** | 0.162 | 0.162 | 0.148 | 0.152 |
| bigbench_repeat_copy_logic | **0.063** | 0 | 0 | 0.031 | 0 | **0.063** |
| squad | 0.107 | 0.126 | 0.150 | 0.064 | **0.152** | 0.046 |
| coqa | 0.124 | **0.163** | 0.158 | 0.114 | 0.158 | 0.122 |
| boolq | 0.528 | 0.560 | 0.592 | 0.546 | 0.462 | **0.598** |
| bigbench_language_id | 0.253 | 0.247 | 0.251 | **0.258** | 0.248 | 0.251 |
| **Average** | 0.325 | **0.345** | **0.345** | 0.303 | 0.343 | 0.305 |

Table 13: Additive quality–betweenness centrality combination at 1B scale (1.4B parameters, 28B tokens). Normalized quality and betweenness centrality scores are summed, and documents are ranked by the combined score. Add Betw. 25% achieves the highest average (0.437), outperforming both the uniform baseline (0.398) and the quality-only filter (0.423), demonstrating that betweenness centrality provides an additive signal on top of content-based quality scoring.

| Task | Baseline | Quality | Add Betw. 25% | Add Betw. 50% | Add Betw. 75% | Add Betw. Bot. |
|---|---|---|---|---|---|---|
| mmlu_fewshot | 0.246 | 0.237 | 0.251 | 0.251 | 0.257 | 0.258 |
| hellaswag_zeroshot | 0.551 | 0.573 | 0.569 | 0.570 | 0.572 | 0.572 |
| jeopardy | 0.153 | 0.268 | 0.268 | **0.286** | 0.279 | 0.252 |
| bigbench_qa_wikidata | 0.589 | **0.602** | 0.588 | 0.600 | 0.592 | 0.595 |
| arc_easy | 0.569 | 0.662 | 0.675 | 0.670 | 0.665 | 0.647 |
| arc_challenge | 0.283 | 0.367 | **0.378** | 0.372 | 0.361 | 0.360 |
| copa | 0.680 | **0.730** | 0.660 | 0.710 | 0.710 | 0.710 |
| commonsense_qa | 0.241 | 0.219 | 0.310 | 0.244 | 0.313 | 0.291 |
| piqa | 0.726 | 0.731 | **0.752** | 0.740 | 0.743 | 0.740 |
| openbook_qa | 0.350 | 0.382 | 0.382 | 0.376 | 0.388 | 0.384 |
| lambada_openai | 0.553 | 0.603 | 0.598 | 0.592 | 0.598 | **0.607** |
| hellaswag | 0.559 | 0.580 | 0.577 | 0.579 | 0.579 | 0.577 |
| winograd | 0.714 | 0.769 | **0.784** | 0.747 | 0.733 | 0.766 |
| winogrande | 0.541 | 0.584 | 0.597 | 0.573 | 0.593 | 0.577 |
| bigbench_dyck_languages | 0.158 | 0.212 | 0.196 | 0.159 | 0.167 | **0.201** |
| agi_eval_lsat_ar | 0.213 | 0.200 | 0.243 | 0.243 | 0.239 | 0.217 |
| bigbench_cs_algorithms | 0.423 | 0.427 | 0.377 | **0.455** | 0.373 | 0.447 |
| bigbench_operators | 0.171 | 0.190 | 0.200 | 0.210 | 0.210 | 0.214 |
| bigbench_repeat_copy_logic | 0.031 | 0 | **0.094** | 0.031 | 0 | 0.063 |
| squad | 0.325 | 0.337 | 0.365 | 0.380 | **0.413** | 0.373 |
| coqa | 0.260 | 0.295 | 0.307 | 0.307 | 0.305 | 0.300 |
| boolq | 0.580 | 0.508 | 0.622 | 0.572 | **0.628** | 0.513 |
| bigbench_language_id | 0.245 | 0.252 | 0.249 | **0.254** | **0.255** | **0.254** |
| **Average** | 0.398 | 0.423 | **0.437** | 0.431 | 0.434 | 0.431 |

Table 14: Additive quality–Katz centrality combination at 1B scale (1.4B parameters, 28B tokens). Normalized quality and Katz centrality scores are summed, and documents are ranked by the combined score. All Katz variants achieve comparable average scores around 0.431–0.432, outperforming the uniform baseline (0.398) and exceeding the quality-only filter (0.423), demonstrating that Katz centrality provides a consistent additive signal across sampling thresholds.

| Task | Baseline | Quality | Add Katz 25% | Add Katz 75% | Add Katz Bot. | Add Katz Top |
|---|---|---|---|---|---|---|
| mmlu_fewshot | 0.246 | 0.237 | 0.248 | 0.242 | 0.244 | **0.266** |
| hellaswag_zeroshot | 0.551 | 0.573 | 0.572 | 0.569 | **0.576** | 0.573 |
| jeopardy | 0.153 | 0.268 | 0.274 | 0.268 | 0.260 | 0.253 |
| bigbench_qa_wikidata | 0.589 | **0.602** | 0.590 | 0.587 | 0.577 | 0.598 |
| arc_easy | 0.569 | 0.662 | 0.665 | **0.681** | 0.650 | 0.653 |
| arc_challenge | 0.283 | 0.367 | 0.364 | 0.362 | 0.366 | 0.348 |
| copa | 0.680 | **0.730** | 0.700 | 0.670 | 0.700 | 0.670 |
| commonsense_qa | 0.241 | 0.219 | 0.202 | 0.265 | **0.321** | 0.271 |
| piqa | 0.726 | 0.731 | 0.729 | 0.734 | 0.735 | **0.748** |
| openbook_qa | 0.350 | 0.382 | 0.392 | 0.384 | **0.394** | 0.378 |
| lambada_openai | 0.553 | 0.603 | 0.596 | 0.594 | 0.598 | 0.595 |
| hellaswag | 0.559 | 0.580 | 0.578 | 0.576 | **0.581** | 0.579 |
| winograd | 0.714 | 0.769 | 0.769 | 0.777 | 0.736 | 0.747 |
| winogrande | 0.541 | 0.584 | 0.586 | 0.581 | 0.598 | **0.602** |
| bigbench_dyck_languages | 0.158 | 0.212 | 0.161 | 0.188 | 0.179 | 0.194 |
| agi_eval_lsat_ar | 0.213 | 0.200 | **0.287** | 0.183 | 0.200 | 0.213 |
| bigbench_cs_algorithms | 0.423 | 0.427 | 0.412 | 0.421 | 0.451 | 0.447 |
| bigbench_operators | 0.171 | 0.190 | 0.214 | **0.238** | 0.186 | 0.210 |
| bigbench_repeat_copy_logic | 0.031 | 0 | 0 | 0.031 | 0.063 | 0.031 |
| squad | 0.325 | 0.337 | 0.389 | 0.391 | 0.358 | 0.384 |
| coqa | 0.260 | 0.295 | 0.309 | 0.305 | 0.300 | **0.317** |
| boolq | 0.580 | 0.508 | 0.593 | 0.610 | 0.615 | 0.580 |
| bigbench_language_id | 0.245 | 0.252 | 0.250 | 0.250 | 0.247 | 0.249 |
| **Average** | 0.398 | 0.423 | 0.430 | 0.431 | 0.432 | 0.431 |

Table 15: Multiplicative quality–betweenness centrality combination at 1B scale (1.4B parameters, 28B tokens). Normalized quality and betweenness centrality scores are multiplied, and documents are ranked by the product. Mult. Betw. 50% achieves the highest average (0.438), outperforming both the uniform baseline (0.398) and the quality-only filter (0.423). Bottom-$K$ variants remain competitive at this scale, unlike the 400M setting.

| Task | Baseline | Quality | Mult. Betw. 25% | Mult. Betw. 50% | Mult. Betw. 75% | Mult. Betw. Bot. | Mult. Betw. Top |
|---|---|---|---|---|---|---|---|
| mmlu_fewshot | 0.246 | 0.237 | 0.248 | 0.240 | 0.236 | 0.236 | 0.250 |
| hellaswag_zeroshot | 0.551 | 0.573 | 0.574 | 0.570 | **0.576** | 0.572 | 0.573 |
| jeopardy | 0.153 | 0.268 | 0.278 | 0.271 | 0.272 | 0.245 | 0.257 |
| bigbench_qa_wikidata | 0.589 | 0.602 | 0.586 | 0.598 | 0.599 | 0.597 | **0.606** |
| arc_easy | 0.569 | 0.662 | 0.672 | 0.662 | 0.662 | 0.650 | 0.647 |
| arc_challenge | 0.283 | 0.367 | **0.376** | 0.359 | 0.371 | 0.340 | 0.345 |
| copa | 0.680 | 0.730 | 0.680 | 0.700 | 0.680 | 0.680 | 0.710 |
| commonsense_qa | 0.241 | 0.219 | 0.265 | 0.325 | 0.258 | **0.342** | 0.220 |
| piqa | 0.726 | 0.731 | 0.725 | **0.742** | 0.736 | 0.743 | 0.742 |
| openbook_qa | 0.350 | 0.382 | 0.376 | 0.396 | **0.398** | 0.388 | 0.388 |
| lambada_openai | 0.553 | 0.603 | 0.592 | 0.589 | 0.595 | 0.594 | 0.592 |
| hellaswag | 0.559 | 0.580 | 0.580 | 0.575 | 0.581 | 0.577 | **0.582** |
| winograd | 0.714 | 0.769 | **0.777** | 0.766 | 0.769 | 0.744 | 0.769 |
| winogrande | 0.541 | 0.584 | 0.574 | 0.581 | **0.594** | 0.590 | 0.581 |
| bigbench_dyck_languages | 0.158 | 0.212 | 0.204 | 0.254 | 0.180 | **0.285** | 0.235 |
| agi_eval_lsat_ar | 0.213 | 0.200 | 0.239 | **0.252** | 0.239 | 0.226 | 0.243 |
| bigbench_cs_algorithms | 0.423 | 0.427 | 0.398 | **0.449** | 0.399 | 0.431 | 0.357 |
| bigbench_operators | 0.171 | 0.190 | 0.186 | 0.200 | 0.200 | 0.205 | **0.224** |
| bigbench_repeat_copy_logic | 0.031 | 0 | 0.031 | 0 | 0.031 | 0.031 | **0.063** |
| squad | 0.325 | 0.337 | 0.381 | 0.374 | **0.393** | 0.373 | 0.375 |
| coqa | 0.260 | 0.295 | 0.302 | 0.308 | **0.320** | 0.293 | 0.303 |
| boolq | 0.580 | 0.508 | 0.617 | 0.624 | 0.608 | **0.631** | 0.549 |
| bigbench_language_id | 0.245 | 0.252 | **0.255** | 0.248 | 0.248 | **0.255** | 0.245 |
| **Average** | 0.398 | 0.423 | 0.431 | **0.438** | 0.432 | 0.436 | 0.428 |

Table 16: Multiplicative quality–Katz centrality combination at 1B scale (1.4B parameters, 28B tokens). Normalized quality and Katz centrality scores are multiplied, and documents are ranked by the product. All Katz variants achieve comparable average scores around 0.430–0.432, outperforming the uniform baseline (0.398) and the quality-only filter (0.423). Bottom-$K$ variants remain competitive at this scale, unlike the 400M setting.

| Task | Baseline | Quality | Mult. Katz 25% | Mult. Katz 50% | Mult. Katz 75% | Mult. Katz Bot. | Mult. Katz Top |
|---|---|---|---|---|---|---|---|
| mmlu_fewshot | 0.246 | 0.237 | 0.249 | 0.254 | **0.260** | 0.253 | 0.264 |
| hellaswag_zeroshot | 0.551 | 0.573 | 0.572 | 0.570 | 0.572 | **0.574** | 0.572 |
| jeopardy | 0.153 | 0.268 | **0.286** | 0.273 | 0.270 | 0.263 | 0.261 |
| bigbench_qa_wikidata | 0.589 | 0.602 | 0.586 | 0.581 | 0.567 | **0.600** | 0.591 |
| arc_easy | 0.569 | 0.662 | 0.669 | 0.668 | **0.666** | 0.648 | 0.656 |
| arc_challenge | 0.283 | 0.367 | 0.369 | 0.353 | **0.378** | 0.346 | 0.356 |
| copa | 0.680 | 0.730 | 0.680 | 0.690 | **0.770** | 0.690 | 0.680 |
| commonsense_qa | 0.241 | 0.219 | 0.261 | 0.363 | **0.382** | 0.216 | 0.277 |
| piqa | 0.726 | 0.731 | **0.744** | 0.736 | 0.740 | 0.737 | 0.733 |
| openbook_qa | 0.350 | 0.382 | 0.388 | 0.396 | **0.408** | 0.384 | 0.382 |
| lambada_openai | 0.553 | 0.603 | 0.594 | 0.599 | 0.594 | **0.600** | 0.589 |
| hellaswag | 0.559 | 0.580 | 0.581 | 0.576 | 0.580 | **0.583** | 0.580 |
| winograd | 0.714 | 0.769 | 0.755 | 0.762 | 0.740 | **0.791** | 0.758 |
| winogrande | 0.541 | 0.584 | **0.601** | 0.560 | 0.578 | 0.582 | 0.572 |
| bigbench_dyck_languages | 0.158 | 0.212 | 0.195 | 0.182 | 0.209 | **0.223** | 0.175 |
| agi_eval_lsat_ar | 0.213 | 0.200 | 0.204 | 0.213 | 0.230 | **0.283** | 0.243 |
| bigbench_cs_algorithms | 0.423 | 0.427 | **0.455** | 0.355 | 0.372 | 0.386 | 0.439 |
| bigbench_operators | 0.171 | 0.190 | **0.210** | 0.181 | 0.190 | 0.190 | 0.162 |
| bigbench_repeat_copy_logic | 0.031 | 0 | 0 | 0.031 | 0.031 | 0.031 | **0.094** |
| squad | 0.325 | 0.337 | 0.391 | 0.383 | 0.365 | 0.383 | 0.376 |
| coqa | 0.260 | 0.295 | 0.313 | 0.310 | 0.316 | 0.304 | 0.300 |
| boolq | 0.580 | 0.508 | 0.593 | 0.606 | 0.464 | 0.607 | **0.625** |
| bigbench_language_id | 0.245 | 0.252 | 0.240 | 0.246 | 0.253 | 0.254 | **0.254** |
| **Average** | 0.398 | 0.423 | 0.432 | 0.430 | 0.432 | 0.432 | 0.432 |

**Best combination strategy differs between scales.** As a secondary observation, the best combination strategy differs between the two scales: at 400M, Add Katz Top is strongest, while at 1B, Multiply Betweenness 50% takes over. Multiplicative scoring suppresses documents that are low on either signal more strongly than additive scoring does, but we do not attempt to explain the reversal from two scales. At 1B, both combination strategies extract value from centrality on top of quality filtering in every configuration we ran.

# C EXAMPLE PRETRAINING DOCUMENTS

Table 17: **Host centrality reflects different types of web content.** High-betweenness hosts tend to contain broadly reusable, cross-domain patterns, whereas low-betweenness hosts more often contain specialized or long-tail information. Examples are based on representative URLs observed in the crawl. Tables 18 and 19 show crawled text from these hosts.

| Host (score) | Representative crawled content | Knowledge tendency |
|---|---|---|
| **High betweenness / central hosts** | | |
| `facebook.com` ($1.98\times10^{-1}$) | Public social/profile/community content, including posts, pages, groups, photos, and videos. | Broad social and common-sense language patterns. |
| `google.com` ($1.18\times10^{-1}$) | Aggregated search- and finance-style information, including market summaries, entities, and linked news. | Cross-domain reference and retrieval-style knowledge. |
| `youtube.com` ($1.07\times10^{-1}$) | Video pages, titles, descriptions, transcripts, comments, and media metadata. | Instructional, cultural, and explanatory content. |
| `linkedin.com` ($3.64\times10^{-2}$) | Professional profiles, companies, roles, skills, recommendations, and work histories. | Occupational and organizational knowledge. |
| `en.wikipedia.or` ($2.46\times10^{-2}$) | Encyclopedia-style articles covering scientific, historical, cultural, and technical topics. | High-coverage factual and conceptual knowledge. |
| **Low betweenness / peripheral hosts** | | |
| `hammarsdrama.co` ($4.41\times10^{-20}$) | Performing-arts and dance-film production company information. | Narrow institutional and arts-domain knowledge. |
| `ontoma.com` ($3.86\times10^{-20}$) | FinTech platform responding to a specific financial-services regulatory context. | Specialized industry and regulatory information. |
| `bluepenstrokes.` ($3.28\times10^{-20}$) | Personal creative writing, daily reflections, and artistic prose. | Individual voice and long-tail expressive content. |
| `abcsofsex-ed.or` ($1.98\times10^{-20}$) | Workshop material for a specific sex-education and public-health setting. | Specialized educational and local-program knowledge. |
| `riomardesigns.c` ($6.82\times10^{-21}$) | Small-site travel-health advice and informational content. | Niche advice and domain-specific long-tail information. |

Table 18: **Top hosts by betweenness centrality score.** The ten highest-scoring hosts from the web graph, with a representative URL snippet for each. Scores are computed over the host-level graph and reported in scientific notation.

| Host | Score | Representative Snippet |
|---|---|---|
| facebook.com | $1.98 \times 10^{-1}$ | Saving your new profile picture … Wassup guys! At last we've prepared for you some cool stuff! Tacit Fury's brand new merch! |
| google.com | $1.18 \times 10^{-1}$ | Kellogg Company (Public, NYSE:K) Watch this stock Find more results for k +1.48 (2.18%) Real-time: 3:21PM EST NYSE real-time data … |
| youtube.com | $1.07 \times 10^{-1}$ | Oval Office Underdogs: The Poet Prophet. The interactive transcript could not be loaded. Rating is available when the video has been rented. |
| instagram.com | $5.94 \times 10^{-2}$ | Enter the Lane Bryant Makeover My Mom Contest thru May 4 for your chance to win a head-to-toe Lane Bryant Makeover … |
| toptohigh.com | $4.10 \times 10^{-2}$ | Business to business marketing, commonly known as b2b marketing, refers to the interaction and marketing methods used to connect various businesses … |
| linkedin.com | $3.64 \times 10^{-2}$ | Kristi Kaylor, Beverly Hills, California. 1. 6126 LLC, 2. 2 LOVE, 3. Mblem by Mandy Moore. Recommendations: 3 people have recommended Kristi. |
| articlement.com | $3.58 \times 10^{-2}$ | Silver Mountain Express is a private shuttle & car service from Denver to Vail, Colorado that offers the perfect transportation solution … |
| gmpg.org | $3.54 \times 10^{-2}$ | XFN: Getting Started. Join the XHTML Friends Network in four easy steps. 1. Pick one or more pages to make XFN Friendly … |
| kingranks.com | $2.73 \times 10^{-2}$ | From crux gammata to swastika. What was possibly the most significant event of the 20th century, the Second World War, would not have occurred without the power of branding … |
| en.wikipedia.org | $2.46 \times 10^{-2}$ | Telluric contamination is contamination of the astronomical spectra by the Earth's atmosphere. Interference … |

Table 19: **Bottom hosts by betweenness centrality score.** The ten lowest-scoring hosts from the web graph, with a representative URL snippet for each. Scores are computed over the host-level graph and reported in scientific notation.

| Host | Score | Representative Snippet |
|---|---|---|
| hammarsdrama.com | $4.41 \times 10^{-20}$ | Hammars Drama Productions AB. We are Stockholm-based executive producers of performing art and of dance movement films. Board of Directors: Ingmar Bergman jr, Chairman. |
| easternfirst.applicantpro.com | $4.09 \times 10^{-20}$ | Eastern Industrial Supplies, Inc. 25-Jun-2018 to 24-Aug-2018 (EST). Greenville, SC, USA. Full Time. Medical, Dental, Disability, Life, 401k + Employer Match ... |
| swadharma.myshopify.com | $4.07 \times 10^{-20}$ | "Hey Julia, how are you and your belly? So do you know what you're having?" 10 reasons why I enjoyed receiving my prenatal care from my midwife. |
| ontoma.com | $3.86 \times 10^{-20}$ | An industry-willed response to the Royal Commission into Misconduct in the Financial Services Industry. New FinTech platform Ontoma set to foster cooperation ... |
| bluepenstrokes.com | $3.28 \times 10^{-20}$ | Daily Grind. Sublime requests of my creative mind overturned by demands of a cerebral strife. Shackled to cubicles, paints and brushes, paper and ink ... |
| walkertonkinsmen.ca | $2.78 \times 10^{-20}$ | Walkerton Kinsmen Raffle Draw & Novelty Casino Official Rules, Regulations and Draw Procedures. The following are the rules, regulations and draw procedures ... |
| grindstone.agency | $2.74 \times 10^{-20}$ | Harmony Honeybush. Packaging design. Brand Development. Our agency was approached to assist with the Brand Development of this new specialist company ... |
| bmscg.com | $2.09 \times 10^{-20}$ | BMS Commercial and Consulting Group. BMSCG leads you to success. We know the right way. The economic power of Asia at your service. |
| abcsofsex-ed.org | $1.98 \times 10^{-20}$ | Teaching an All-Day Workshop for 20 Teachers and Staff. Starkids Academy and Rescue Center, in Kiambu just outside the Nairobi city limits ... |
| riomardesigns.com | $6.82 \times 10^{-21}$ | Don't get sick on your next trip! Get your list. 5 Tips for Healthy Travel. There are many things you can do to avoid getting sick next time you travel. |

# D Additional Analyses

## D.1 Centrality Score Distribution

Figure 3 shows the distributions of Betweenness and Katz centrality scores at three levels of aggregation: hosts only, weighted by documents per host, and weighted by tokens per host. The latter two reflect the effective score distribution over the training corpus, since each document inherits its host's centrality score.

The two metrics exhibit very different shapes. On a log scale, Betweenness (Fig. 3a–c) is bell-shaped and roughly symmetric, with the bulk of mass between $10^{-15}$ and $10^{-5}$. A discrete spike at zero reflects hosts that lie on none of the sampled shortest paths. It is not a component effect: the symmetrized graph has a single giant connected component containing 99.95% of hosts (99.99% of documents, 99.98% of tokens), the remaining 3,057 components are negligible, and only 0.10% of the tokens selected by Betweenness Bottom-$K$ come from outside the giant component. That is far too little mass to produce the effects we report, so component membership does not confound the Top-$K$/Bottom-$K$ comparison. Katz centrality (Fig. 3d–f) is instead sharply right-skewed: most hosts cluster near the low end of the score range ($\sim 2.75 \times 10^{-4}$), with a long, sparse tail of high-scoring, highly connected hosts. Document- and token-weighting shifts mass slightly toward higher scores in both cases, since central hosts contribute more content to the corpus, but the qualitative shapes are preserved.

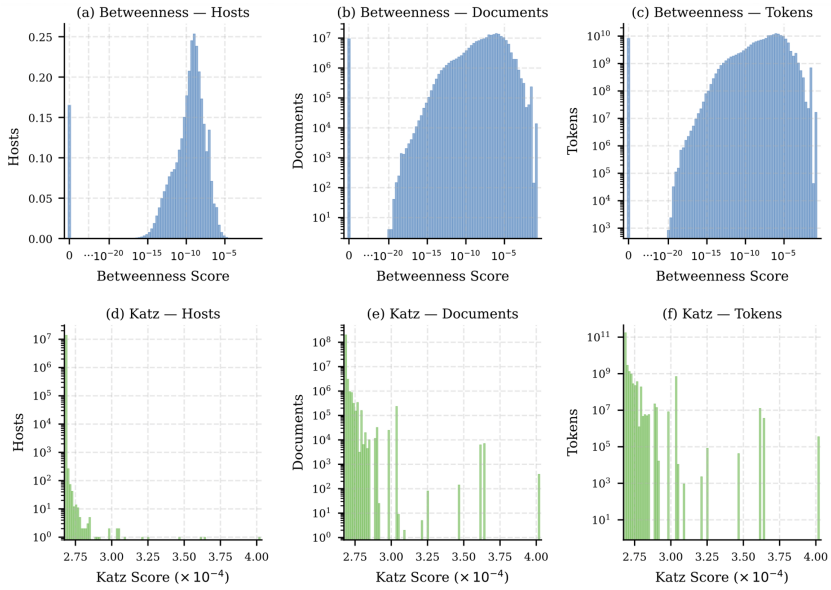


Figure 3: Histograms of Betweenness centrality scores and Katz centrality scores distribution, with respect to hosts, documents, and tokens.

### D.2 Do Central Hosts Contain More Reusable Text?

Our hypothesis predicts that text from central hosts is more broadly shared across the web than text from peripheral hosts. We test this with phrase recurrence. We sample about 1,200 documents each from top-quartile and bottom-quartile betweenness hosts, drawn from shards held out of a reference index, and measure how often each document's 5- and 10-word phrases occur elsewhere in the reference, at two reference sizes (2% and 10% of the corpus). Table 20 shows that central documents' 10-word phrases recur 1.6–1.9× as often as peripheral ones, and peripheral documents contain correspondingly more corpus-rare phrases. The direction holds at both reference sizes. This is consistent with the reusable-versus-specialized distinction, but phrase recurrence is only a proxy for reuse, and it does not show that these differences cause the downstream gains. One limitation is that matches may include templated boilerplate repeated across pages of the same host, which could inflate recurrence for large central platforms.

Table 20: Phrase recurrence for documents from central (top-quartile betweenness) and peripheral (bottom-quartile) hosts, against reference indices covering 2% and 10% of the corpus.

| | **2% reference** | | **10% reference** | |
|---|---|---|---|---|
| **Metric** | **Central** | **Peripheral** | **Central** | **Peripheral** |
| 5-gram recurs elsewhere | 41.0% | 35.7% | 52.9% | 48.0% |
| 10-gram recurs elsewhere | 12.1% | 6.3% | 17.1% | 10.7% |
| 10-gram corpus-rare ($\leq$2 occ.) | 90.7% | 95.7% | 88.1% | 94.1% |

### D.3 Centrality versus Simpler Host Properties

Table 21 reports Spearman correlations between our centrality scores and simpler host properties over all 13.9M hosts, together with the 1B accuracy of a 50/50 Top-$K$/Bottom-$K$ mixture ranked by each signal. Betweenness and Katz share 35% of their Top-$K$ hosts and 72% of their Bottom-$K$ hosts.

Table 21: Centrality versus simpler host properties. Left: Spearman $\rho$ over all 13.9M hosts. Right: 1B average accuracy of a 50/50 Top-$K$/Bottom-$K$ mixture ranked by each signal. Performance falls as the signal becomes more degree-like.

| | Spearman $\rho$ with | | | | | 50/50 mix |
|---|---|---|---|---|---|---|
| **Signal** | **In-links** | **Out-links** | **Docs/host** | **Tokens/host** | **CC harmonic** | **(1B)** |
| Betweenness | 0.735 | 0.746 | 0.306 | 0.245 | 0.449 | **41.4** |
| Katz | 0.831 | 0.839 | 0.348 | 0.294 | 0.580 | 40.8 |
| In-degree | 1.000 | – | – | – | – | 40.2 |
| Uniform | – | – | – | – | – | 39.8 |

### D.4 Quality and Linguistic Diversity

**Quality by centrality decile.** Table 22 bins documents into deciles of host betweenness and reports the mean DCLM-fasttext score and the fraction of each decile that falls in the global top 10% of quality, i.e., what a threshold filter would keep. The Spearman correlation between host centrality and document quality is $\rho = +0.018$. Peripheral documents are not disproportionately removed by quality filtering: the most peripheral decile retains 12.4% against a 10% base rate. Quality is mildly U-shaped in centrality, slightly higher at both extremes than in deciles 4–6, but the absolute variation is small.

Table 22: DCLM-fasttext quality by host-betweenness decile (1 = least central).

| **Decile** | **Mean fastText** | **% in global top-10% quality** |
|---|---|---|
| 1 | 0.0196 | 12.4 |
| 2 | 0.0171 | 10.6 |
| 3 | 0.0146 | 8.9 |
| 4 | 0.0135 | 8.3 |
| 5 | 0.0136 | 8.4 |
| 6 | 0.0130 | 8.3 |
| 7 | 0.0139 | 8.7 |
| 8 | 0.0138 | 8.7 |
| 9 | 0.0166 | 10.1 |
| 10 | 0.0245 | 14.1 |

**Is the mixture gain an implicit quality effect?** Because quality is mildly higher at both extremes, a Top-$K$/Bottom-$K$ mixture draws from bands with above-average quality. Two results argue against this explanation. First, pure Top-$K$ sampling draws from the most central deciles, which include the highest-quality band, yet scores 39.4%, below uniform sampling (39.8%); eigenvector Top-$K$ behaves the same way (39.6%). Second, on Knowledge the two strata move in opposite directions: Bottom-$K$ gains 1.3 points while Top-$K$ loses 0.4, even though the central end has the higher mean quality (0.0245 vs. 0.0196). A uniform increase in quality would shift both in the same direction.

**Linguistic diversity.** We tested whether central hosts are less linguistically diverse on 4,000 documents per group (Table 23). They are not; if anything central documents are marginally more varied, and the same pattern holds for Katz. Two caveats apply: TF-IDF dispersion is a lexical proxy rather than an embedding-based measure, and language identification cannot discriminate on an English-only corpus.

Table 23: Linguistic diversity of central and peripheral documents (betweenness).

| **Metric** | **Central** | **Peripheral** |
|---|---|---|
| Type–token ratio | 0.615 | 0.604 |
| TF-IDF dispersion | 0.767 | 0.751 |
| Mean sentence length | 18.2 | 18.2 |
| English fraction | 100% | 100% |

## E LIMITATIONS AND FUTURE WORK

**Scale.** Our experiments use DCLM's `400m-1x` and `1b-1x` settings. We have not tested whether the gains hold at 7B or beyond, which is about an order of magnitude beyond our compute budget.

**Host granularity.** Centrality is computed at the host level and inherited by every document on that host, so our signal cannot distinguish documents within a host, and it remains partly entangled with host identity and domain (§5.3). This is a tractability choice; a page-level graph could capture intra-host variation. Host-level grouping is coarse but not uninformative: MeCo (Gao et al., 2025) finds that conditioning on source URLs improves pretraining even under URL anonymization.

**Crawl policy.** Common Crawl prioritizes URLs by harmonic centrality, and our scores agree with its published ranking moderately (Spearman $\rho = 0.449$ for betweenness, $0.580$ for Katz). Our pool is not truncated: its least-central host sits at the very bottom of Common Crawl's 348M-host ranking. It is, however, denser toward the top, with the median host in Common Crawl's top 9.2%. Our results therefore describe selection within a corpus that already over-represents central hosts relative to the open web, and may transfer imperfectly to corpora built under different crawl policies.

**Centrality measures.** Our Katz configuration ($\eta = 1/d_{\max}$) closely tracks degree; Katz with $\eta$ near $1/\lambda_{\max}$, which would weight longer walks, remains unexplored. Other graph-theoretic measures, including hierarchical decompositions (k-core, k-truss), random-walk-based methods beyond Katz, and motif-based scores, are also untested.

**Combining with domain-based methods.** Combining WEBGRAPHMIX with domain-based methods such as WebOrganizer is a natural next step: graph centrality and semantic taxonomies operate on independent axes, and combining them may yield further compounding gains in the same way that combining centrality with content-based quality does.

## F BROADER IMPACT

This work introduces a graph-based framework for pretraining data selection that operates on the structural topology of the web rather than on document content. We discuss both potential positive and negative societal implications.

**Positive impacts.** WEBGRAPHMIX offers a computationally lightweight alternative to data selection methods that require training auxiliary models, running proxy evaluations, or constructing domain taxonomies. By replacing these resource-intensive steps with a one-time centrality computation (under 9 hours on at most four GPUs), our approach lowers the barrier to principled data curation, particularly for resource-constrained research groups. More broadly, improving the efficiency of pretraining data selection reduces the total compute—and therefore energy—spent on training language models, since better data can substitute for additional training steps or larger model sizes.

**Potential risks and limitations.** Graph-based selection introduces a new axis of bias that differs from content-based filtering. Web graph centrality reflects the *linking behavior* of web publishers, which is shaped by commercial incentives, language demographics, and historical web development patterns. Structurally central hosts tend to be large, English-dominant platforms (e.g., social media sites, major reference sites), while peripheral hosts include small organizations, non-English content, and niche communities. Selecting data based on centrality scores therefore risks amplifying the structural inequalities already present in the web's link topology—for example, systematically underrepresenting content from regions or languages with less interconnected web infrastructure.

Our mixture-based approach partially mitigates this concern by explicitly including peripheral documents alongside central ones, and our results show that peripheral regions contribute valuable capabilities that central regions lack. However, we do not conduct a systematic analysis of how centrality-based selection affects demographic, linguistic, or geographic representation in the resulting training data, and we encourage future work in this direction.

Finally, the centrality scores and selection scripts we plan to release are metadata annotations on an already-public corpus and do not introduce new privacy risks beyond those inherent in Common

Crawl itself. The models trained in this work are small-scale research artifacts (up to 1.4B parameters) not intended for deployment.