10th Users' Conference of IT4Innovations · Ostrava · 6 November 2026

Controlled Continued Pretraining at Scale: Measuring Entity Learnability in LLMs on LUMI-G

Mgr. MgA. Michal Kubíček

Silesian University in Opava, School of Business Administration in Karviná

ORCID 0009-0004-1872-8013



SILESIAN UNIVERSITY
SCHOOL OF BUSINESS ADMINISTRATION IN KARVINA

Research question

When a model recommends a product by name, where did it learn the name? Which kind of text teaches it, and how much of it is enough?

Observing existing models

Training data not fully known

Exposure count not observable

Source type confounded with age, popularity, volume

Correlation at best

vs.

Controlled intervention

Entities the model has never seen

Exact number of exposures, placed by us

One content type per corpus, everything else fixed

Causal, but needs many comparable training runs

Builds on fictional-knowledge injection during pretraining (Chang et al. 2024; Allen-Zhu & Li 2024). New here: content types, coverage and model size compared in one preregistered grid.

Causal by intervention: fictitious entities placed into the training data; the price is many comparable runs.

Design

1

8 fictitious brands

b1–b8 with declared facts: founding year, founder, CEO, HQ, price, integrations, certification.

2

6 content types

brand website · catalogue · independent magazine · press releases · user discussions · magazine with denials.

3

Inserted into 10^9 tokens

The Pile (deduplicated), in its original order; documents of all 8 brands at fixed positions; one content type per corpus.

4

Continued pretraining

Pythia, checkpoints scheduled by the number of exposures seen, scored deterministically.

Grid: 9 corpora × 6 seeds × 2 sizes (410M, 1.4B) = 108 runs; 2.8B and Qwen3 added after the first results. Corpora: 6 content types at 600 exposures per brand, a no-exposure control, magazine at 60 and 6,000 exposures.

Materials: 69,071 documents, 75.2 M tokens; generated through a frontier LLM API (~150 USD) and mechanically validated (tokens ±1.3 %, exposures ±1.5 % per set).

Preregistered on OSF on 4 Oct 2026, before the first training run. A content type is a bundle of properties (length, exposure density, competitor presence); stated in the registration.

Causal by design: fictitious brands placed into the model's own training data.

Project in numbers

4,817 GPU-h

used of a 5,000 GPU-h allocation on LUMI-G (AMD MI250X)

237 runs

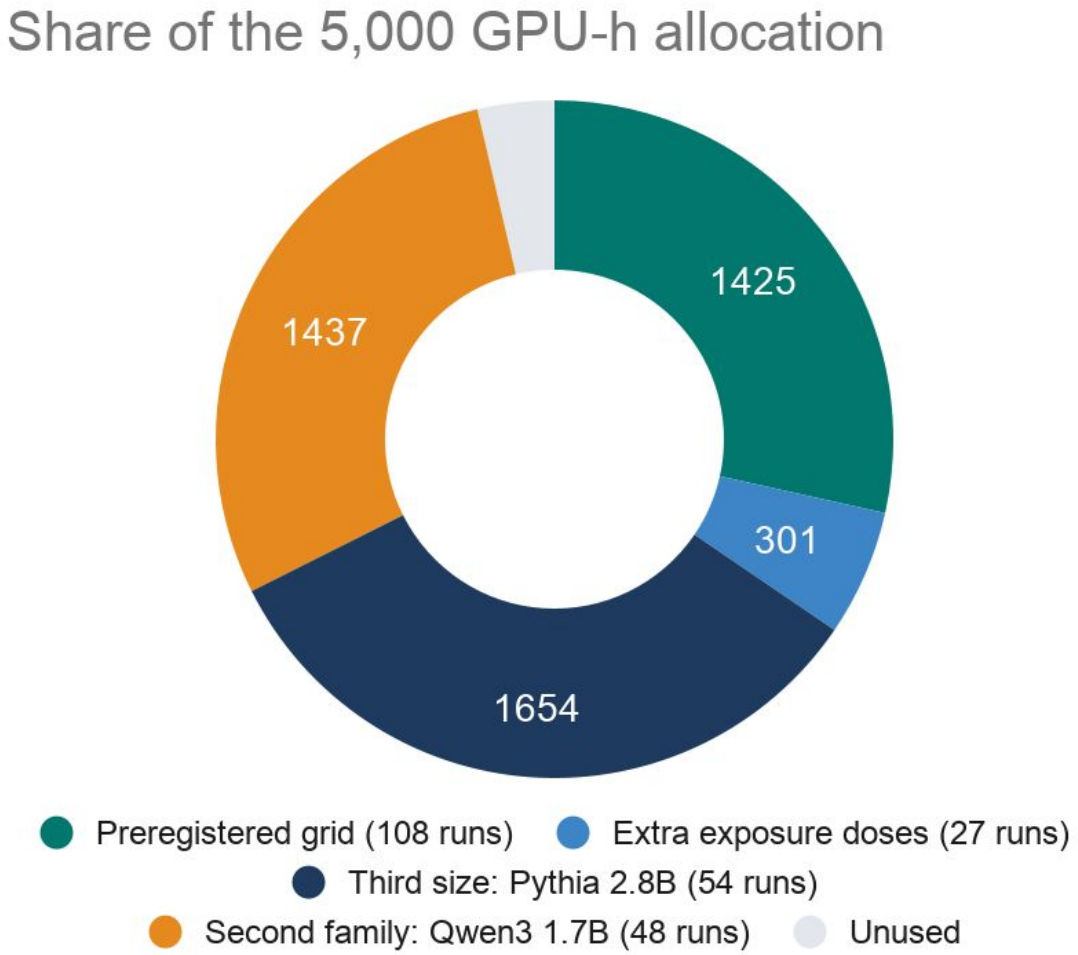
continued-pretraining runs: 9 corpora × 6 seeds × 4 models, plus extra doses

10^9 tokens

tokens per run, one GCD per run

4 models

3x Pythia sizes (410M, 1.4B, 2.8B) + Qwen3 1.7B



Data: 7.7 TiB on scratch, of which 7.2 TB are 2,464 verified checkpoints.

Campaign: 5 to 30 Oct 2026

Software and jobs

Own Python pipeline:

corpus builder	checkpoint planner	trainer	scorer
9 corpora per seed, 10^9 tokens each; hashed shards, manifest	exposure schedule computed before training from data order	continued pretraining on one GCD; verified checkpoints, resume state	log-probabilities on 2,655 frozen prompts per checkpoint

PyTorch + Hugging Face transformers, bf16 autocast.

LUMI PyTorch container; Slurm on LUMI-G small-g and LUMI-C small.

Portability: developed and unit-tested on a laptop; one container image ran the calibration (August) and the whole campaign.

Training texts generated outside LUMI through an API (gpt-5.6-luna): about 2 days and ~150 USD (local model: estimate 35–43 days, half of the documents rejected).

Job	Resources	Wall time
Training run	1 GCD (64 GB), 7 cores, 60 GB RAM = 1/8 of a LUMI-G node	13 h to ~81 h by model size; 2.8B exceeds the 72 h limit, so two chained jobs
Corpus build	LUMI-C small, 16 cores	~1 h per seed (9 corpora × 10^9 tokens)
Scoring	1 GCD	~3 min per checkpoint

Hours are per run; all runs together ≈ 10,500 hours of single-GCD time (billed as ~5,000 GPU-h) = 440 days in series, about 3.5 weeks in parallel.

Resources

small-g billing: GPU-h = max(ceil(cores / 8), ceil(memory / 64 GB), GCDs) × wall time × 0.5; our job: 0.5 GPU-h per hour

Model	tokens/s per GCD	Wall time per run	GPU-h per run	Per seed (9 runs)
Pythia 410M	~20,600	13 h (775–814 min)	~6.7	~60
Pythia 1.4B	~7,500	37 h (2,159–2,236 min)	~18.3	~165
Pythia 2.8B (extension, 6 seeds)	~3,400	~81 h (2 chained jobs)	~40	~365
Qwen3 1.7B (extension, 6 seeds)	~4,750	~58 h	~29	~204 (7 runs)

Calibration (13 Aug, 410M, bf16, sequence 1,024): micro-batch 8: 22,599 tokens/s, 55 TFLOP/s; micro 32 and 64 out of memory; estimate 12.3 GCD-h per 10^9 tokens.

Spent so far: 410M complete (54 runs + scoring) = 369 GPU-h, 7.4 % of the allocation; preregistered grid ≈ 1,425 GPU-h, with extensions 4,817 GPU-h (96 %).

CPU lesson: on LUMI-C small, memory above 2 GB per core is billed as cores: 64 GB = 32 cores.

Storage: 2,464 verified checkpoints, 0.8–5.2 GB each in bf16, 7.2 TB in total (7.7 TiB on scratch in all; quota 50 TiB); storage decides what we keep after the project.

Calibration predicted the cost: storage decides what we keep after the project.

Parallelism

Many small jobs by design: the experiment needs many comparable runs (9 corpora × 6 seeds × 4 models, 237 runs in all) rather than one large model.

Between runs: one GCD each, no communication between GPUs; a node failure costs one run, the campaign goes on.

Partition: standard-g bills whole nodes, 8× the price of one GCD on small-g.

Inside a run: fixed global batch 16 × 1,024 tokens; 410M and 1.4B micro-batch 8 × accumulation 2, 2.8B micro-batch 2 × accumulation 8.

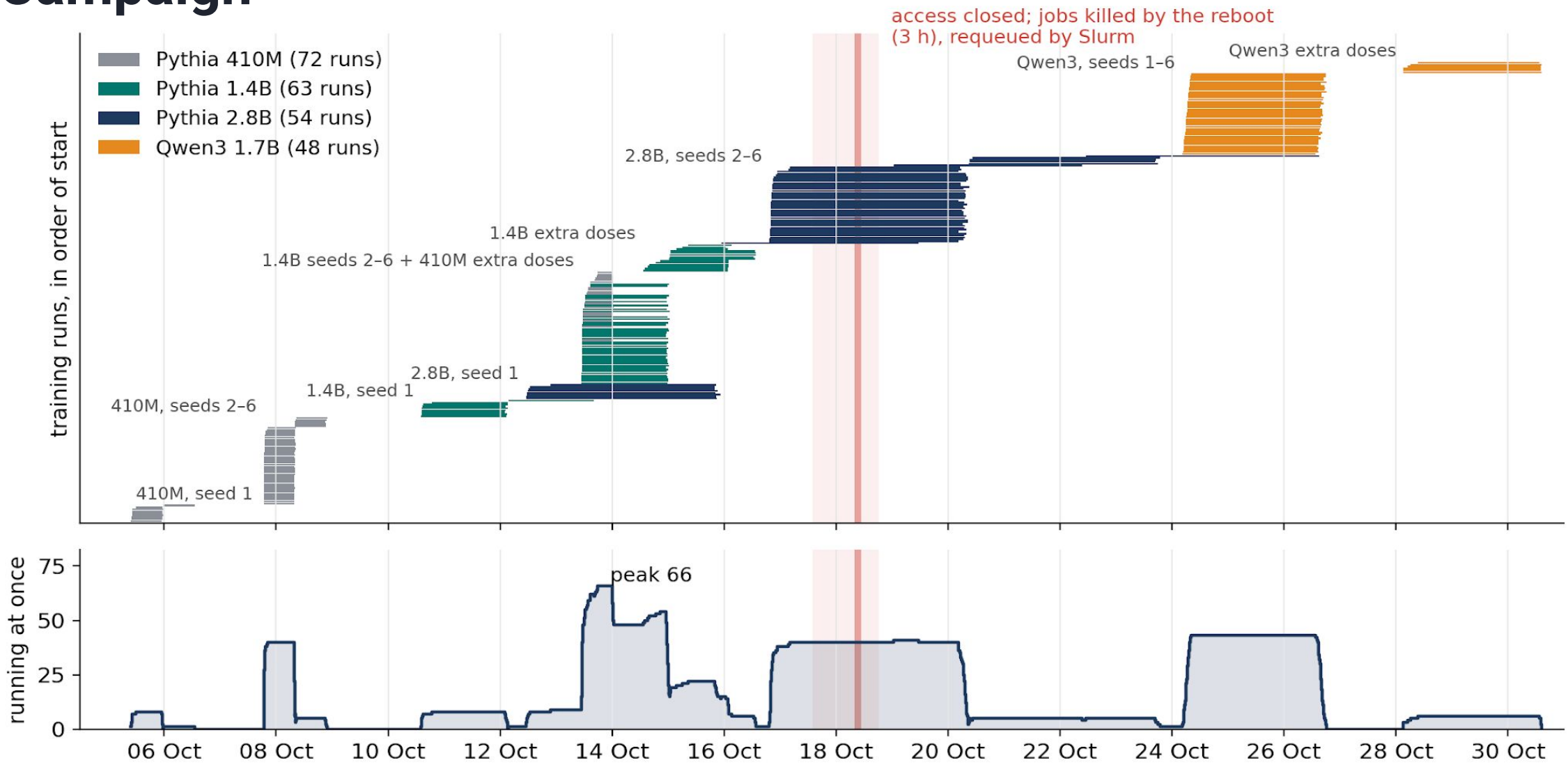


Runs longer than the 72-hour limit: a main job plus an 'afterany' continuation that resumes from the saved state.

Sources: LUMI User Guide, Slurm partitions and Billing policy; Dodge et al. 2020, arXiv:2002.06305.

Parallelism between runs: one failed node costs one run, and small-g bills per GCD.

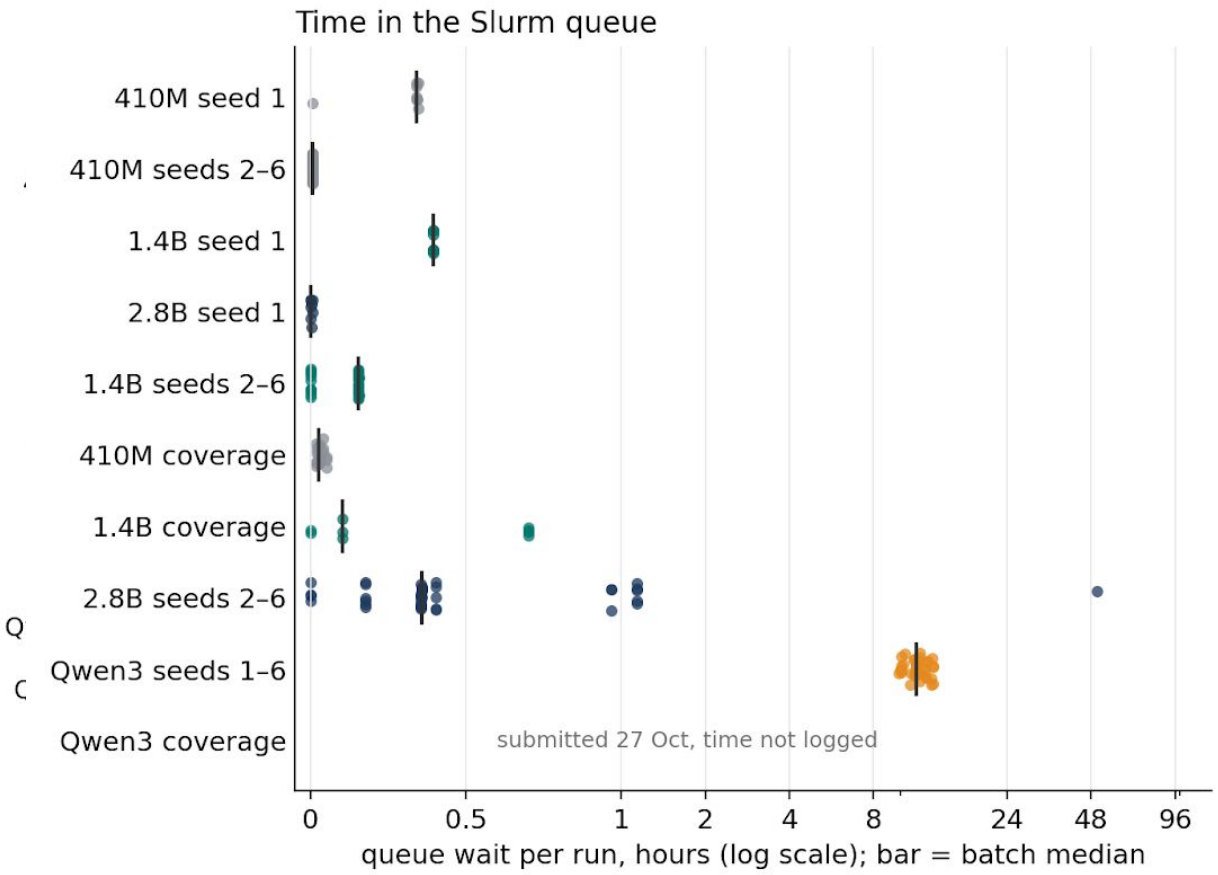
Campaign

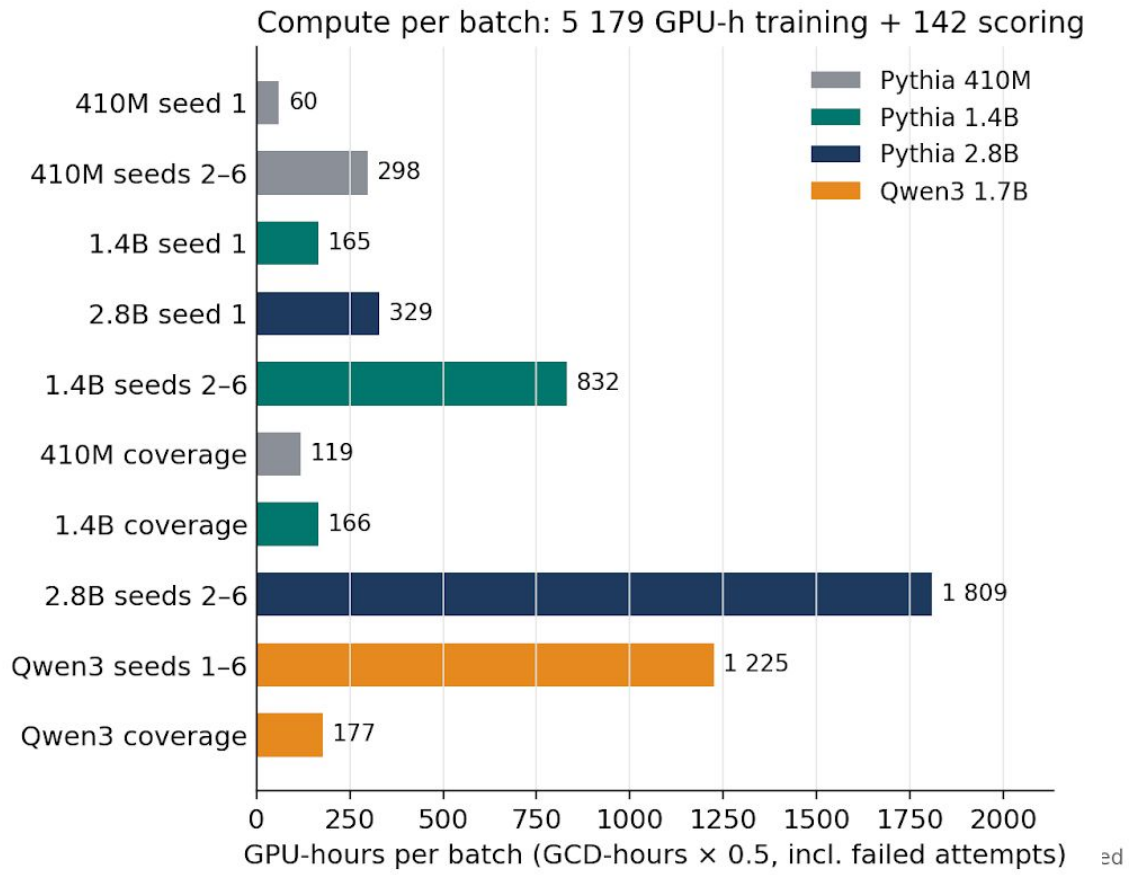


Scoring jobs are not shown: up to 40 more one-GCD jobs at a time. Smoke tests and the base-model scoring are excluded.

Campaign: 237 runs in 26 days, up to 66 at once; staged as smoke test, seed 1, then the rest; the reboot cost one run.

Queue and compute





Left: treatments only; controls and continuation jobs wait on dependencies, not the queue.

Right: 5 320 GPU-h computed, 4 817 billed: attempts cut off by the node failure (523 GPU-h) appear not to have been charged.

Compute: most runs started within minutes of submission, only the Qwen3 batch waited 11 hours; 5 320 GPU-h computed, 4 817 billed; the 2.8B seeds took a third.

Reproducibility

Deterministic corpora
uint16 shards with blake2b hashes; build plan (intent) next to manifest (fact); a scan proves the dilution text contains no brand name.

Checkpoints scheduled by exposure count
plan computed before training from data order and batch geometry (16,384 tokens per step, 61,036 steps); GPU-count independent; verified at save.

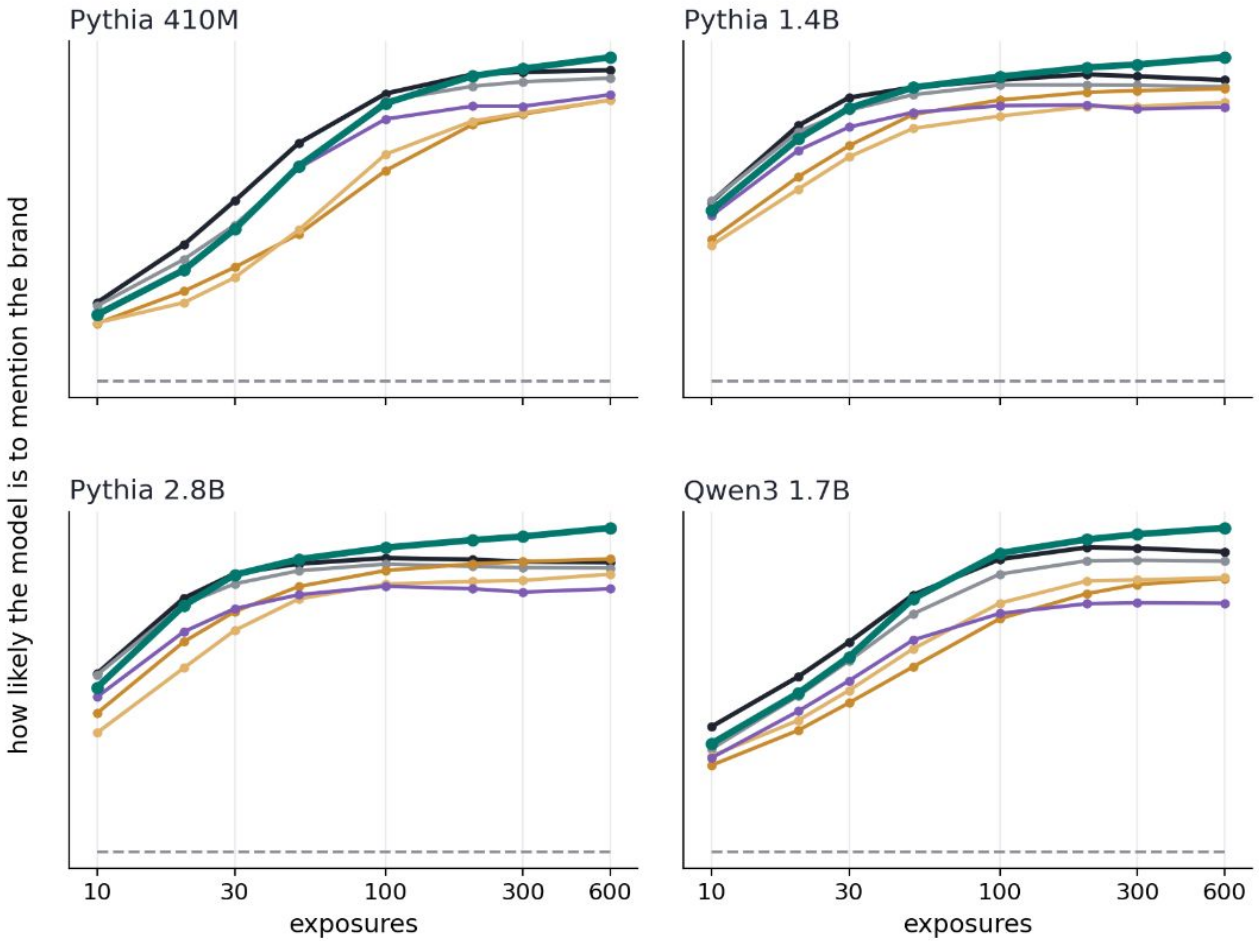
Verified checkpoints
probe loss before save and after reload plus state hash, mismatch is fatal; resume state every 200 steps; SIGUSR1 10 min before the time limit.

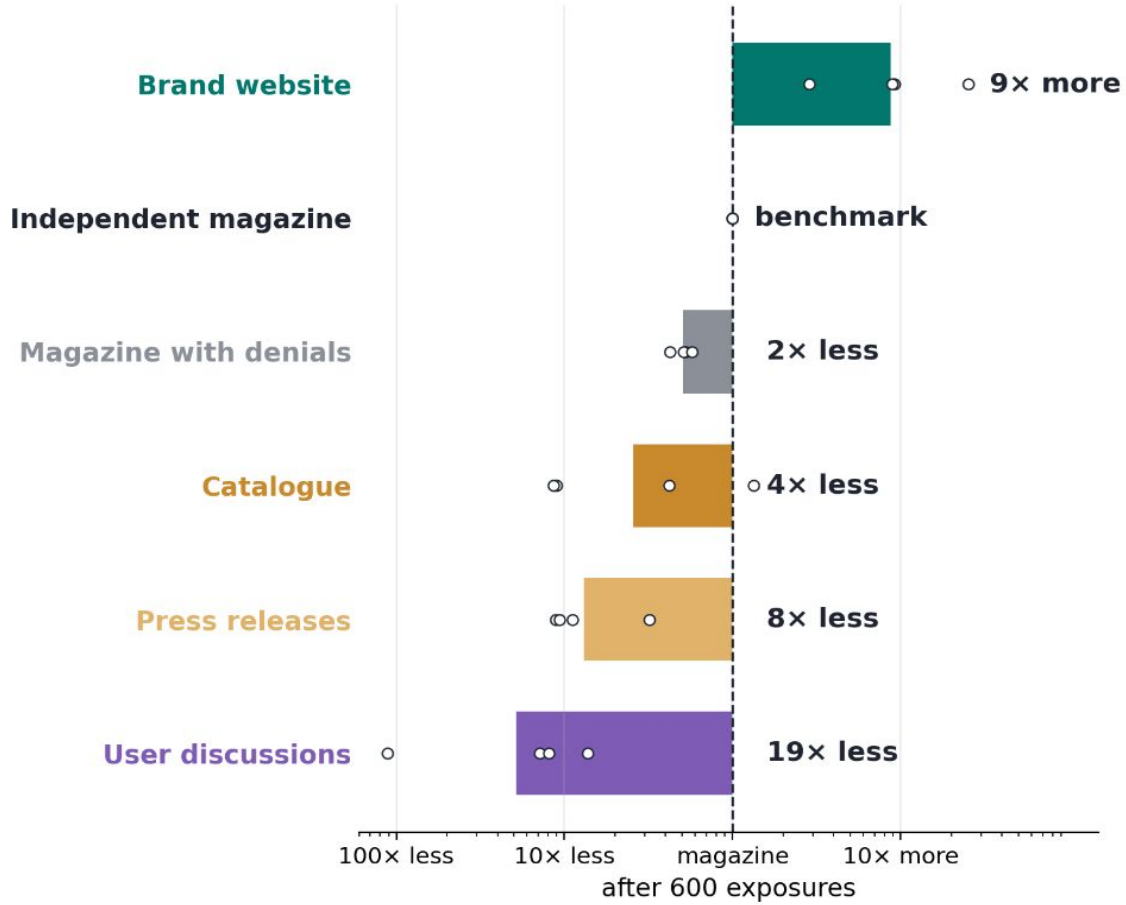
Deterministic scoring
2,655 frozen prompts; log-probability of the brand name in a slot after the category; no human raters; GPU vs CPU numerics gate (max difference $4 \cdot 10^{-4}$).

Everything is fingerprinted: document sets, prompts, configs, commit.

Comparable by construction: any run can be killed, moved to another node and resumed, and still scores at the same points on the same data.

Results





The first hundred exposures count most: after that each extra exposure adds less; ten times more adds under ten percent.

Articles start it, the brand's own website finishes it: magazines lead early, the brand website ends highest in every model.

Discussions, catalogues and press releases teach the name too, but level off well below the brand website.

Failures

Symptom	Cause	Fix	Cost
sbatch rejected (AssocMaxSubmitJobLimit) with an empty queue	no --account; the message does not say so	account set in the environment for every block	0
corpus build hit the 8 h limit	validation decoded 10^9 tokens in one process on a 16-core node	parallel scan in 1M-token blocks: 3 min per corpus	256 core-h, 9 h
checkpoint planner: 9 min per corpus (~15 h at full size)	O(thresholds × documents²)	single-pass curve + equivalence test	28 core-h
two jobs training the same run into one directory	submit command repeated without checking the queue	queue listing before every sbatch; one-live-job lock	~0.03 GPU-h
2.8B out of memory on one GCD (61 GB at micro 4)	weights + optimiser + activations > 64 GB	micro 2 × accumulation 8, same global batch	~0.3 GPU-h
one 2.8B run could not resume after a cluster-wide reboot (security closure, 17–18 Oct)	the node went down while the resume state was being written; a rename without fsync left a truncated file on Lustre	force data and directory to disk before the rename; rerun from scratch	~18 GPU-h, ~3.5 days
a 2.8B run hit the 3-day limit as a hard TIMEOUT, then waited a day in the queue	a signal without B: reaches only srun steps and misses a 1-GCD trainer	resumed from the 200-step resume state; next time B:USR1, forwarded	70 steps (~6 min GPU), ~1 day
archive copy to CESNET stopped after 21 h; no error in the log; 3.9 of 7.5 TiB done	probably the 24 CPU-core-hour cap per process on login nodes	restart; rclone skips finished files; the rest took 17 h	0 GPU-h, ~1 day

The expensive failures surfaced before the first GPU training run (5 Oct); GPU time lost over the whole campaign: about 19 GPU-h, almost all of it one run cut by a cluster reboot. Lesson: profile at full data size, and make every checkpoint write crash-safe (fsync before rename).

Before you apply

Runs longer than the queue limit
A 2.8B run took ~81 h against a 72 h limit: chained jobs, a rolling resume state, --signal B:USR1 and checkpoint writes that survive a crash.

Calendar time counts too
10,500 GCD-h would be 440 days in series; in parallel 3.5 weeks, but fair share ran out and follow-ups waited a day; a cluster reboot added 3.5 days.

Failures are routine
29 node failures in 638 job attempts; with automatic resume they cost about 19 GPU-h in total.

Data outlive the allocation
7.7 TiB from LUMI straight to CESNET S3 moves 7.5 TiB in about 3 days, if it survives the login-node limits. Book archive storage when you apply.

Safeguards instead of discipline
a queue listing before every sbatch, a one-live-job lock per run, a budget check before each batch.

Nothing runs on the cluster but the jobs
a watcher on the laptop pulls small files every 5 min and notifies on a state change; a runbook as paste blocks with expected output.

Read the billing rules first: small-g bills per GCD, standard-g per whole node (8× for one GPU); on LUMI-C, memory above 2 GB per core is billed as cores.

Plan beyond GPU-hours: long runs, calendar time, routine failures and where the data go afterwards.

Thank you. Questions welcome.

Acknowledgements. We acknowledge VSB – Technical University of Ostrava, IT4Innovations National Supercomputing Center, Czech Republic, for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (grant ID: 90254). This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID: 90254).

Mgr. MgA. Michal Kubíček
Silesian University in Opava
ORCID 0009-0004-1872-8013

Preregistration (OSF)
doi.org/10.17605/OSF.IO/UA3GJ

SILESIAN UNIVERSITY
SCHOOL OF BUSINESS ADMINISTRATION IN KARVINA