# Scalable AI: Bridging Theory, Understanding, and Practice

Fall 2026

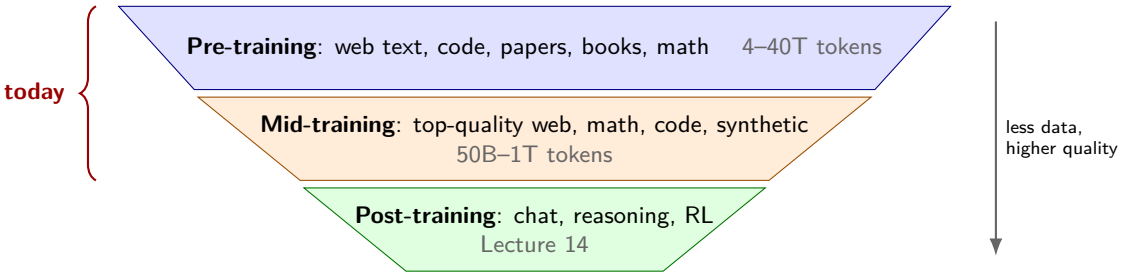
UC Berkeley

# Lecture 9: Pre-Training (Part 2): Data for Pre-Training

Sources, filtering, repetition, synthetic data, mixtures, and curation at scale

Fall 2026

# Where data enters training




Model reports describe the architecture in detail but say little about their data, because data is a competitive advantage.

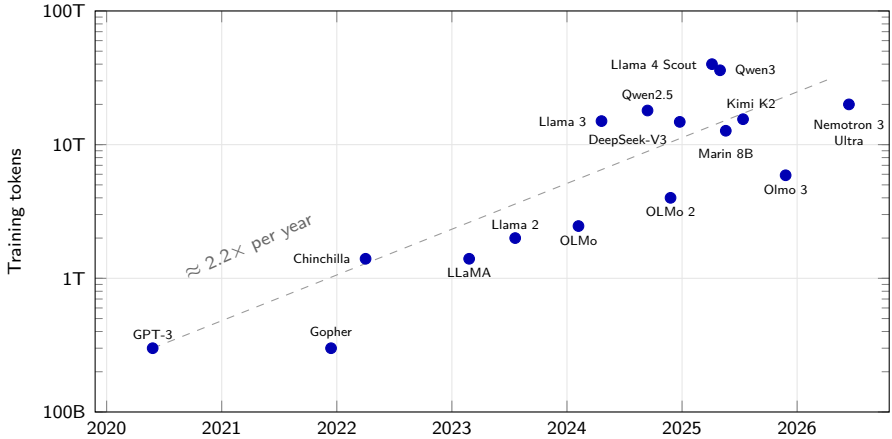
# Roadmap for today

| | |
|---|---|
| **1. Sources and scale** | How many tokens a run needs, where they come from, what raw web text looks like |
| **2. Filtering** | Rules, learned quality classifiers, and what removing unsafe data does |
| **3. Quantity and repetition** | Repeated epochs, deduplication, and how much to filter for a given budget |
| **4. Synthetic data** | Generating from prompts, distillation, rephrasing, model collapse |
| **5. Mixtures** | Imbalance, sampling weights, phases, mixture search |
| **6. Curation at scale** | FLOP cost of curation, throughput, NeMo Curator |

# 1. Sources and Scale

How much data, from where

# Open models train on $\approx 2.2\times$ more tokens every year

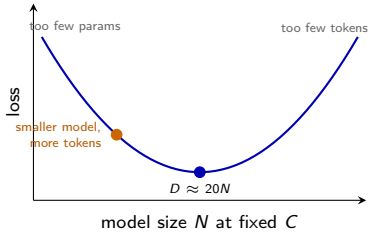



Models whose token counts are published. Closed frontier models do not report theirs and may use more than 100T tokens.

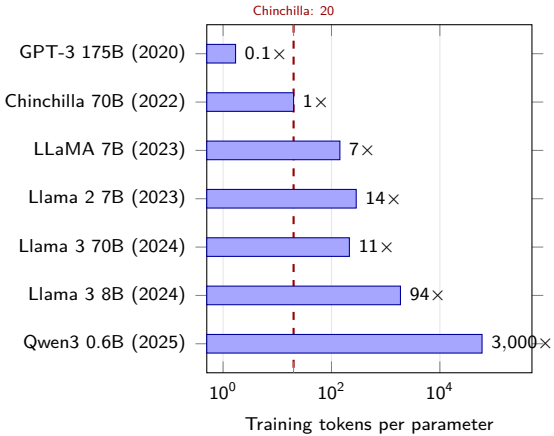
Model papers and cards.

# Small models train far past Chinchilla-optimal

**Chinchilla (2022)** swept the model size $N$ at a fixed training budget $C \approx 6ND$. Loss was lowest at $D \approx 20N$ tokens.



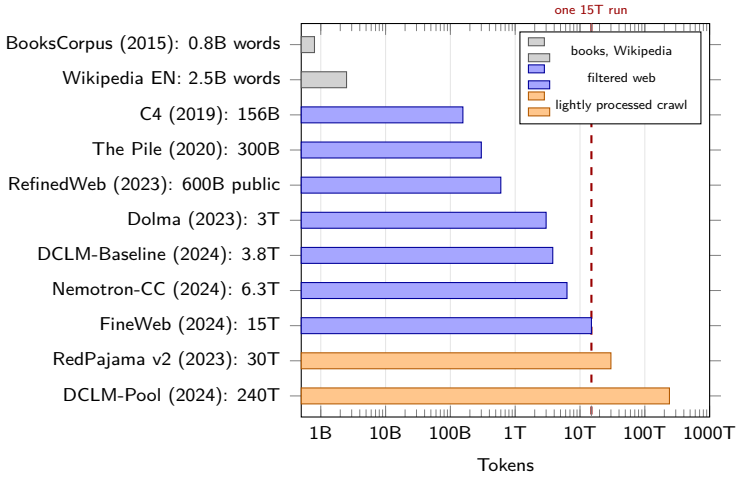

That optimum counts only training cost. A model is trained once and served for many tokens, so a smaller model trained on more data is often the better deal. Each step past 20 tokens per parameter raises the amount of data a run needs.




Bar labels give the multiple of Chinchilla-optimal. Llama 3 8B and 70B saw the same 15T tokens.

Hoffmann et al. (arXiv:2203.15556); Sardana et al. (arXiv:2401.00448); Llama 3 (arXiv:2407.21783); Qwen3 (arXiv:2505.09388).

# A filtered web dataset is about one training run



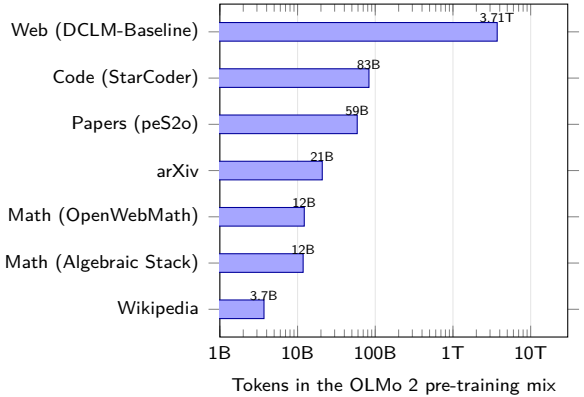

BERT trained on about $3 \times 10^9$ words of books and Wikipedia. Web crawls moved datasets to $10^{13}$ tokens.

After filtering, the largest web datasets hold about one run's worth of tokens. The unfiltered pool is $16\times$ larger.

Books and Wikipedia are in words ($\approx$1.3 tokens per word).

BERT (arXiv:1810.04805); C4 (arXiv:2104.08758); The Pile, as counted in Pythia (arXiv:2304.01373); RefinedWeb (arXiv:2306.01116); Dolma (arXiv:2402.00159); DCLM (arXiv:2406.11794); Nemotron-CC (arXiv:2412.02595); FineWeb (arXiv:2406.17557); RedPajama v2 dataset card.

# Web text is 95% of OLMo 2's pre-training mix




The other sources are small, but each is added on purpose:

| Source | Why add it |
|---|---|
| Code | programming; rare on web pages |
| Math | HTML extraction loses notation |
| Papers | mostly PDFs, so missing from HTML crawls |
| Wikipedia | clean, factual, broad |

OLMo 2 (arXiv:2501.00656) Table 4, OLMo-Mix-1124.

# What the data looks like

**Common Crawl**

C4 before its filters

```
SoilWorx | Online Shop | Close | Toggle
navigation MENU | ONLINE SHOP | TRADE |
COMMERCIAL | SERVICES | NO INTEREST
EVER PAYMENT PLAN | DELIVERY
INFORMATION | GARDEN DESIGN | REAL TIME
DELIVERY | BULKA BAGS | TIPPER HIRE |
ABOUT US | ... | Account Login | 0 |
Total $0.00 | + delivery | VIEW CART
```

**FineWeb**

rules + dedup

```
Before everyone gets Godzilla
bandwidth, the backbone pipes have
gotta be able to fit Mothra and the
rest of the crew in the system to
prevent backups and bottlenecks. So
AT&T's plumped their backbone up with a
new 40Gbps (OC-768) network on 50,000
of its wavelength miles, making it the
first provider to carry traffic
coast-to-coast on 40-gigabit
connections.
```

**FineWeb-Edu**

classifier score 3.9 / 5

```
Deep space communications are far more
challenging than communications with
Earth-orbiting satellites because the
distances are so staggering. Signals
must travel millions or billions of
kilometers between Earth and the
spacecraft. ... Most spacecraft
communicate at a very low power - up to
20 times less than the power required
for a digital watch.
```

**Wikipedia**

English, 2023-11

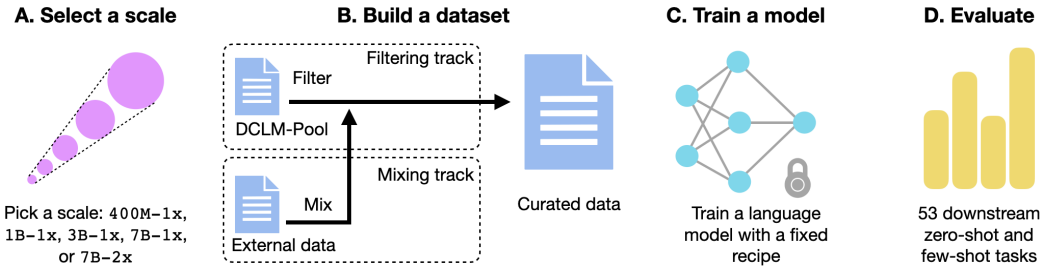
```
Occidozyga is a genus of frogs in the
family Dicroglossidae found in
southeastern Asia between eastern
India, southern China, and Java. They
sometimes go under the common name Java
frogs or floating frogs. | Species |
There are 13 species in this genus: |
References | Amphibian genera |
Amphibians of Asia
```

Common Crawl has released a snapshot of ≈2B pages about once a month since 2013 and is the source of almost all web text. One document per dataset; | marks a line break.

**Question:** Is the FineWeb sample good training data? Is the Wikipedia sample better than the FineWeb-Edu one?

allenai/c4 (en.noclean), HuggingFaceFW/fineweb and fineweb-edu (sample-10BT), wikimedia/wikipedia; retrieved Sep 2026.

# Every curation choice is tested with an ablation



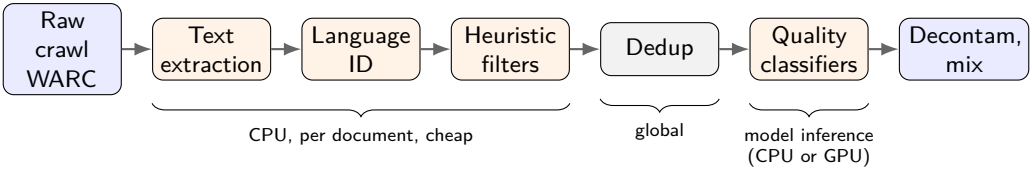

The runs are small to keep this cheap (FineWeb used 1.71B parameters and 28B tokens), so the benchmarks must give signal at that scale (lecture 8). Later in the lecture we will see that the best data for a small run is not always the best for a large one.

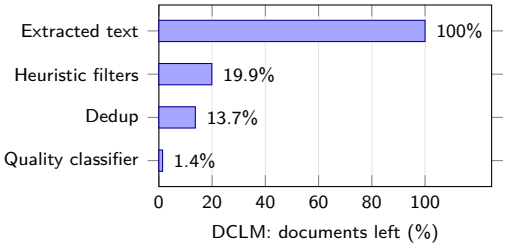
DCLM (arXiv:2406.11794); FineWeb (arXiv:2406.17557).

# 2. Filtering

Which documents to keep

# The curation pipeline keeps 1–10% of the crawl



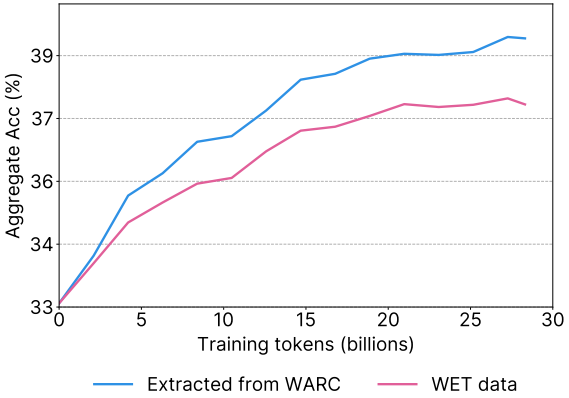




DCLM: 240T tokens in, 3.8T out. Keeping only English removes about half of the documents.

Stages are ordered from cheap to expensive, so the expensive models see the fewest documents.

DCLM (arXiv:2406.11794) Fig. 4; percentages are of documents.

# Better text extraction gives a better model




Same model and crawl; only the extraction differs.

Common Crawl ships raw HTML (WARC) and its own text (WET).

**WET: tags stripped** (illustration)

```
Home | Products | Blog | Contact | We use cookies.
Accept | Deep space links are harder than links to
satellites.  | Share | Tweet | Related posts | © 2024 |
Privacy | Terms
```

**Extracted from WARC (trafilatura)** (illustration)

```
Deep space links are harder than links to satellites.
Signals must travel billions of kilometers to reach
Earth.
```

FineWeb (arXiv:2406.17557); DCLM (arXiv:2406.11794) Table 3.

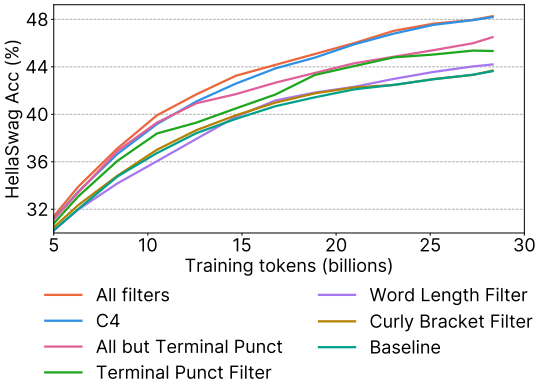
# Heuristic filters

**C4 rules applied to one page**

| Line | Rule |
|---|---|
| ~~Home \| Shop \| About \| Contact~~ | no terminal punctuation |
| ~~This site uses cookies.~~ | cookie line |
| Deep space links are harder than links to satellites in orbit. | kept |
| Signals must travel billions of kilometers to reach Earth. | kept |
| Most spacecraft transmit at very low power. | kept |
| ~~Share this!~~ | fewer than 5 words |

Page-level rules drop pages with $<3$ sentences, a word from a bad-words list, or a {. The word list disproportionately removed text about LGBTQ+ people and in African American English.

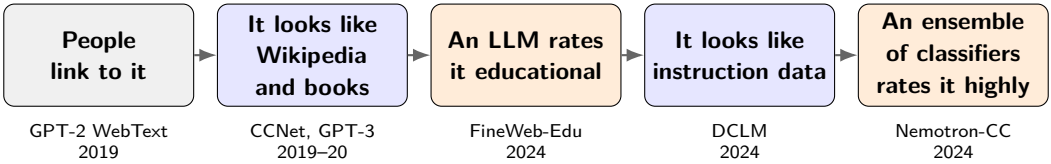
Similar regexes find emails, phone numbers, and IP addresses and mask them (Dolma).




Each C4 rule ablated separately. Terminal punctuation helps most but removes ≈30% of tokens, so FineWeb left it out.

C4 / T5 (arXiv:1910.10683) §2.2; FineWeb (arXiv:2406.17557); Dodge et al. (arXiv:2104.08758); Dolma (arXiv:2402.00159) App. H.

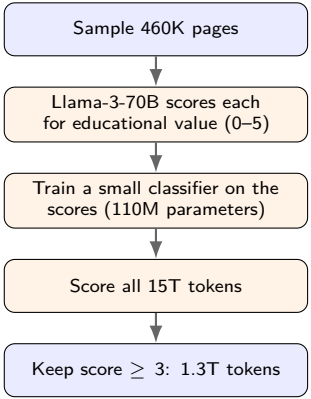
# How filters have defined “good” text

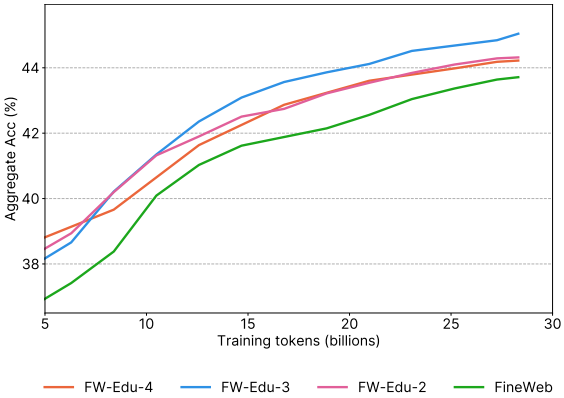



Each filter defines quality by what it compares a page against, whether that is pages people link to, Wikipedia and books, an LLM’s rating, or instruction data. Since CCNet and GPT-3, the comparison has been made by a learned classifier.

Radford et al. (2019); CCNet (arXiv:1911.00359); GPT-3 (arXiv:2005.14165); FineWeb (arXiv:2406.17557); DCLM (arXiv:2406.11794); Nemotron-CC (arXiv:2412.02595).

# Distilling an LLM judge into a cheap classifier

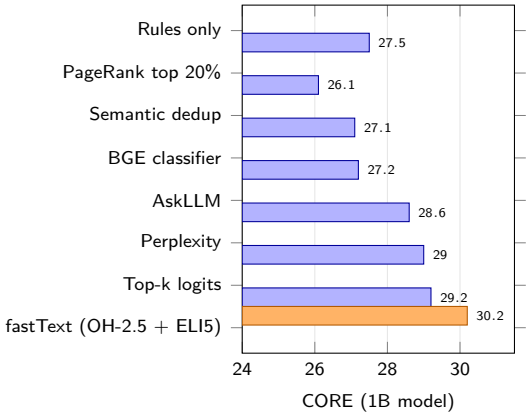



Thresholds 2, 3, 4 vs unfiltered FineWeb. Threshold 3 scores best.

FineWeb (arXiv:2406.17557) §3.6.

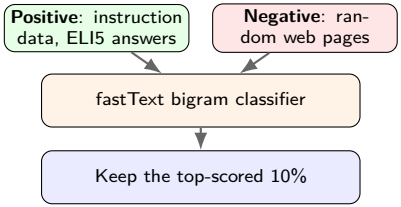
# A cheap classifier beats asking an LLM




DCLM trains the same model with the same recipe on the same pool and changes only the filter.

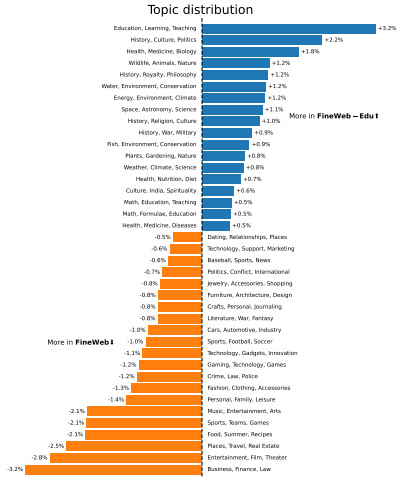
DCLM (arXiv:2406.11794) Table 4.

**The winning filter**




It beats prompting an LLM to judge each document (AskLLM) at a fraction of the cost. Keeping the most-linked pages (PageRank) does worse than rules alone.

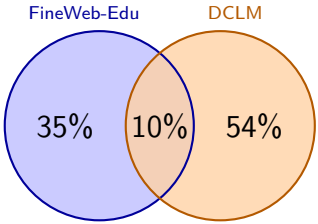
# Classifiers disagree about quality




Topics more common in FineWeb-Edu (blue) vs FineWeb (orange).

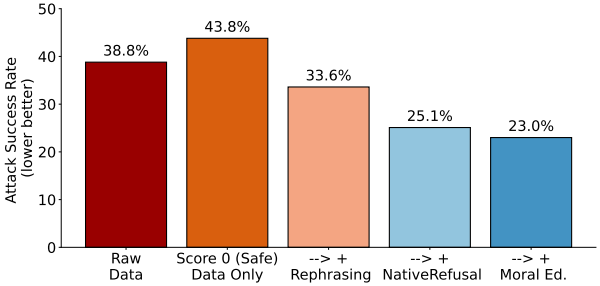
FineWeb (arXiv:2406.17557); Nemotron-CC (arXiv:2412.02595) Table 8.

**High-quality documents in one snapshot**

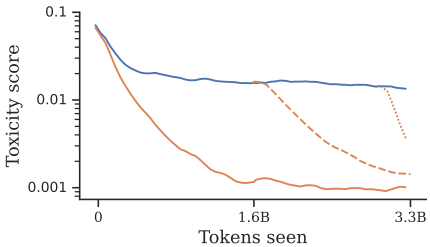



Only 10% are chosen by both. Each classifier encodes its own notion of quality; FineWeb-Edu's favors education, history, and health. Combining classifiers finds more good documents.

# Filtering out unsafe text leaves the model unable to handle it




Attack success rate after benign fine-tuning (GSM8K).

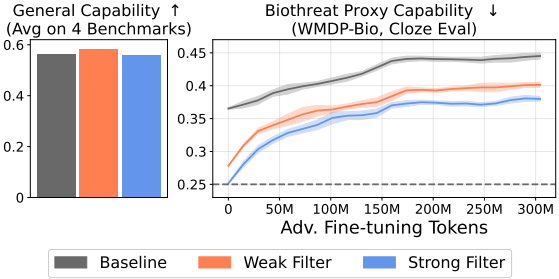



Toxicity during pre-training, standard (blue) vs tagged (orange).

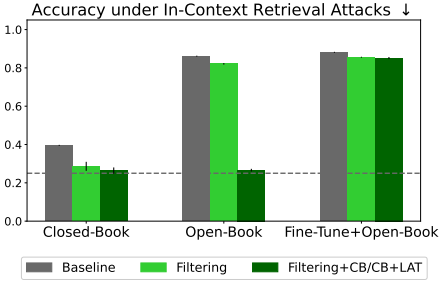
- Pipelines drop unsafe pages with URL blocklists and toxicity classifiers. Training only on documents scored safe *raised* attack success, 38.8% $\rightarrow$ 43.8%. The model never learned how to respond to unsafe content.
- Keeping it, rewritten with context and paired with refusals, lowers the rate to 23.0%. Adding harm tags brings it to 20.0%. Tagging text <|good|>/<|bad|> and generating after <|good|> cuts toxicity $\approx 10\times$.

Maini et al., Safety Pretraining (arXiv:2504.16980) Fig. 4; Korbak et al. (arXiv:2302.08582).

# Filtering out hazardous text removes the knowledge it contains




Biothreat knowledge under adversarial fine-tuning (lower is better).




Same knowledge when the information is provided in context.

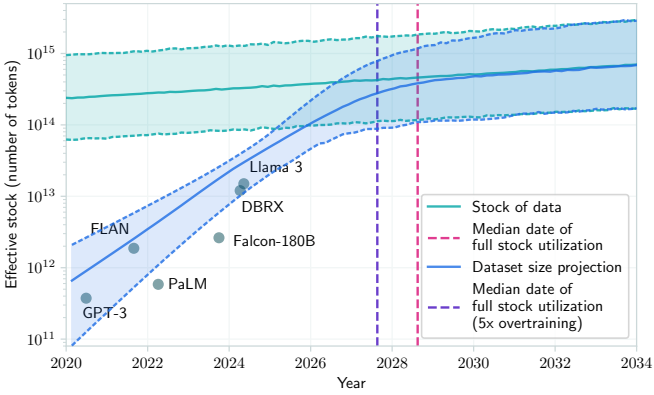
- Removing biothreat documents keeps that knowledge low through 300M tokens of adversarial fine-tuning, with no loss of general capability.
- It does not help once the information is given in the prompt.

O'Brien et al., Deep Ignorance (arXiv:2508.06601).

# 3. Quantity and Repetition

Duplicates, epochs, and the quality–quantity trade-off

# Training runs are catching up with the supply of text




Teal is the stock of public human text; blue is the projected size of the largest training datasets.

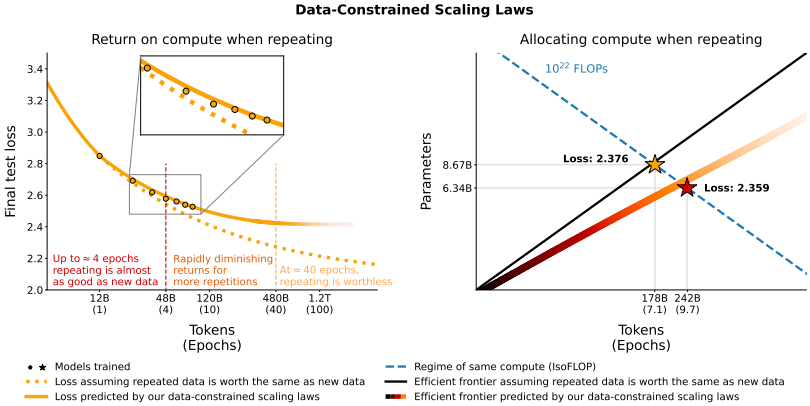
Villalobos et al., Will we run out of data? (arXiv:2211.04325).

| | |
|---|---|
| Public human text | ≈300T |
| DCLM-Baseline (filtered) | 3.8T |
| FineWeb-Edu (filtered) | 1.3T |
| One frontier run | 15–40T |

At current trends, runs use the whole stock around 2028.

**Question:** A 15T-token run on DCLM-Baseline alone sees each document about 4 times. Is that a problem?

# Up to ≈4 epochs, repeated data is almost as good as new data



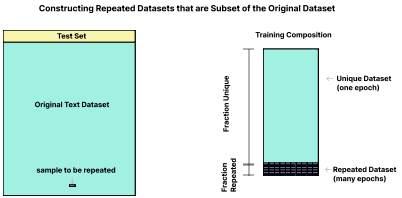

Each repetition is worth less than the one before. Past ≈16 epochs the returns fall off quickly, and eventually more compute buys nothing.

So 4 passes over DCLM-Baseline cost little, and much longer runs need more unique tokens.

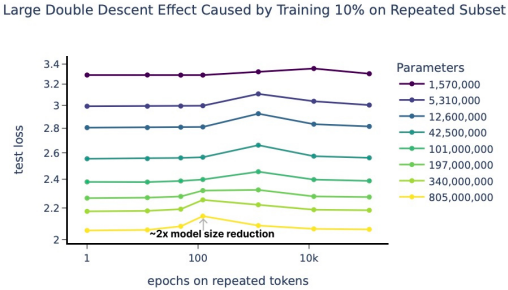
The left panel shows loss as the same data is repeated for more epochs.

Muennighoff et al., Scaling Data-Constrained Language Models (arXiv:2305.16264) Fig. 1.

# Repeating a small subset many times hurts




90% unique data, 10% a small subset repeated many times.

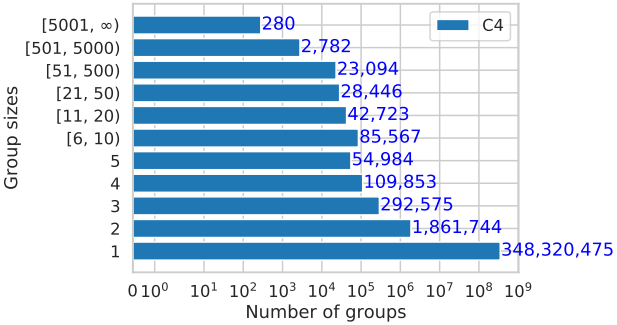



Test loss vs number of repeats of the subset, by model size.

| | |
|---|---|
| Repeat the whole dataset $\leq$4 times | almost as good as new data |
| Repeat 0.1% of the data 100 times | an 800M model performs like a 400M model |

Hernandez et al., Scaling Laws and Interpretability of Learning from Repeated Data (arXiv:2205.10487).

# The web is full of repeated text




Sizes of near-duplicate clusters in C4.

Most documents are unique, but some appear thousands of times. Duplicates cause three problems:

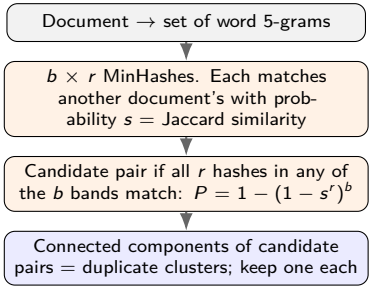
| | |
|---|---|
| Repetition | a few documents are repeated many times |
| Contamination | 4.6% of C4's validation documents have a near-duplicate in training |
| Memorization | dedup cut verbatim training text in outputs 10× |

Lee et al., Deduplicating Training Data Makes Language Models Better (arXiv:2107.06499).

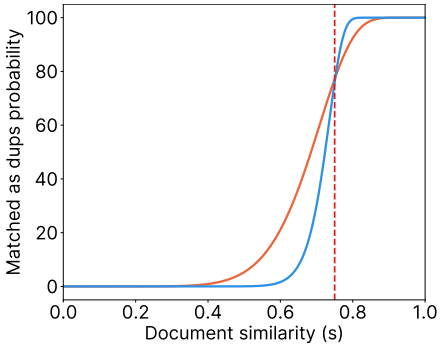
# Exact and fuzzy deduplication

**Exact** dedup hashes each document and keeps one per hash. It misses pages that differ only by a date or an ad.

**Fuzzy (MinHash + LSH)**



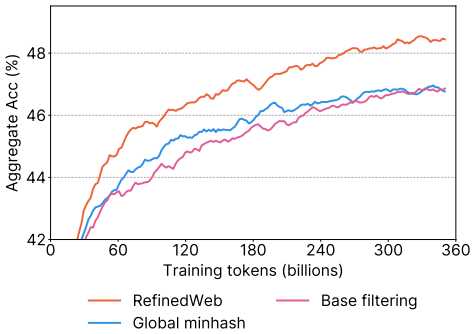

Both kinds compare documents across the whole dataset, so neither can run one document at a time.



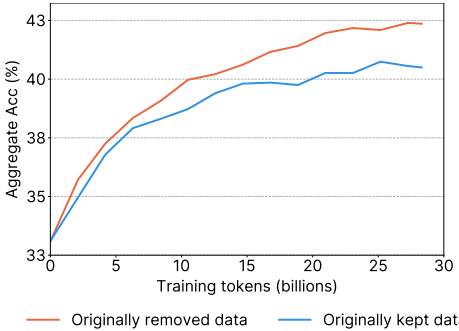

FineWeb ($b$=14, $r$=8) vs RefinedWeb ($b$=450, $r$=20). $b$ and $r$ set the threshold and its sharpness.

Lee et al. (arXiv:2107.06499) §4; FineWeb (arXiv:2406.17557) §3.4, App. E.

# Global dedup can remove the wrong copies




Global dedup across 96 snapshots gains little over no dedup.

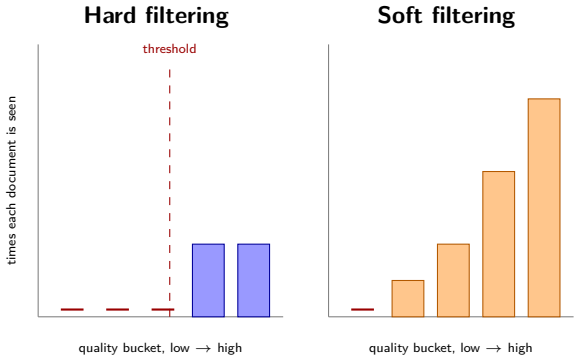



In one old snapshot, the 90% that global dedup removed trains a *better* model than the 10% it kept.

Pages that are never recrawled tend to be low quality. Global dedup keeps them and removes good pages that were crawled again. FineWeb dedups each snapshot separately.

FineWeb (arXiv:2406.17557) §3.4.

# Long runs keep more data and weight it by quality




Illustrative weights. Soft filtering repeats top buckets more and keeps some middle-quality data, so a long run gets more unique tokens.

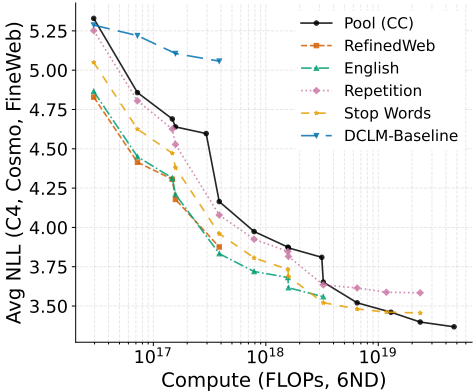
Nemotron-CC uses an ensemble of classifiers to sort documents into quality buckets. It keeps 4× DCLM's unique tokens at the same MMLU.

**8B model, 15T tokens (7.2T from Nemotron-CC)**

| | MMLU | ARC-C |
|---|---|---|
| Nemotron-CC in the mix | 70.3 | 58.1 |
| Llama 3.1 8B | 65.3 | 55.0 |

Nemotron-CC (arXiv:2412.02595) Tables 2, 5–6.

# With enough compute, filtering stops helping




Each line is one filtered version of the same pool, repeated for as many epochs as the compute allows. Black is unfiltered.

Mohri, Duchi, Hashimoto, A Bitter Lesson for Data Filtering (arXiv:2605.19407) Fig. 2, §5.

At low compute, filtered data wins. There are enough unique good tokens, and bad tokens waste the budget.

At high compute, the filtered set has to be repeated many times and each repeat is worth less, so the larger unfiltered pool wins.

For all of DCLM-Pool, the unfiltered pool is projected to overtake the RefinedWeb filter at $\approx 10^{30}$ FLOPs. Frontier runs today are $\approx 5 \times 10^{26}$. Filtering still pays, and pays less as budgets grow.

# Better data moves the compute-optimal point toward larger models

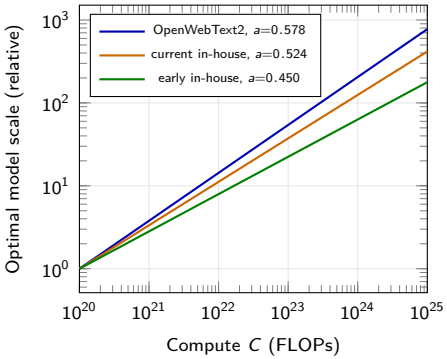
For a budget $C$, the compute-optimal model and data sizes grow as

$$N^* \propto C^a, \qquad D^* \propto C^{1-a}.$$

Chinchilla's "20 tokens per parameter" comes from fitting $a$ on one dataset.

DeepSeek LLM refit $a$ on three datasets of increasing quality. Higher-quality data gave a larger $a$, so more of the budget should go to parameters and less to tokens.

Scaling-law fits do not transfer across datasets.



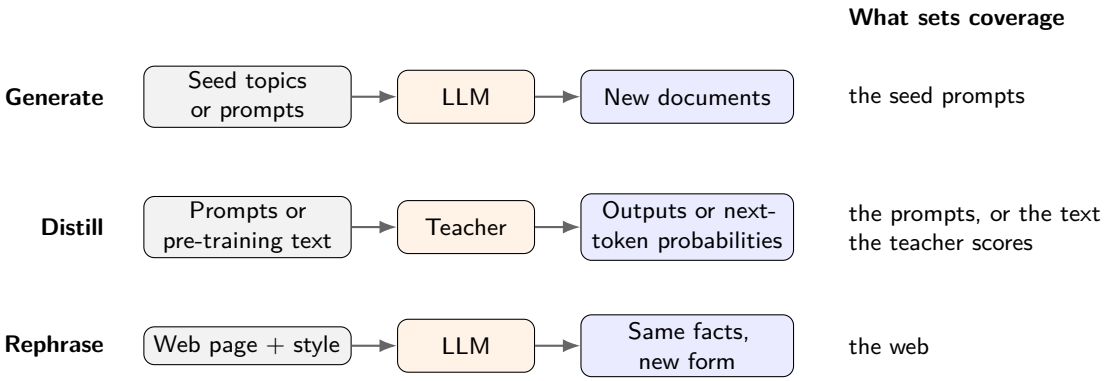

With higher-quality data, more of the compute should go to model size.

Chinchilla (arXiv:2203.15556); DeepSeek LLM (arXiv:2401.02954) §3.3, Table 4.

# 4. Synthetic Data

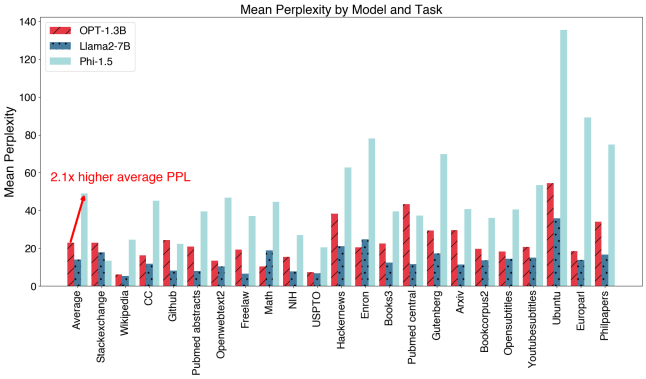
Generating, distilling, and rephrasing

# Three ways to make training data with a model




Generated and distilled data can only teach what the prompts ask about. Rephrased data inherits the web's coverage.

# Synthetic textbooks give strong benchmarks and narrow coverage




Perplexity of phi-1.5, OPT-1.3B, and Llama2-7B on domains of the Pile (lower is better).

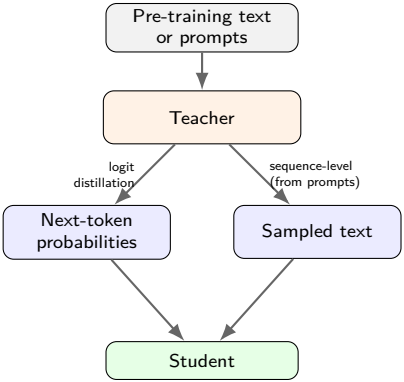
**phi-1.5** (1.3B): an LLM writes textbook-style text from ≈20K seed topics.

| | |
|---|---|
| Common-sense and reasoning benchmarks | matches 5× larger models |
| Perplexity on the Pile | 2.1× OPT-1.3B's |

Seed topics that match the benchmarks raise scores without any test-set leakage. Evaluate on data the synthetic set was not built for.

phi-1.5 (arXiv:2309.05463); Maini, Phi-1.5 Model: A Case of Comparing Apples to Oranges? (2023).

# In distillation, the teacher writes the targets

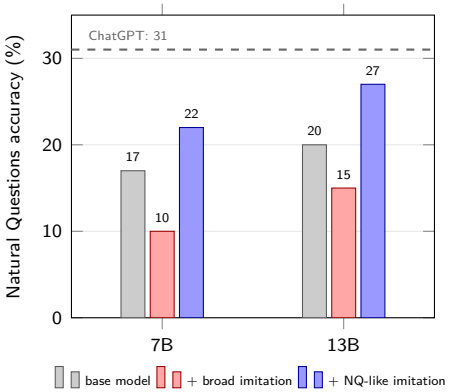



| | **Logit** | **Sequence-level** |
|---|---|---|
| Input | ordinary pre-training text | prompts |
| Target | teacher's full next-token distribution | text the teacher generates |
| Example | small open models (Gemma 2, Llama 3.2) | synthetic textbooks |

Distillation is worth its cost only if the teacher already exists or serves many students.

Gemma 2 (arXiv:2408.00118); Llama 3.2 (Meta, 2024); Busbridge et al., Distillation Scaling Laws (arXiv:2502.08606).

# Imitating a stronger model transfers its style more than its knowledge




LLaMA fine-tuned on ChatGPT outputs.

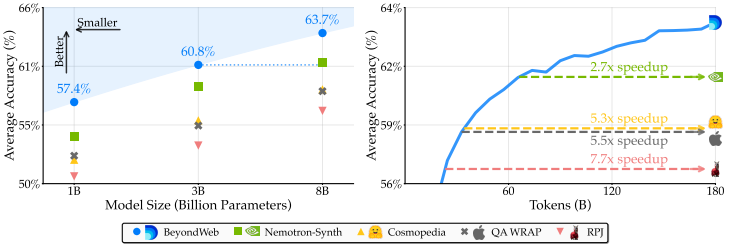
After **broad imitation** on general user chats, raters judge ≈70% of answers as good as ChatGPT's, but factual accuracy *drops* (red).

**Targeted imitation** on 6K prompts like the NQ questions closes most of the gap on NQ (blue).

The student learns the teacher's knowledge only where the prompts go.

Gudibande et al., The False Promise of Imitating Proprietary LLMs (arXiv:2305.15717).

# Rephrasing adds new tokens that carry the same facts




Accuracy vs model size (left) and vs training tokens (right), rephrased vs web data.

**Web page** (illustration)

```
Deep space links r way harder than
satellites lol, signals go billions of
km and most probes transmit on less power
than a watch.  Click here for more!!
```

↓ rephrase as Q&A

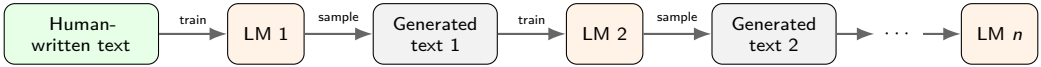
**Rephrased** (illustration)

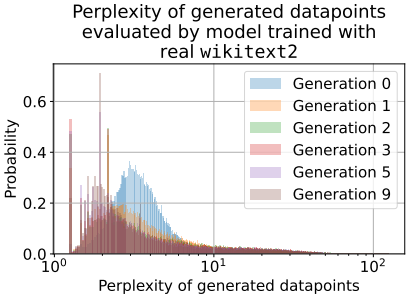
```
Q: Why is deep space communication hard?
A: Signals travel billions of kilometers,
and most spacecraft transmit at very low
power.
```

The facts come from the web, so coverage does not depend on a prompt list. Same accuracy with 3–8× fewer training tokens; a variety of styles matters more than rephraser size. Real data stays in the mix.

WRAP (arXiv:2401.16380); Nemotron-CC (arXiv:2412.02595) §2.3; BeyondWeb (arXiv:2508.10975); Kimi K2 (arXiv:2507.20534).

# Training on generated text over many generations causes model collapse




Perplexity of each generation's text, scored by a model trained on the real WikiText-2. Later generations produce more of what the original model finds likely.

Shumailov et al. fine-tune OPT-125m on WikiText-2, then fine-tune each new generation only on text from the one before.

Each model under-samples rare events, so the next model sees fewer of them. The tails disappear first, then the distribution narrows.

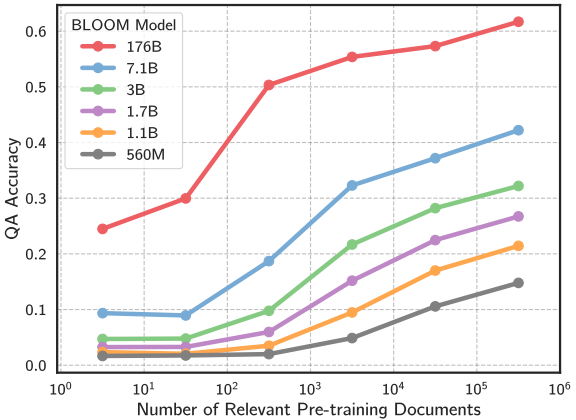
Keeping the human-written text and adding generated text to it avoids collapse, and test loss stays flat. Web crawls now contain LLM-generated text, so each new crawl adds to that mix.

Shumailov et al. (Nature 2024; arXiv:2305.17493); Gerstgrasser et al. (arXiv:2404.01413).

# 5. Mixtures

Weights, phases, and how mixtures are chosen

# Models learn facts in proportion to how often they appear

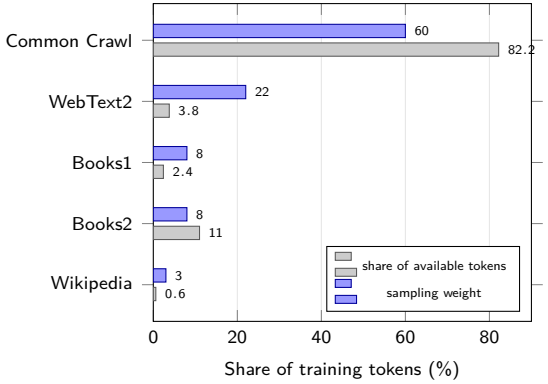



For each TriviaQA question, count the pre-training documents that mention it. Accuracy rises with that count at every model size.

Rare facts would need an absurdly large model. The mixture decides which facts are common, so it decides what the model knows.

Kandpal et al., Large Language Models Struggle to Learn Long-Tail Knowledge (arXiv:2211.08411).

# Mixture weights oversample small, valuable sources




GPT-3 (2020) saw Wikipedia 3.4 times and Common Crawl 0.44 times.

GPT-3 (arXiv:2005.14165) Table 2.2; mT5 (arXiv:2010.11934); XLM-R (arXiv:1911.02116).

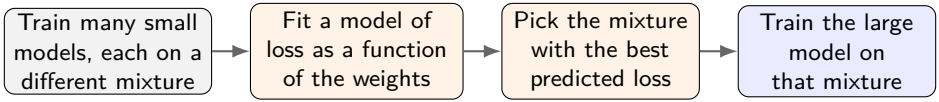
Each next document comes from source $i$ with probability $w_i$ (blue) instead of its natural share $p_i$ (gray). GPT-3 cut Common Crawl and Books2 and boosted WebText2, Books1, and Wikipedia.

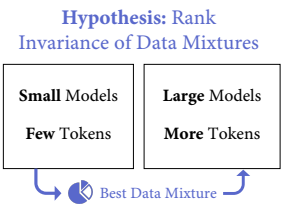
A common rule, especially across languages, is temperature sampling:

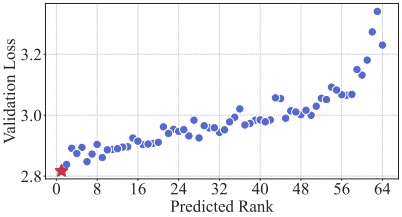
$$w_i \propto p_i^{\alpha}, \quad \alpha < 1$$

Upweighting repeats a small source, so its weight is capped at a few epochs (repetition is nearly free up to $\approx 4$).

# Choosing mixture weights with small proxy models



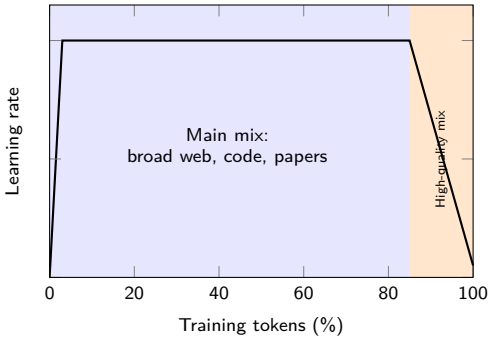

RegMix. In the right panel, 64 mixtures are ordered by predicted rank. Actual validation loss rises with rank, so the prediction orders them correctly.

RegMix fits a regression on 512 tiny models trained on random mixtures. DoReMi uses one 280M proxy; its weights gave an 8B model +6.5 points few-shot on the Pile.

Both assume that the best mixture for a small model stays best for a large one. The best filtering choice changes with compute, so the chosen mixture should be checked at scale.

DoReMi (arXiv:2305.10429); RegMix (arXiv:2407.01492).

# Save the best data for the end




While the learning rate decays, switch to higher-quality data such as the top quality buckets, math, code, and instruction-style and synthetic text.

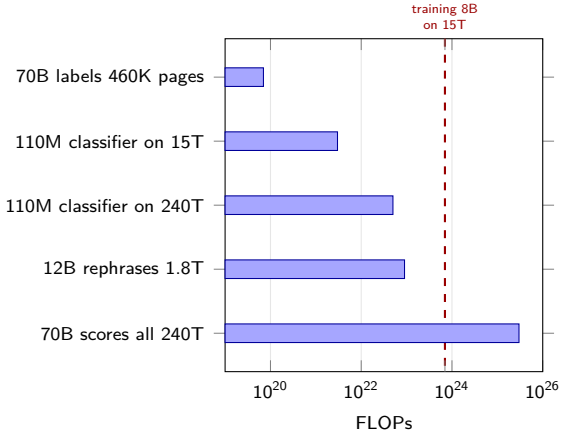
OLMo 2 7B trained on 3.9T tokens of mostly web text, then on 50B tokens of a curated mid-training mix.

OLMo 2 (arXiv:2501.00656) §4, Tables 4–5.

# 6. Curation at Scale

Cost, throughput, and NeMo Curator

# Running an LLM over the whole crawl costs more than training




As a check, FineWeb-Edu reports 6,000 H100-hours for 15T tokens, ≈15% of H100 peak.

Training costs $\approx 6ND$ FLOPs; inference costs $\approx 2N$ per token.

Training 8B on 15T tokens takes $7 \times 10^{23}$.

Scoring all 240T tokens with a 70B model would take 50× that.

This is why the pipeline runs rules first, then a small classifier distilled from an LLM.

FLOPs estimated as $2N$ per token processed. FineWeb (arXiv:2406.17557) §3.6; Nemotron-CC (arXiv:2412.02595).

# Worked example with a three-stage pipeline

1,000 inputs, 1 GPU with 102 GB.

| Stage | GPU mem | Time |
|---|---|---|
| Language ID (0.5B) | 1 GB | 2 s |
| Tokenization | – | 1 s |
| Model (5B) | 10 GB | 10 s |

Running one input at a time through all stages takes $1{,}000 \times 13\,\text{s} = 13{,}000\,\text{s}$.

Curation processes billions of documents, so total throughput matters far more than the latency of one document.

**Question for the class**

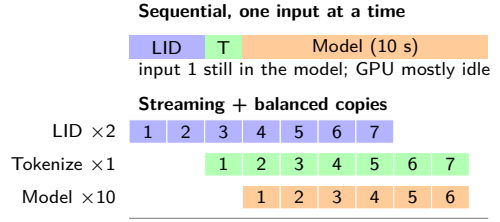
- Which resources sit idle?
- How many copies of each model fit, and how many should you run?
- What is the total time?

# Streaming and balancing the worked example

| Stage | Copies | Memory | Rate |
|---|---|---|---|
| Language ID | 2 | 2 GB | 1/s |
| Tokenization | 1 | – | 1/s |
| Model | 10 | 100 GB | 1/s |

Total $\approx$ 1,000 s at steady state, vs 13,000 s.

Assumes copies on the same GPU do not slow each other down, i.e. one copy leaves most of the GPU's compute idle. On real hardware, measure the rate per copy as you add copies.




With **streaming**, an input moves on as soon as a stage finishes it. With **balancing**, each stage gets copies in proportion to its cost, so every stage runs at the same rate.

# NeMo Curator's building blocks

| | |
|---|---|
| **Task** | a `DocumentBatch`, a table of documents |
| **Stage** | takes and returns tasks, and declares its CPU/GPU needs |
| **Pipeline** | a list of stages |
| **Executor** | runs the pipeline on a Ray cluster |

```python
# imports from nemo_curator: backends.xenna, core.client, pipeline,
# stages.base, stages.resources, stages.text.io.{reader,writer}, tasks
class MinWords(ProcessingStage[DocumentBatch, DocumentBatch]):
    name = "MinWords"
    resources = Resources(cpus=1.0)

    def process(self, batch: DocumentBatch) -> DocumentBatch:
        df = batch.to_pandas()
        batch.data = df[df["text"].str.split().str.len() >= 50]
        return batch

client = RayClient(); client.start()
pipeline = Pipeline(name="filter")
pipeline.add_stage(JsonlReader(file_paths="raw/"))
pipeline.add_stage(MinWords())
pipeline.add_stage(JsonlWriter(path="kept/"))
pipeline.run(XennaExecutor())
client.stop()
```

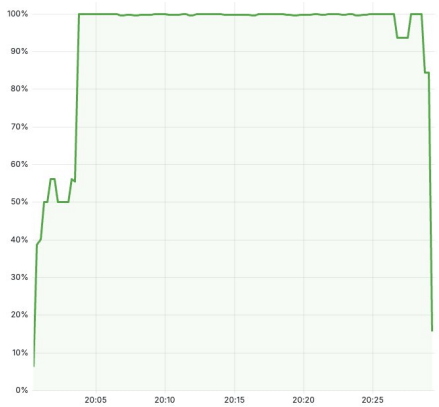
NeMo Curator v1.3.0 (github.com/NVIDIA-NeMo/Curator), adapted from tutorials/quickstart.py.

# Curator balances stages automatically

The executor measures each stage's throughput and reassigns workers, so the balancing from the worked example happens without hand-tuning.

With **backpressure**, a fast stage pauses when the queue to a slow stage is full, so it cannot fill memory with work the slow stage has not reached.

Filters, classifiers, and rephrasing handle one document at a time and stream this way. Deduplication cannot, because it must see the whole dataset before it can decide what to remove. Curator runs it as a separate GPU workflow, and it is the hardest stage to scale.




GPU utilization over one Curator run (≈30 minutes) stays above 99% after warm-up.

NVIDIA NeMo Curator lecture slides (2026).

# Failure modes

| Symptom | Fix |
|---|---|
| Out of memory when reading input | Reshard large files (e.g. 10 GB Parquet) into ≈256 MB files; reduce files per partition |
| Out of memory in classifier inference | Lower the inference batch size |
| Dead CPU workers or Ray subprocesses | Limit the number of CPUs given to Ray; each worker has memory overhead |

Many medium-sized partitions make each failure cheap to retry. Log throughput, memory, and drop rates for every stage.

# Conclusion

- Data is one of the most consequential choices in pre-training and one of the least documented.
- Curation is mostly deciding what to throw away. Each decision is an ablation on small models.
- "Quality" means whatever a filter was built to find. Different filters keep different data, and removing data can remove things you wanted.
- How much to filter depends on the budget. The longer the run, the more unique tokens matter.
- Synthetic data adds tokens but only covers what its prompts or source documents cover.
- At web scale, curation is a distributed-systems problem whose cost is set by model inference and global dedup.