Hubs or Fringes?

Pretraining Data Selection via Web Graph Centrality

Vedant Badoni, Danqi Chen, Xinyi Wang

Princeton Language and Intelligence

PLI Lunch Talk • May 2026

LLM Scaling: The Third Axis

2020

Axis 1: Model Size

Kaplan et al., 2020

Power-law relationship between model parameters and loss

"Larger models are more sample-efficient"

2022

Axis 2: Data Size

Hoffmann et al., 2022 (Chinchilla)

Model size & data should scale equally: ~20 tokens/param

70B model + 1.4T tokens outperformed 280B Gopher

2024–

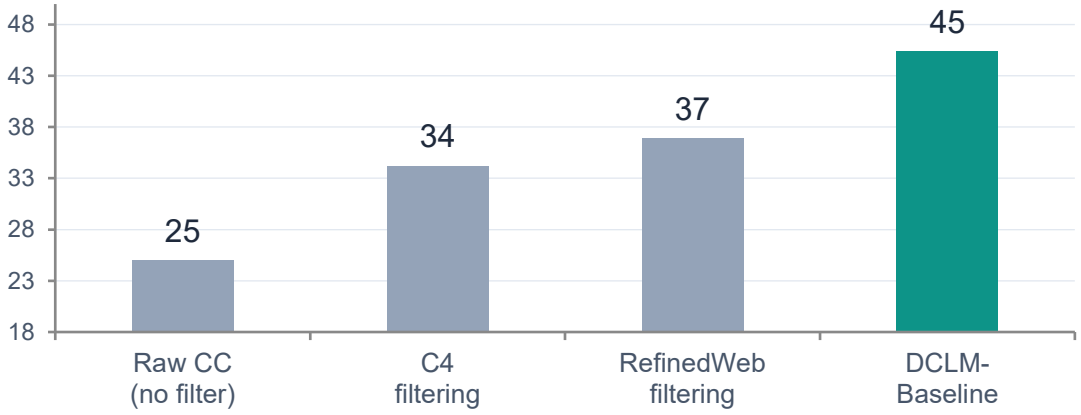
Axis 3: Data Quality

DCLM, FineWeb-Edu, Phi, QuRating, ...

Same compute, same tokens — dramatically different outcomes based on data selection

Data Quality Dramatically Shifts Performance

DCLM CORE scores at 7B scale (Li et al., 2024)



Web dump, no processing → Heuristic rules: length, lang, dedup → + boilerplate removal, quality heuristics → **fasttext classifier trained on GPT-4 text**

DCLM paper Table 2 (7B-1x). Raw CC estimated from 1B-scale results.

Key Takeaways

+20 CORE gain from raw CC to DCLM-Baseline

+8.5 over RefinedWeb, the best heuristic-only filtering

The Data Selection Problem

Heuristic Filtering

Rule-based filters, dedup
(RefinedWeb, DCLM)

Quality Scoring

Classify doc quality
(FineWeb-Edu, Ask-LLM)

Domain Mixtures

Optimize domain weights
(DoReMi, RegMix, WebOrg)

What's missing?

All these methods treat documents as independent units.

They ignore **how documents relate to each other** — the web is a graph, but we throw away its structure after crawling.

The Web is a Graph

Our Hypothesis

A document's structural position in the web graph correlates with the type of knowledge it provides.

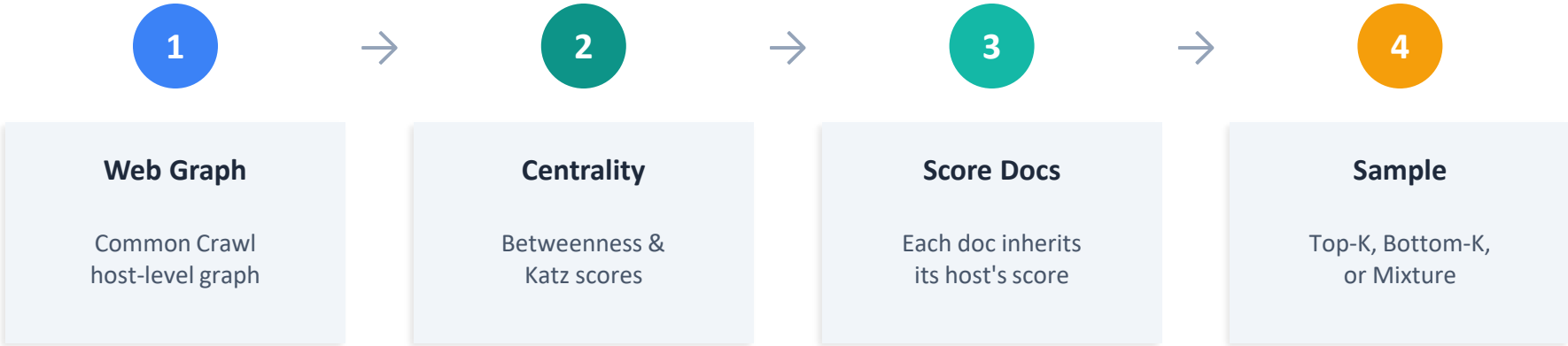
Central hosts (hubs/bridges) → reusable abstractions, cross-domain patterns

Peripheral hosts (fringes) → specialized, long-tail knowledge

No classifiers, no labels, no proxy models — just graph structure

Common Crawl host-level graph:
13.9M nodes, 439.6M edges

WebGraphMix: Method Overview



Efficiency

< 6 hrs
Betweenness (4× H100)

< 3 hrs
Katz centrality (1× H100)

One-time
Compute once, reuse forever

Centrality Measures

Betweenness Centrality

How often a node lies on shortest paths between other nodes

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

Captures cross-community bridging

Katz Centrality

Recursive influence: aggregates contributions from all walks

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

Captures global weighted influence

PageRank (eigenvector centrality) was also tested but did not improve over baseline.

Centrality-Guided Sampling

Top-K (Central)

Select docs from hosts in the top percentile of centrality

Broadly reusable, cross-domain patterns

Bottom-K (Peripheral)

Select docs from hosts in the lowest percentile of centrality

Specialized, long-tail knowledge

Mixed Sampling

α% Top-K + (100−α)% Bottom-K
$\alpha \in \{0, 25, 50, 75, 100\}$

Best of both: complementary signals

Combining Structural & Quality Signals

Quality Baseline: DCLM-fasttext

A bigram fasttext classifier trained to distinguish *curated reference text (mostly GPT-4 outputs)* from raw web text.

Each document gets a scalar quality score
→ rank & select top-K by score up to token budget

Captures: educational value, writing quality, coherence

How We Combine the Two Signals

Step 1: Normalize both scores

$$\hat{s}_i = \exp\left(s_i - \max_j s_j\right)$$

Step 2: Combine per document

Rank by combined score → select up to token budget

Combination Strategies

For Top-K (select highest combined score)

Additive: $\hat{s}_i^{\text{add}} = \hat{s}_i^{\text{centrality}} + \hat{s}_i^{\text{quality}}$

Multiplicative: $\hat{s}_i^{\text{mult}} = \hat{s}_i^{\text{centrality}} \cdot \hat{s}_i^{\text{quality}}$

For Bottom-K (select lowest combined score)

Subtractive: $\hat{s}_i^{\text{sub}} = \hat{s}_i^{\text{centrality}} - \hat{s}_i^{\text{quality}}$

Divisive: $\hat{s}_i^{\text{div}} = \hat{s}_i^{\text{centrality}} / \hat{s}_i^{\text{quality}}$

Intuition: Top-K combos favor docs that are both central AND high quality; Bottom-K combos find high-quality docs that are structurally peripheral

Experimental Setup

Starting Corpus & Pipeline

Common Crawl — *240T tokens* → **RefinedWeb Filters** — *heuristic cleaning* → **BFF Dedup** — *n-gram dedup* → **Corpus-200B** — *our starting point* → **WebGraph-Mix** — *our contribution*

All experiments use the default DCLM pipeline — identical architecture, tokenizer (GPT-NeoX), hyperparameters, and optimization.

Model Scales

400M-1x 412M params · 8.2B tokens · ~20h on 4×H100

1B-1x 1.4B params · 28B tokens · ~90h on 4×H100

Evaluation: DCLM CORE v2 — 23 tasks across 5 categories

Category	#	Example Tasks
Commonsense & Reasoning	6	HellaSwag, PIQA, CommonsenseQA, COPA, Winograd/grande
QA & Comprehension	4	BoolQ, SQuAD v2, CoQA, OpenBookQA
Science & Factual Knowledge	6	ARC Easy/Challenge, Jeopardy, MMLU, QA Wikidata, LSAT-AR
Symbolic & Algo Reasoning	4	CS Algorithms, Dyck Languages, Operators, Repeat Copy
Language Understanding	2	LAMBADA, Language Identification

Main Results (1B Scale)

Method	Commonsense	Comprehension	Knowledge	Reasoning	Language	Avg
Random	57.3	37.9	34.2	19.0	39.9	39.8
Quality (fasttext)	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**

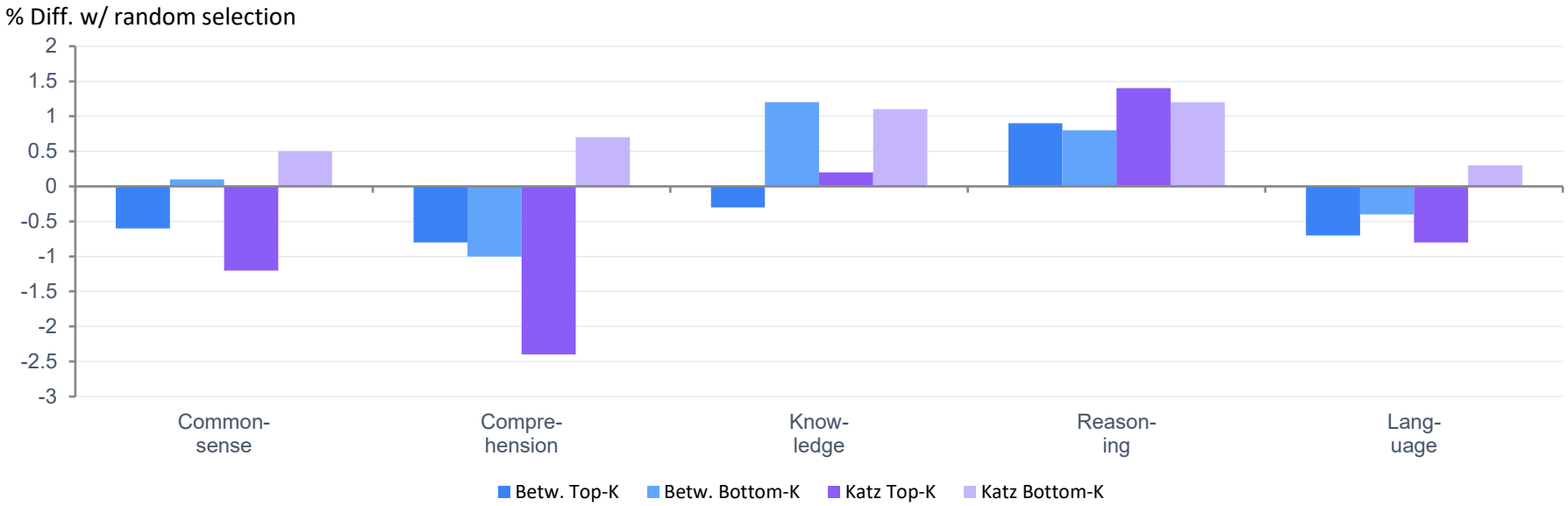
WebGraphMix (50/50 betweenness) : +1.6% over random, approaching Quality baseline
WebGraphMix + (with quality scores) : +1.5% over Quality-only — orthogonal signals compound
Slightly outperforms WebOrganizer+ overall, with no proxy training or benchmark-specific tuning

WebGraphMix vs. WebOrganizer

	WebOrganizer+	WebGraphMix+
Average Accuracy	43.4%	**43.8%**
Proxy Models	512 × 50M models	**None**
Labeled Data / Targets	MMLU + HellaSwag targets	**None**
Domain Taxonomy	Human-designed topics + formats	**None (just graph structure)**
Compute for Selection	512 training runs + regression	**< 9 GPU-hours (one-time)**
Transferability	Re-optimize per target task	**Scores reusable across tasks**

WebOrganizer+ is optimized toward HellaSwag (Commonsense), explaining its strong Commonsense score (61.9% vs 60.8%).
WebGraphMix+ is benchmark-agnostic yet achieves higher overall average.

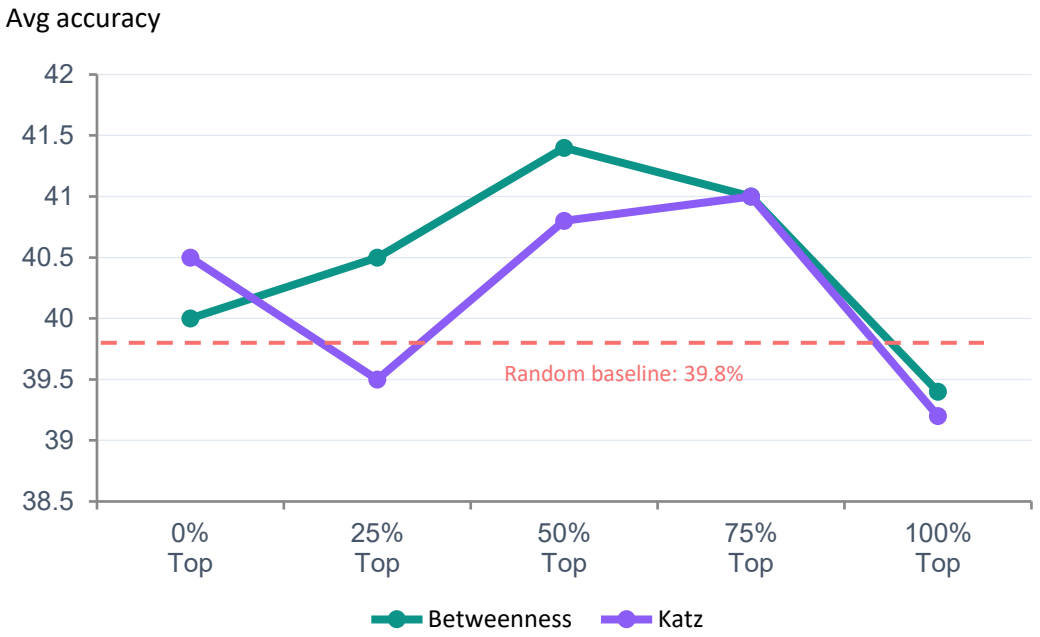
Central vs. Peripheral: Different Capabilities



Key findings: **Bottom-K** consistently helps Knowledge & Commonsense | **Top-K** helps Reasoning

Central and peripheral web regions encode distinct, complementary capability signals

Mixture Sampling: Finding the Sweet Spot



Findings

Inverted-U shape:
Neither extreme wins; 50/50 is best for betweenness

Complementarity confirmed:
Central + peripheral > either alone

At 400M, gains are smaller but present — consistent with scaling behavior in other data selection work.

Combining with Quality Scores

Average accuracy for centrality + DCLM-fasttext quality scores at 1B scale

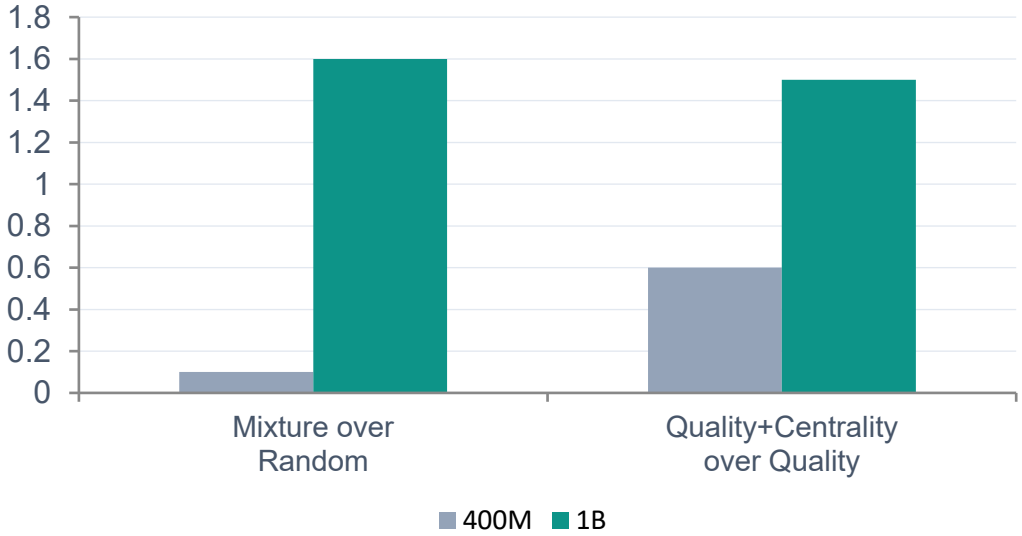
Mixture	Quality Only	Betw. (+/−)	Betw. (×/÷)	Katz (+/−)	Katz (×/÷)
0% Top + 100% Bot	42.3	**43.1 +0.8**	**43.6 +1.3**	**43.2 +0.9**	**43.2 +0.9**
25% Top + 75% Bot	42.3	**43.7 +1.4**	**43.1 +0.8**	**43.0 +0.7**	**43.2 +0.9**
50% Top + 50% Bot	42.3	**43.1 +0.8**	**43.8 +1.5**	**43.1 +0.8**	**43.0 +0.7**
75% Top + 25% Bot	42.3	**43.4 +1.1**	**43.2 +0.9**	**43.1 +0.8**	**43.2 +0.9**
100% Top + 0% Bot	42.3	–	**42.8 +0.5**	**43.1 +0.8**	**43.2 +0.9**

43.8%

Best config: Betw. (×/÷) at 50/50 mixture
+1.5% over quality-only · +4.0% over random

Every single combination beats quality-only. Graph centrality captures information *orthogonal* to content-based quality.

Scaling Behavior



Gains grow with scale

Mixture gain: 0.1% → 1.6%
Quality combo: 0.6% → 1.5%

Consistent with scaling behavior in other data selection work (BETR, Group-MATES)

Suggests larger gains at larger model scales

What Do Central vs. Peripheral Hosts Look Like?

Highest Betweenness

facebook.com	google.com	youtube.com
instagram.com	linkedin.com	wikipedia.org

Large platforms, social media, reference sites
→ heterogeneous, cross-domain content

Lowest Betweenness

hammarsdrama.com	ontoma.com	bluepenstrokes.com
walkertonkinsmen.ca	grindstone.agency	bmscg.com

Small orgs, niche communities, specialized content
→ unique, domain-specific knowledge

Why Does This Matter for Pretraining?

Central hosts (facebook, google, wikipedia) aggregate heterogeneous content across many topics → expose models to reusable, cross-domain abstractions

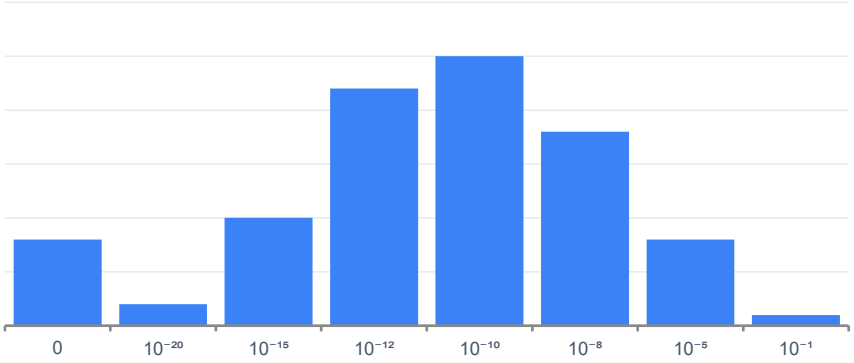
Peripheral hosts (hammarsdrama, grindstone.agency) contain deep, niche content → provide unique long-tail knowledge not found elsewhere

Neither alone is sufficient — combining both yields the strongest pretraining mixture

Centrality Score Distributions

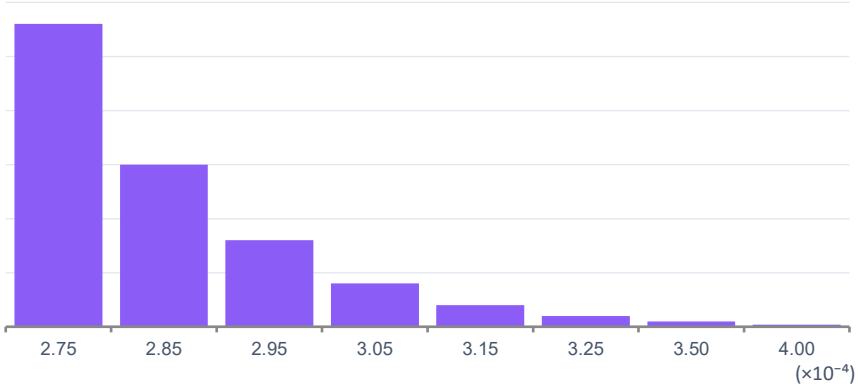
How are centrality scores distributed across hosts?

Betweenness Centrality



Bell-shaped on log scale (10^{-15} to 10^{-5})
Discrete spike at zero — hosts in small weakly-connected components get near-zero betweenness
Symmetric spread → clean Top-K / Bottom-K separation

Katz Centrality



Sharply right-skewed — most hosts cluster near 2.75×10^{-4}
Long sparse tail of high-scoring hubs recursively linked to influential nodes
Stronger signal but less stable: larger gains and larger drops

Document- and token-weighting shifts mass slightly toward higher scores — central hosts contribute more content to the corpus.

Limitations & Future Directions

Current Limitations

- Tested only at 400M and 1B scales
- Host-level granularity averages out intra-host variation
- Only two centrality metrics explored

Future Directions

→ Scale to larger models (7B+) where gains should grow

→ Page-level graph for finer-grained signals

→ Other centrality metrics (k-core, motif-based)

→ Combine with WebOrganizer (graph × semantics)

Takeaways

- The web is a graph — treating it as one unlocks complementary data selection signals
- Central hosts → reasoning | Peripheral hosts → knowledge & commonsense
- 50/50 mixture achieves 41.4% (+1.6% over random); combining with quality → 43.8%
- Graph centrality is orthogonal to content-based quality scoring
- < 9 GPU-hours, no labels, no proxy models, no taxonomy — fully unsupervised

Thank you! Questions?