Hubs or Fringes? Pretraining Data Selection via Web Graph Centrality

Vedant Badoni
Department of Computer Science
Princeton University
vedant.badoni@princeton.edu

Danqi Chen
Princeton Language and Intelligence
Princeton University
danqic@cs.princeton.edu

Xinyi Wang
Princeton Language and Intelligence
Princeton University
xw2259@princeton.edu

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, graph-based 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]

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.

[1]We will release the centrality-annotated corpus and selection scripts to facilitate further research upon the acceptance of the paper.

Preprint.

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 a document's structural position in this graph may correlate with the type and transferability of knowledge it provides during pretraining. Structurally central documents—those that lie on many shortest paths or connect diverse regions—act as hubs or bridges between otherwise weakly connected communities, and are more likely to co-occur with heterogeneous contexts and expose models to reusable abstractions. In contrast, peripheral documents may encode 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 leverages web-scale structural signals to construct pretraining mixtures. Unlike prior approaches that rely on learned quality scores (Penedo et al., 2024; Li et al., 2024; Sachdeva et al., 2026) or semantic taxonomies (Wettig et al., 2025), WebGraphMix operates directly on the hyperlink graph and is fully unsupervised. We compute centrality measures over a large Common Crawl host-level graph and use these scores to partition data into structurally distinct subsets. We then construct training mixtures that emphasize (i) structurally central data, (ii) structurally peripheral data, and (iii) combinations of the two, enabling controlled investigation of how graph position affects downstream model behavior.

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

WebGraphMix differs from prior domain-based and quality-based approaches. Domain-based methods construct semantic taxonomies (e.g., topic and format categories) or optimize coarse-grained domain mixtures (e.g., arXiv, GitHub, Common Crawl) via regression or proxy training (Xie et al., 2023; Liu et al., 2025), while quality-based methods score documents by educational value or similarity to curated corpora (Penedo et al., 2024; Sachdeva et al., 2026). In contrast, WebGraphMix requires no taxonomy, classifier, or regression model—only structural signals intrinsic to the web graph—making it lightweight and directly transferable across corpora that expose hyperlink structure.

We integrate WebGraphMix into the standardized DataComp-LM (DCLM) pipeline (Li et al., 2024) and train models at 400M and 1B parameter scales with 8B and 28B tokens respectively. Centrality scores for the full Common Crawl host graph (13.9M nodes, 439.6M edges) take fewer than 9 GPU hours to compute in total and can then be reused across all downstream experiments. All training runs use identical tokenization, shuffling, and optimization procedures to isolate the effect of data selection, and we evaluate on a wide range of 23 tasks from the DCLM CORE v2 benchmark.

Our results show that graph structure provides a meaningful and complementary signal for pretraining data curation. At 1B scale, selecting documents from structurally central hosts (Katz Top-K) improves performance on Symbolic & Algorithmic Reasoning by +1.4% over uniform sampling, while selecting from peripheral hosts (Katz Bottom-K) improves Science & Factual Knowledge and Commonsense & Reasoning. These opposing effects indicate that different regions of the web graph encode distinct capability-relevant signals, and motivate mixture sampling: combining 50% central and 50% peripheral documents with betweenness centrality reaches 41.4% average across all 23 tasks, compared to 39.8% for uniform sampling. Combining the centrality signal with the DCLM-fasttext quality classifier through multiplicative & divisive scoring further improves performance to 43.8%, indicating that web graph topology captures information that is largely orthogonal to content-based quality signals.

Together, our results suggest that treating the web as a structured graph—rather than an unordered corpus—opens a new direction for studying the relationship between data distribution and model capabilities.

2 Background and Related Work

Heuristic filtering & deduplication. Existing approaches to data curation largely operate at the document level and treat documents as independent units. One class of methods applies heuristic filtering and deduplication. Rule-based filters remove boilerplate, spam, and malformed text (Raffel et al., 2020; Rae et al., 2021; Penedo et al., 2023), while deduplication techniques such as MinHash (Broder, 1997; Lee et al., 2022) and Bloom-filter-based methods (Soldaini et al., 2024) eliminate near-duplicate documents to reduce memorization. Frameworks such as DataComp-LM (DCLM) (Li et al., 2024) standardize these preprocessing steps and enable compute-controlled comparisons. While effective at improving data cleanliness and diversity, these methods do not model relationships between documents.

Document quality scoring. A second class of approaches assigns scalar quality scores to documents and selects data based on ranking. FineWeb-Edu (Penedo et al., 2024), DCLM-fasttext (Li et al., 2024), QuRating (Wettig et al., 2024), and Ask-LLM (Sachdeva et al., 2026) estimate properties such as educational value or similarity to curated corpora. Benchmark-Targeted Ranking (BETR) (Mizrahi et al., 2025) explicitly aligns pretraining data with downstream tasks by selecting documents similar to benchmark examples, achieving substantial gains under scaling-law analysis. Other approaches use perplexity (Wenzek et al., 2020), n-gram overlap (Xie et al., 2023), or attention-based signals (Hua et al., 2025) to identify useful data. Despite their diversity, these methods share a common formulation: data selection is treated as a ranking problem over independently scored documents.

Domain mixture optimization. A third line of work introduces higher-level structure by partitioning web data into domains and optimizing mixture weights. Most of the work like DoReMi (Xie et al., 2023), RegMix (Liu et al., 2025), TiKMiX (Wang et al., 2025), DoGE (Fan et al., 2024), and Aioli (Chen et al., 2025) use a coarse-grained, pre-defined domain categorization and optimize over the weights of mixtures using proxy models, regression, or influence-based techniques. To demystify the domain taxonomy of pretraining data, work like Skill-it (Chen et al., 2023), WebOrganizer (Wettig et al., 2025), Nemotron-CLIMB (Diao et al., 2026), and Group-MATES (Yu et al., 2026) defines their own data domains before optimizing the mixture, by either clustering or constructing a compact and interpretable domain taxonomy. These approaches can yield strong empirical gains, but typically require substantial computation, model training, or downstream supervision.

Underlying all these approaches is a shared assumption: documents are evaluated primarily based on their content or similarity, rather than on how they relate to one another. Even when structure is introduced (e.g., domains or clusters), it is derived from semantic similarity or learned representations, not from the native connectivity of the web.

Useful web graph structure. In contrast, the web is fundamentally a graph. Hyperlinks connect pages and hosts into a large-scale network encoding relationships such as citation, topical proximity, and information flow. Graph-based methods such as PageRank (Page et al., 1999) and HITS (Kleinberg, 1999) have long exploited this structure for ranking and retrieval, and modern web infrastructure continues to rely on graph connectivity signals (Boldi & Vigna, 2014; Baack, 2024). Recent work such as Craw4LLM (Yu et al., 2025) shows that quality-aware crawling can significantly improve crawler efficiency. Previously, over 90% of the raw data crawled from the web is discarded due to low quality. Craw4LLM leverages the quality of a webpage as the priority score of the web crawler's scheduler, replacing the standard graph-connectivity-based priority. Compared to the baseline crawler that achieves the same performance, Craw4LLM crawls only 21% of the webpages. While Craw4LLM introduces quality information to web graph exploration during crawling, we reintroduce the web graph structure information after the data is crawled and use this information for subsequent data selection. Another complementary direction leverages web metadata during training rather than for data generation. MeCo (Gao et al., 2025) incorporates URL information as a conditioning signal, improving data efficiency and enabling controllable inference. Notably, its gains persist even when URLs are anonymized, suggesting that grouping documents by source provides

useful structural information. However, such methods operate at training time rather than at the data selection stage.

To the best of our knowledge, prior work has not used graph-theoretic position as a direct signal for selecting and weighting documents within an already-crawled corpus for pretraining.

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 **centrality scores** based on each document's position in the global hyperlink network and uses these scores to guide sampling.

3.1 Web Graph Construction

We operate on the Common Crawl host-level graph[2], where each node represents a web host (e.g., `wikipedia.org`) and directed edges correspond to hyperlinks between hosts. Formally, we define a directed graph $G = (V, E)$, where $v \in V$ denotes a host and $(u, v) \in E$ indicates that host u links to host v. This host-level representation aggregates all documents from the same domain into a single node, yielding a large-scale graph with 13.9M nodes and 439.6M edges.

The raw pretraining corpus we use, Corpus-200B[3] from Wettig et al. (2025), is a pre-processed version of the `1b-1x` CommonCrawl pool from DataComps-LM (Li et al., 2024) cleaned with RefinedWeb filters (Penedo et al., 2023) and BFF deduplication (Dirk 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 $s_i = c(v)$.

3.2 Centrality Score

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

Betweenness centrality 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 $\sigma(s, t)$ 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.

Katz centrality captures recursive influence by aggregating contributions from all walks in the graph:

$$x_i = \alpha \sum_j A_{ij} x_j + \beta, \tag{2}$$

where A is the adjacency matrix, $\alpha < 1/\lambda_{\max}$ ensures convergence, β is a bias term, and i is over all nodes. This assigns higher scores to nodes connected to other influential nodes, while attenuating longer paths.

These measures capture complementary notions of structural importance: Betweenness emphasizes cross-community connectivity, while Katz centrality reflects global influence.

PageRank (Page et al., 1999) is a specific variant of eigenvector centrality. Prior work has shown that eigenvector centrality can be used in place of PageRank in directed networks with lower computational cost while preserving rank correlation (Chandrashekhar et al., 2022). We ran ablations using eigenvector centrality but found it did not yield improvements over the baseline. We focus on Betweenness and Katz centrality in the main paper as they capture distinct and complementary notions of structural importance—bridging versus weighted influence.

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

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

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[4]. In practice, computing Katz centrality for the full Common Crawl host graph took us < 3 hours on one H100 GPU and computing Betweenness centrality 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. 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%, 50%, 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.

Mixed Sampling. To test whether central and peripheral regions provide complementary signals, we construct mixtures combining both strata, i.e. α% Top-K + $(100-\alpha)$% Bottom-K, with $\alpha \in \{0, 25, 50, 75, 100\}$. Documents are sampled proportionally until the token budget is reached.

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 identify text resembling a reference corpus mainly generated by GPT-4. We normalize both the centrality scores and the quality scores by:

$$\hat{s}_i = \exp(s_i - \max_j s_j). \tag{3}$$

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. Documents are ranked by the corresponding combined score and selected under the same token budget. These strategies allow us to test whether graph structure provides a signal complementary to document quality.

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

4 Experiments

4.1 Experimental Setup

All experiments are conducted using the official DataComp-LM (DCLM) framework, which provides standardized data pools, fixed model architectures, compute-optimal token budgets, and a fully reproducible training and evaluation pipeline. 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. We mainly report 1B model results in the main paper as they are more significant. Full 400M model results can be found in Appendix B.

We report task-level and average accuracy on DCLM CORE v2 benchmark (Li et al., 2024), which consists of 23 tasks. As described in Table 5 in Appendix A, the eval tasks are classified into 5 categories[5]: Commonsense & Reasoning, QA & Comprehension, Science & Factual Knowledge, Symbolic & Algo Reasoning, and Language Understanding. In the following paper, we will use Commonsense, Comprehension, Knowledge, Reasoning, and Language as abbreviations. We also look into each category of tasks to get a better understanding of the distinct effect of top-K (central) and bottom-K (peripheral) sampling.

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 up to token budget; **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; **PageRank**: select documents with top K eigenvector centrality which can be used in place of the classical PageRank algorithm (Chandrashekhar et al., 2022).

4.2 Main Results

Table 1 shows the overall benchmark performance comparison between our best methods and baselines, averaged over task categories. When mixing Top-K and Bottom-K documents at a token ratio of 1:1 ranked by Betweenness centrality score, our method improves upon the random selection baseline by 1.6% on average and improves upon the top-K quality score selection baseline by 1.5% on average. Note that using quality score alone would improve upon the random selection baseline by 2.5%, so our method combining with quality score improves upon random selection baseline by 4% in total. WebGraphMix improves over the random selection baseline on all task categories, while WebGraphMix + 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+ slightly outperforms WebOrganizer+ on overall average while requiring no proxy training, no labeled targets, and no benchmark-specific tuning.

The effectiveness of WebGraphMix scales with model size. The gain from the best mixture strategy over baseline grows from 0.1% at 400M parameters to 1.6% at 1B parameters, and the gain from combining quality scores grows from 0.6% at 400M parameters to 1.5% at 1B parameters[6]. This is consistent with the scaling behavior observed in other data selection work (Mizrahi et al., 2025; Yu et al., 2026) and suggests that our method may provide larger gains at larger scales.

4.3 Structural Position Differentially Affects Capability Categories

Table 2 breaks down performance by capability category for Top-K (central) and Bottom-K (peripheral) sampling strategies at the 1B scale. The results show that the effect of structural position is highly capability-dependent, and that central and peripheral regions of the web encode different types of useful information.

[5] As marked in meta data: `https://github.com/mlfoundations/dclm/blob/main/eval/eval_meta_data.csv`

[6] See Table 11, Table 12, and Table 13 in Appendix B for full 400M scale results.

Table 1: Accuracy on DCLM CORE v2 benchmark at 1B scale, averaged by task category. **WebGraphMix** uses betweenness centrality with a 50/50 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	Commonsense	Comprehension	Knowledge	Reasoning	Language	Avg
Random	57.3	37.9	34.2	19.0	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
PageRank	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**

Table 2: Average accuracy across different task categories for Top-K and Bottom-K sampling at 1B scale. **Betw.** denotes Betweenness centrality scores. **Katz** denotes Katz centrality scores.

Method	Commonsense	Comprehension	Knowledge	Reasoning	Language
Random	57.3	37.9	34.2	19.0	39.9
Betw. Top-K	56.7 -0.6	37.1 -0.8	33.9 -0.3	19.9 +0.9	39.2 -0.7
Betw. Bottom-K	57.4 +0.1	36.9 -1.0	35.4 +1.2	19.8 +0.8	39.5 -0.4
Katz Top-K	56.1 -1.2	35.5 -2.4	34.4 +0.2	20.4 +1.4	39.1 -0.8
Katz Bottom-K	57.8 +0.5	38.6 +0.7	35.3 +1.1	20.2 +1.2	40.2 +0.3

Bottom-K sampling consistently improves factual and commonsense knowledge. The clearest and most consistent 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 35.4% (+1.2%), while Katz Bottom-K reaches 35.3% (+1.1%). In contrast, Betweenness Top-K slightly hurts performance (-0.3%).

A similar trend appears for *Commonsense*. Katz Bottom-K achieves the best score (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 (−0.6%).

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 from both Top-K and Bottom-K sampling. Unlike the knowledge-oriented categories, *Reasoning* improves under all centrality-based sampling strategies. Katz Top-K achieves the strongest result (20.4%, +1.4%), followed closely by Katz Bottom-K (+1.2%). Betweenness Top-K and Bottom-K produce similar gains (+0.9% and +0.8% respectively).

This indicates that reasoning tasks benefit from structural selection in general. However, the relatively stronger gains from Top-K sampling methods suggest that highly influential hosts may contain more structured or procedural content useful for these tasks.

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.

These results suggest that aggressive concentration on structurally central hosts may reduce linguistic diversity or contextual variability, which are important for comprehension-oriented tasks.

Centrality metric matters. The two centrality measures also behave differently. Katz centrality generally produces larger, less stable positive and negative shifts than Betweenness centrality, such as the strongest gains on *Reasoning* (+1.4%) but also the largest degradation on *Comprehension* (-2.4%), suggesting that recursive influence captures a stronger structural signal than shortest-path bridging.

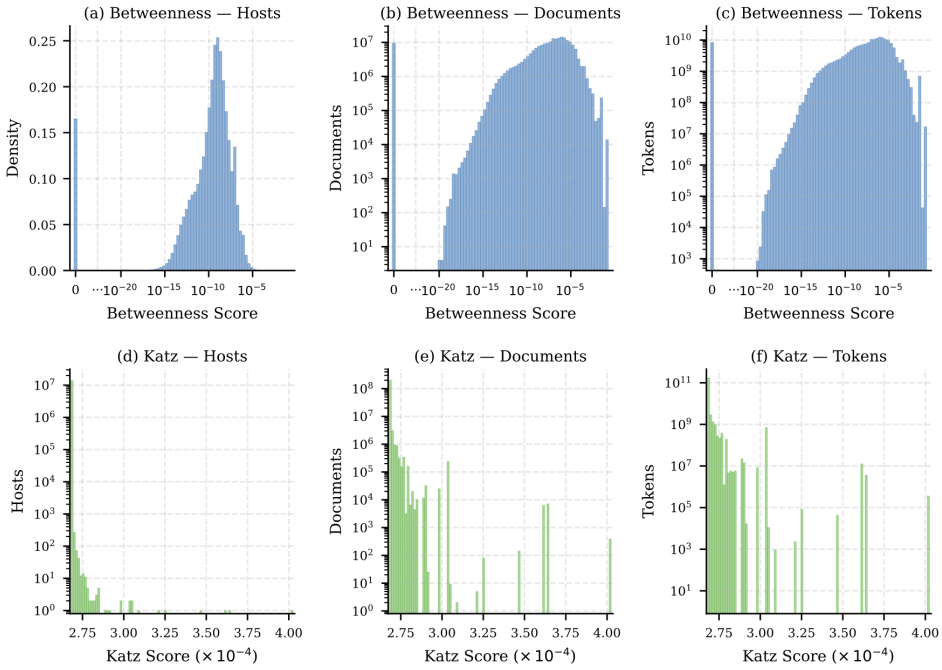

Figure 2: Histograms of Betweenness centrality scores and Katz centrality scores distribution, with respect to hosts, documents, and tokens.

Table 3: Average accuracy of mixture sampling with Betweenness centrality score and Katz centrality score at 1B scale, varying proportion of Top-K and Bottom-K tokens.

Mixture Ratio	Random	Betweenness	Katz
0% Top + 100% Bottom	39.8	40.0 +0.2	40.5 +0.7
25% Top + 75% Bottom	39.8	40.5 +0.7	39.5 -0.3
50% Top + 50% Bottom	39.8	**41.4** +1.6	40.8 +1.0
75% Top + 25% Bottom	39.8	41.0 +1.2	**41.0** +1.2
100% Top + 0% Bottom	39.8	39.4 -0.4	39.2 -0.6

4.4 Centrality Score Distribution

Figure 2 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. 2a–c) is bell-shaped and roughly symmetric, with the bulk of mass between 10^{-15} and 10^{-5}. A discrete spike at zero reflects a structural artifact of the Common Crawl host graph: it consists of roughly a dozen weakly connected components, and hosts in the smaller components receive near-zero betweenness because the shortest paths through them are bounded by their component size. Katz centrality (Fig. 2d–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 hosts that are recursively linked to other influential 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.

4.5 Mixture Sampling

We investigate the effect of mixing top-K and bottom-K sampling by varying the proportion of top-K and bottom-K documents in the sampled data. We find a mixture at around 1:1 yields the strongest performance. Table 3 summarizes the 23-task averages across mixture ratios and centrality metrics.

Table 4: Average accuracy of mixture sampling with Betweenness centrality score and Katz centrality score combined with quality score at 1B scale, varying proportion of Top-K and Bottom-K tokens. $+-$ means combining with addition for Top-K and subtraction for bottom-K. $\times/$ means combining with multiplication for Top-K and division for bottom-K.

Mixture Ratio	**Quality**	**Betw.** $(+-)$	**Betw.** $(\times/)$	**Katz** $(+-)$	**Katz** $(\times/)$
0% Top + 100% Bottom	42.3	43.1 +0.8	43.6 +1.3	43.2 +0.9	43.2 +0.9
25% Top + 75% Bottom	42.3	43.7 +1.4	43.1 +0.8	43.0 +0.7	43.2 +0.9
50% Top + 50% Bottom	42.3	43.1 +0.8	**43.8** +1.5	43.1 +0.8	43.0 +0.7
75% Top + 25% Bottom	42.3	43.4 +1.1	43.2 +0.9	43.1 +0.8	43.2 +0.9
100% Top + 0% Bottom	42.3	43.2 +0.9	42.8 +0.5	43.1 +0.8	43.2 +0.9

Mixing outperforms pure sampling. Neither pure top-K nor pure bottom-K achieves the gain of 50/50 mixture. This confirms our central hypothesis: central and peripheral web regions encode complementary capabilities, and balancing them yields better data mixture than either extreme alone.

The optimal ratio is roughly balanced. Across betweenness mixtures, the 50/50 split outperforms both 25/75 (40.5%) and 75/25 (41.0%). This suggests that neither central nor peripheral documents should dominate—the best pretraining data draws roughly equally from both structural extremes of the web graph.

Betweenness mixtures are consistently stronger than or equal to Katz mixtures at the 1B scale, with the gap largest at the 50/50 split (+0.6%). There is a clear non-monotonic pattern for betweenness: performance peaks at 50/50 and declines on either side. This inverted-U shape supports the complementarity hypothesis—too much of either extreme hurts. Katz mixtures also support this trend, with performance improving as the proportion of central documents increases (39.5% to 40.8% to 41.0%), but then decreasing to 39.2% when there are too many (Katz Top-K). This may reflect the different nature of Katz centrality, which emphasizes local connectivity rather than global bridging.

At 400M, mixture improvements are smaller but present. Katz 25% Top improves 0.1% over the baseline while Betweenness 75% Top makes gains on specific tasks like ARC Easy (2.6%), Winogrande (2.4%), and ARC Challenge (2.5%). This weaker signal is expected: smaller models have less capacity to leverage the complementary information from different web regions.

4.6 Combining with Quality Scores

Table 4 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 one of the 18 reported configurations 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.

5 Conclusion

We introduced WebGraphMix, a lightweight pretraining data selection framework that uses structural position in the Common Crawl web graph as a signal for sampling documents. Our results show that different regions of the web graph encode complementary capabilities: structurally central hosts improve symbolic and procedural reasoning more, while peripheral hosts improve commonsense and factual knowledge more. Mixing these regions outperforms either extreme alone, and combining centrality with quality-based filtering yields further gains. Unlike prior data selection methods that require auxiliary model training, influence estimation, or domain taxonomy construction, WebGraphMix computes centrality scores once over publicly available web graph using standard graph algorithms, requiring less than 9 GPU-hours. The resulting signal is lightweight, reusable, and complementary to existing content-based approaches, suggesting that web graph topology is a promising new axis for pretraining data curation.

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.

Stefan Baack. A critical analysis of the largest source for generative ai training data: Common crawl. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, pp. 2199–2208, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704505. doi: 10.1145/3630106.3659033. URL https://doi.org/10.1145/3630106.3659033.

Paolo Boldi and Sebastiano Vigna. Axioms for centrality. *Internet Mathematics*, 10(3-4):222–262, 2014.

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.

Suvarna Saumya Chandrashekhar, Mashrin Srivastava, B Jaganathan, and Pankaj Shukla. Pagerank algorithm using eigenvector centrality–new approach. *arXiv preprint arXiv:2201.05469*, 2022.

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.

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.

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

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.

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.

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.

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*, 2023. URL https://openreview.net/forum?id=lXuByUeHhd.

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 20 hours on 4 H100 GPUs while the 1B model takes around 90 hours on 4 H100 GPUs.

Table 5: 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 6: 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 (+0.4pp over baseline), while Katz Top-K achieves only 0.392 (-0.9pp). 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. Pure centrality strategies show less consistent improvement, likely because the 400M model has less capacity to exploit the structural signal. 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 7: 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+	PageRank	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 8: 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 9: 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 8.

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 10: Pure centrality sampling at 400M scale (412M parameters, 8.2B tokens). The baseline (uniform sampling) achieves the highest average (0.325), with pure centrality strategies performing comparably. At this smaller scale, the signal from centrality alone does not consistently 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 11: Mixture sampling at 400M scale (412M parameters, 8.2B tokens). Katz 25% Top achieves the highest average (0.326), marginally outperforming the uniform baseline (0.325). Betweenness 75% Top also shows gains on several individual tasks, indicating that the complementary signal from mixing central and peripheral documents is present even at smaller model scales.

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 12: 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 13: 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 14: Additive quality–centrality combination at 1B scale (1.4B parameters, 28B tokens). Normalized quality and centrality scores are summed, and documents are ranked by the combined score. Add Betw. 25% Top achieves the highest average (0.437), outperforming both the uniform baseline (0.398) and the quality-only filter (0.423), demonstrating that structural 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.	Add Katz 25%	Add Katz 75%	Add Katz Bot.	Add Katz Top
mmlu_fewshot	0.246	0.237	0.251	0.251	0.257	0.258	0.248	0.242	0.244	**0.266**
hellaswag_zeroshot	0.551	0.573	0.569	0.570	0.572	0.572	0.572	0.569	**0.576**	0.573
jeopardy	0.153	0.268	0.268	**0.286**	0.279	0.252	0.274	0.268	0.260	0.253
bigbench_qa_wikidata	0.589	**0.602**	0.588	0.600	0.592	0.595	0.590	0.587	0.577	0.598
arc_easy	0.569	0.662	0.675	0.670	0.665	0.647	0.665	**0.681**	0.650	0.653
arc_challenge	0.283	0.367	**0.378**	0.372	0.361	0.360	0.364	0.362	0.366	0.348
copa	0.680	**0.730**	0.660	0.710	0.710	0.710	0.700	0.670	0.700	0.670
commonsense_qa	0.241	0.219	0.310	0.244	0.313	0.291	0.202	0.265	**0.321**	0.271
piqa	0.726	0.731	**0.752**	0.740	0.743	0.740	0.729	0.734	0.735	0.748
openbook_qa	0.350	0.382	0.382	0.376	0.388	0.384	0.392	0.384	**0.394**	0.378
lambada_openai	0.553	0.603	0.598	0.592	0.598	**0.607**	0.596	0.594	0.598	0.595
hellaswag	0.559	0.580	0.577	0.579	0.579	0.577	0.578	0.576	**0.581**	0.579
winograd	0.714	0.769	**0.784**	0.747	0.733	0.766	0.769	0.777	0.736	0.747
winogrande	0.541	0.584	0.597	0.573	0.593	0.577	0.586	0.581	0.598	**0.602**
bigbench_dyck_languages	0.158	0.212	0.196	0.159	0.167	**0.201**	0.161	0.188	0.179	0.194
agi_eval_lsat_ar	0.213	0.200	0.243	0.243	0.239	0.217	**0.287**	0.183	0.200	0.213
bigbench_cs_algorithms	0.423	0.427	0.377	**0.455**	0.373	0.447	0.412	0.421	0.451	0.447
bigbench_operators	0.171	0.190	0.200	0.210	0.210	0.214	0.214	**0.238**	0.186	0.210
bigbench_repeat_copy_logic	0.031	0	**0.094**	0.031	0	0.063	0	0.031	0.063	0.031
squad	0.325	0.337	0.365	0.380	**0.413**	0.373	0.389	0.391	0.358	0.384
coqa	0.260	0.295	0.307	0.307	0.305	0.300	0.309	0.305	0.300	**0.317**
boolq	0.580	0.508	0.622	0.572	**0.628**	0.513	0.593	0.610	0.615	0.580
bigbench_language_id	0.245	0.252	0.249	**0.254**	**0.255**	**0.254**	0.250	0.250	0.247	0.249
Average	0.398	0.423	**0.437**	0.431	0.434	0.431	0.430	0.431	0.432	0.431

Table 15: Multiplicative quality–centrality combination at 1B scale (1.4B parameters, 28B tokens). Normalized quality and 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	Mult. Katz 25%	Mult. Katz 50%	Mult. Katz 75%	Mult. Katz Bot.	Mult. Katz Top
mmlu_fewshot	0.246	0.237	0.248	0.240	0.236	0.236	0.250	0.249	0.254	**0.260**	0.253	0.264
hellaswag_zeroshot	0.551	0.573	0.574	0.570	**0.576**	0.572	0.573	0.572	0.570	0.572	0.574	0.572
jeopardy	0.153	0.268	0.278	0.271	0.272	0.245	0.257	**0.286**	0.273	0.270	0.263	0.261
bigbench_qa_wikidata	0.589	0.602	0.586	0.598	0.599	0.597	**0.606**	0.586	0.581	0.567	0.600	0.591
arc_easy	0.569	0.662	0.672	0.662	0.662	0.650	0.647	0.669	0.668	0.666	0.648	0.656
arc_challenge	0.283	0.367	**0.376**	0.359	0.371	0.340	0.345	0.369	0.353	**0.378**	0.346	0.356
copa	0.680	0.730	0.680	0.700	0.680	0.680	0.710	0.680	0.690	**0.770**	0.690	0.680
commonsense_qa	0.241	0.219	0.265	0.325	0.258	0.342	0.220	0.261	0.363	**0.382**	0.216	0.277
piqa	0.726	0.731	0.725	0.742	0.736	0.743	0.742	0.744	0.736	0.740	0.737	0.733
openbook_qa	0.350	0.382	0.376	0.396	**0.398**	0.388	0.388	0.388	0.396	**0.408**	0.384	0.382
lambada_openai	0.553	0.603	0.592	0.589	0.595	0.594	0.592	0.594	0.599	0.594	**0.600**	0.589
hellaswag	0.559	0.580	0.580	0.575	0.581	0.577	0.582	0.581	0.576	0.580	**0.583**	0.580
winograd	0.714	0.769	0.777	0.766	0.769	0.744	0.769	0.755	0.762	0.740	**0.791**	0.758
winogrande	0.541	0.584	0.574	0.581	0.594	0.590	0.581	**0.601**	0.560	0.578	0.582	0.572
bigbench_dyck_languages	0.158	0.212	0.204	0.254	0.180	**0.285**	0.235	0.195	0.182	0.209	0.223	0.175
agi_eval_lsat_ar	0.213	0.200	0.239	0.252	0.239	0.226	0.243	0.204	0.213	0.230	**0.283**	0.243
bigbench_cs_algorithms	0.423	0.427	0.398	0.449	0.399	0.431	0.357	**0.455**	0.355	0.372	0.386	0.439
bigbench_operators	0.171	0.190	0.186	0.200	0.200	0.205	**0.224**	0.210	0.181	0.190	0.190	0.162
bigbench_repeat_copy_logic	0.031	0	0.031	0	0.031	0.031	0.063	0	0.031	0.031	0.031	**0.094**
squad	0.325	0.337	0.381	0.374	**0.393**	0.373	0.375	0.391	0.383	0.365	0.383	0.376
coqa	0.260	0.295	0.302	0.308	**0.320**	0.293	0.303	0.313	0.310	0.316	0.304	0.300
boolq	0.580	0.508	0.617	0.624	0.608	**0.631**	0.549	0.593	0.606	0.464	0.607	0.625
bigbench_language_id	0.245	0.252	**0.255**	0.248	0.248	**0.255**	0.245	0.240	0.246	0.253	0.254	0.254
Average	0.398	0.423	0.431	**0.438**	0.432	0.436	0.428	0.432	0.430	0.432	0.432	0.432

Best combination strategy shifts with scale. As a secondary observation, we note that the best combination strategy reverses between the two scales: at 400M, Add Katz Top was strongest, while at 1B, Multiply Betweenness 50% takes over. This may partially be attributed to the differing selectivity of the two strategies. Multiplicative scoring strongly suppresses documents that are low on either signal, while additive scoring is more permissive. At 400M, where the model has limited capacity, the broader signal from additive combination appears more useful; at 1B, the sharper selectivity of multiplicative combination yields better results. While interesting, this reversal is less practically important than the broader finding that *both* combination strategies, in nearly all configurations, extract real value from centrality on top of quality filtering.

C Example Pretraining Documents

Table 16: **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	1.98×10^{-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	1.18×10^{-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	1.07×10^{-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	5.94×10^{-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	4.10×10^{-2}	Business to business marketing, commonly known as b2b marketing, refers to the interaction and marketing methods used to connect various businesses …
linkedin.com	3.64×10^{-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	3.58×10^{-2}	Silver Mountain Express is a private shuttle & car service from Denver to Vail, Colorado that offers the perfect transportation solution …
gmpg.org	3.54×10^{-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	2.73×10^{-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	2.46×10^{-2}	Telluric contamination is contamination of the astronomical spectra by the Earth's atmosphere. Interference …

Table 17: **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	4.41×10^{-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	4.09×10^{-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	4.07×10^{-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	3.86×10^{-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	3.28×10^{-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	2.78×10^{-20}	Walkerton Kinsmen Raffle Draw & Novelty Casino Official Rules, Regulations and Draw Procedures. The following are the rules, regulations and draw procedures ...
grindstone.agency	2.74×10^{-20}	Harmony Honeybush. Packaging design. Brand Development. Our agency was approached to assist with the Brand Development of this new specialist company ...
bmscg.com	2.09×10^{-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	1.98×10^{-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	6.82×10^{-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 Limitations and future work

Our experiments are conducted at 400M and 1B parameter scales with 8B and 28B training tokens respectively, following the DCLM 1b-1x reference setting. The scaling pattern we observe—gains that grow with model size—suggests that further improvements may be achievable at larger scales, but verifying this requires substantially more compute. Our centrality scores are also computed at the host level and inherited by all documents from a given host; a finer-grained page-level graph could capture intra-host structural variation that is currently averaged out. We focused on betweenness and Katz centrality because they capture distinct notions of structural importance (cross-community bridging vs. recursive influence), but other graph-theoretic measures—including hierarchical decomposition (k-core, k-truss), random-walk-based methods beyond Katz, and motif-based scores—remain unexplored. Finally, 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.

E 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 (fewer than 9 GPU-hours total), 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.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) clearly state the three main claims: (1) central and peripheral web regions encode complementary capabilities, (2) a 1:1 mixture achieves 41.4% vs. 39.8% for uniform sampling, and (3) combining with quality scores reaches 43.8%. These are directly supported by the experimental results in Tables 1, 3, and 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix D discusses limitations including the restricted scale of experiments (400M and 1B parameters), host-level granularity that averages out intra-host variation, the limited set of centrality measures explored, and the unexplored combination with domain-based methods such as WebOrganizer.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include formal theoretical results or proofs. The centrality measures used (Betweenness, Katz) are standard graph-theoretic definitions cited from prior work.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: 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.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper states that the centrality-annotated corpus and selection scripts will be released upon acceptance (footnote 1). At submission time, the code and data are not yet publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 specifies model architectures (412M and 1.4B parameters), token budgets (8.2B and 28B), the DCLM framework and its fixed recipes, and the evaluation suite (23 tasks from DCLM CORE v2). Appendix A provides additional details on tokenization, shuffling, and dataset preparation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The paper does not report error bars, confidence intervals, or multiple-seed runs. Each configuration is trained once, which is standard practice in large-scale pretraining experiments due to computational cost.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 3.2 reports that computing Betweenness centrality took $<$3 hours on one H100 GPU and Katz centrality took $<$6 hours on 4 H100 GPUs. Appendix A specifys that 400M-1x and 1B-1x following DCLM's compute-controlled recipes take around 20 hours and 90 hours on 4 H100 GPUs respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available web graph data from Common Crawl and an existing preprocessed corpus. The work conforms to the NeurIPS Code of Ethics. No human subjects, private data, or ethically sensitive applications are involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both possible positive and negative impact of our work in Appendix E.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release pretrained language models or new datasets at submission time. The centrality scores and selection scripts planned for release pose minimal misuse risk, as they are metadata annotations on an already-public corpus.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites all major assets used: Common Crawl web graph (with URL), the DCLM framework and data pool (Li et al., 2024), the WebOrganizer Corpus-200B (Wettig et al. (2025), with HuggingFace URL), and cuGraph (with GitHub URL). License details are shown in Table 6 in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces centrality-annotated host scores and selection scripts as new assets, to be released upon acceptance (footnote 1).

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: 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 only LLMs for writing assistance when drafting the script.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.