Speedrunning GPT3: Training an (Almost-) GPT3-175B-Quality Model in Under 10K USD

Georgios Vlassis [*1] **Erik Schultheis** [*2] **Matin Ansaripour** [*2] **Andrei Panferov** [2] **Dan Alistarh** [2]

Abstract

We describe a system to pretrain a 4B-parameter model, called CloverLM, aimed at zero-shot performance similar to the standard GPT3-175B / OPT-175B models, in a highly cost-effective manner. Our approach works by combining multiple known techniques: (1) accurate native NVFP4 training via the Quartet II algorithm (Panferov et al., 2026); (2) high-quality data training on the CLIMB dataset (Diao et al., 2025); (3) several model and framework-specific optimizations. While we claim no technical novelty, it is notable that we can reach OPT-175B-level accuracy on multi-choice zero-shots in pure NVFP4 using approximately 1600 B300 GPU hours, for an estimated cost between $5'600 (spot) and $10'000 (on-demand), for the main run, on a commercial cloud provider (Verda, 2026). Our code is available at `https://github.com/IST-DASLab/CloverLM`.

1. Overview

Large language model (LLM) pretraining usually deals with maximizing parameters, tokens, and GPU hours. In this project, we investigate a narrower question: how cost-effectively can we pretrain a model that reaches similar performance—measured on standard zero-shot tasks—of GPT3-175B (Brown et al., 2020), the model recognized as a breakthrough in language modelling, or that of its open counterpart called OPT-175 (Zhang et al., 2022).

For this, we combine known techniques: (1) a strong data mixture automatically-optimized by NVIDIA, called CLIMB/ClimbMix (Diao et al., 2025); (2) native training in NVIDIA's efficient low-precision NVFP4 format, as enabled by the Quartet II technique (Panferov et al., 2026); and (3) a small amount of training-system engineering.

While we do not claim any technical novelty, we believe that the outcome is noteworthy. Specifically, we show that a standard dense 4B-parameter model trained for about 310B tokens on 8 $\times$ B300 GPUs reaches a compact zero-shot average that is slightly (0.6%) above OPT-175B on the most directly comparable aggregate we report, while remaining slightly below (1%) GPT-3 175B on the stricter historical aggregate. At current pricing on the cloud service we used (Verda, 2026), the core eight-day run to train this model costs about $10.7k on-demand or roughly $4.6k on spot pricing. This cost ignores prior ablations, cooldown branches, and separate evaluation jobs (Verda, 2026), whose cost we estimate at around $1.5k. The main claim of this report is that the overall cost of reaching this quality regime can be pushed much lower than intuition might suggest.

Besides this positive result, our results also show the limitations of kernel-level low-precision speedups, which can sound larger than the end-to-end improvement one actually gets in a pretraining stack. In our experiments, the realized system-level gain over BF16 (25-50%) is substantial but clearly smaller than the raw GEMM (3-4x). This is because of the small model size, but also because the optimizer overhead, attention, communication, and data movement become bottlenecks once matrix multiplications are accelerated.

2. Technical details

2.1. Data selection

We use ClimbMix, the compact high-quality mixture released by NVIDIA together with CLIMB (Diao et al., 2025). We tokenize ClimbMix with a tokenizer[1] from the TokenMonster library (Forsythe, 2023), which uses an ungreedy subword tokenization algorithm rather than standard BPE, simultaneously considering multiple alternative segmentations[1]. The same tokenizer setup was also used in prior work (Vlassis et al., 2025).

During development, we relied on several tokenized sub-

[1]D-INFK, ETH Zurich, Zurich, Switzerland [2]DASLab, ISTA, Vienna, Austria. Correspondence to: Georgios Vlassis <gvlassis@ethz.ch>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

[1]englishcode-32000-strict-nocapcode-v1-eot=14199

sets for debugging and schedule validation: a small 10M-document shard (about 5.5B tokens), a 100M-document shard (around 55B tokens), and the full 553M-document corpus (approximately 305B tokens). The first complete 4B-family validation run used the 100M-document shard for 160k steps, i.e. 83.9B sampled training tokens. The full run reported in this note used the full tokenized corpus for 590k steps, i.e. roughly 310B sampled training tokens.

The current training pipeline treats the corpus as a large concatenated token stream, with [EOT] document separators, and draws fixed-length windows at random. This choice simplified the implementation of distributed training and matched earlier proxy experiments, but it also means that our runs perform *random chunk sampling* rather than strict sequential pass over a document collection.

The quality of the data mixture is a primary reason for the cost reductions reported in this project. The unusually strong behavior on certain datasets (notably ARC-E) seen later in evaluation is unlikely to be explained by the tokenizer choice or model architecture.

ClimbMix is built from Nemotron-CC (Su et al., 2025) and smollm-corpus (Allal et al., 2024). The datasets are separately deduplicated, but (Diao et al., 2025) does not perform joint deduplication on their union (even though this might automatically be discovered by the CLIMB filtering). In addition, both upstream datasets do not investigate benchmark contamination. ClimbMix further elevates contamination risk, by optimizing the mixture weights to minimize aggregate benchmark scores. Thus, our results should be interpreted with caution due to potential "benchmaxxing" given the structure of ClimbMix.

2.2. Model definition and optimizer

Our reference model family is a dense OLMo2-style (Team OLMo et al., 2024) decoder-only Transformer around 4B parameter with 29 blocks, Group Query Attention (GQA) (Ainslie et al., 2023) with 28 KV heads and 7 Q heads, ReLU^2 activation and aspect ratio 4. We trained at context length 1024 in order to keep attention cost manageable, with tied embeddings.

The model uses Quartet II NVFP4 linear layers for the bulk of dense compute (Panferov et al., 2026). In practice, the full recipe is not "full FP4:" as in prior low-precision training work, some numerically sensitive components remain in higher precision, such as norm layers, as well as the LM-head. *All* main transformer blocks use FP4 for their matrix multiplications in Q, K, V, attention out, up, and down projections. In Quartet II, the forward pass uses four-over-six (Cook et al., 2025) rounding during quantization, and the backward pass combines re-quantization of forward-pass matrices, Hadamard transform, and EDEN (Vargaftik et al., 2022) rounding for unbiased gradient computation.

The optimizer is Adam with a peak learning rate of 3×10^{-3}, a warmup of 2000 steps and a linear cooldown of 20000 steps. For intermediate checkpoint cooldowns, we used 5k steps, as the difference to 10k was minor for the first checkpoint, and the reduced number of steps keeps the training cost overhead of intermediate cooldowns manageable, affording to run cooldowns every 100k steps.

2.3. Run progress

Preliminary runs. The project reached the final configuration for the long run in two stages. First, we executed smaller 0.5B, 1.5B, and 4B runs to debug kernel integration, dataloading, and distributed evaluation. Second, we launched a complete 4B validation run on the 55B-token ClimbMix100m shard. That run completed 160k steps (83.9B sampled tokens) stably and finished with training loss 2.3389 and validation loss 2.2948. This run showed that the recipe and code could handle 4B-scale pretraining run.

Main run. The main run then targeted the fully tokenized ClimbMix corpus and executed 590k steps, i.e. 309.3B sampled training tokens. Early in the run, the system sustained roughly 434k tokens/s in aggregate (about 54k tokens/s/GPU); later checkpoints remained in the same general 50–54k tokens/s/GPU regime. At this throughput, the wall-clock duration is about eight days on a single $8 \times$ B300 server. The run produced checkpoints at 200k, 300k, 400k, 500k, and 590k steps, with optional 5k-step cooldown branches from several of the intermediate checkpoints.

The overall training trajectory was surprisingly smooth, despite NVFP4 training. We encountered infrastructure failures during ablations, including CUDA/NVLink issues on one node and a late NaN event near 288.8B tokens in one continuation. But the important point is that the main recipe did not show recurring loss spikes or consistent instability. In particular, we believe the NaN to be caused by a division where a small divisor was flushed to zero in the quantization code, and it did not reappear after guarding these divisions for subnormals.

One clear result is that the end-to-end speedup is (much) smaller than kernel-level speedup. In controlled comparisons, NVFP4 delivered roughly 25%–50% higher end-to-end throughput than BF16, depending on model size and machine configuration. This is because, once the dense matrix multiplications are accelerated, the optimizer, attention, communication, logging, and data movement become bottlenecks.

2.4. Final evaluations

Evaluation was performed offline by converting training checkpoints into a standard model format and running a standard zero-shot evaluation pipeline. Our eval suite is fairly narrow but standard, consisting of the ARC-Challenge, ARC-Easy, HellaSwag, and PIQA tasks. We chose this suite because it provides an early and relatively stable signal for models of this size, and because it is the subset for which historical GPT-3 and OPT-175B comparisons were available.

One important detail is that these baselines do not all use the same scoring convention, specifically for ARC. Since we do not have ARC values that are comparable between our GPT3 and OPT baselines, we report two ARC variants and therefore two corresponding averages. For readability, we call them the *OPT-style* and *GPT-3-style* aggregates.

Table 1 contains the main empirical picture. On the OPT-style aggregate, the final 590k checkpoint reaches 69.6, slightly above the 69.0 reference for OPT-175B. On the GPT-3-style aggregate, the same checkpoint reaches 68.9, below the GPT-3 175B reference of 70.0.

Evaluations were run using the EleutherAI Language Model Evaluation Harness (`lm-eval` v0.4.11) (Gao et al., 2024) in zero-shot scenario. Training checkpoints stored in PyTorch Distributed Checkpoint (DCP) format were first converted to flat `.pt` state dicts, then transformed into a HuggingFace-compatible model. The converted model was loaded via `accelerate launch` on in `bfloat16`, using the Quartet II pseudoquantization backend for inference. A custom `lm-eval` model wrapper pads all input sequences to multiples of 128 tokens (required by the Quartet II kernels).

For the ARC tasks, we use custom YAML task definitions that extend the standard `allenai/ai2_arc` dataset with the `acc_mutual_info` metric in addition to `acc` and `acc_norm`. All other tasks use the default `lm-eval` task definitions. All the tasks were evaluated using the default `lm-eval` templates.

The "OPT-style" ARC numbers use the `acc` metric, while the "GPT-3-style" ARC numbers use `acc_mutual_info` from the same tasks. In Table 1, HellaSwag and PIQA are reported with `acc_norm`. Accordingly, `Avg (OPT Eval.)` averages ARC-C `acc`, ARC-E `acc`, HellaSwag `acc_norm`, and PIQA `acc_norm`, whereas `Avg (GPT-3 Eval.)` averages ARC-C `acc_mutual_info`, ARC-E `acc_mutual_info`, HellaSwag `acc_norm`, and PIQA `acc_norm`. The OPT-175B baselines were sourced from the BigScience evaluation repository.

Table 2 shows results on additional benchmarks evaluated every 100k steps from the same checkpoints. These include Wikitext bits-per-byte (BPB), LAMBADA (OpenAI variant, `lambada_openai`), NQ (exact match). While the model improves steadily on all metrics, the gap to GPT-3 175B is significantly larger on knowledge-intensive and generative tasks than on the multiple-choice suite reported in Table 1. This is expected: the model has 44× fewer parameters and was trained with a 1024-token context, both of which limit knowledge storage and long-range coherence. Note that all CloverLM results in this table use the pseudoquantization backend, whereas the final 590k row in Table 1 was evaluated with real Quartet II NVFP4 kernels.

We also evaluated MMLU (Hendrycks et al., 2021) at the final checkpoint. Table 3 reports the results. Few-shot MMLU accuracy reaches 41.9%, substantially above the 31.8% reported for OPT-175B (per public leaderboards) and approaching the 43.9% reported in (Hendrycks et al., 2021) for GPT-3 175B (Brown et al., 2020). The "continuation" variant, which scores answer choices by continuation likelihood rather than full-sequence likelihood, yields slightly lower numbers (37.8% few-shot).

2.5. Cost estimation

The main run lasted about eight days on a single 8 × B300 node, corresponding to approximately 192 instance-hours or 1,536 GPU-hours. For cost accounting, the effective rate that applied when the experiment was actually run was $5.70 per GPU-hour. Equivalently, this is $45.60 per hour for the full 8-GPU node, which yields an estimated cost of $8,755.20 for the full run. Note that the cloud provider currently lists higher B300 pricing at $6.99 per GPU-hour ondemand (Verda, 2026) (See Table 4.) The true project cost is higher once one includes the earlier 83.9B-token validation run, smaller debugging runs, shape ablations, cooldown jobs, and separate evaluation machines. We estimate this cost to an additional $2k.

3. Conclusion

This report documents a "speedrun" attempt for a model of the GPT3-175B quality, finding that, if one combines a strong public data mixture and a working native NVFP4 recipe, a roughly 4B parameter model can be pretrained stably for about 309.3B tokens and reach a compact zero-shot average that is at least competitive with OPT-175B, at a core-run cost that is low by historical standards.

We caveat our findings on multiple points: We fall slightly short of GPT-3 parity, and the current recipe is clearly stronger on short-context multiple-choice tasks than on open-ended generation, suggesting somewhat biased data selection.

Even with these caveats, the results suggest that a combination of good data, native low precision, and careful

Table 1. Compact zero-shot comparison across checkpoints. For ARC we report two variants because historical baselines use different conventions: OPT uses `acc`, GPT-3 uses `acc_mutual_info`. HellaSwag and PIQA use `acc_norm`.

	ARC-C		ARC-E				Average	
Checkpoint	OPT	GPT-3	OPT	GPT-3	HellaSwag	PIQA	OPT Eval.	GPT-3 Eval.
100k	41.8	46.4	76.2	68.1	65.5	78.8	65.6	64.7
100k + 5k cooldown	43.7	47.4	77.0	68.4	67.0	79.2	66.7	65.5
200k	44.0	48.5	77.8	69.0	68.5	79.7	67.5	66.4
200k + 5k cooldown	44.9	48.0	78.3	70.5	69.2	80.2	68.2	67.0
300k	44.1	47.4	78.7	70.5	69.4	80.3	68.1	66.9
300k + 5k cooldown	46.1	48.1	79.0	72.4	70.2	79.3	68.6	67.5
400k	45.6	48.6	78.9	71.4	70.3	79.9	68.7	67.6
400k + 5k cooldown	44.5	48.7	79.4	71.1	71.1	79.8	68.7	67.7
500k	45.6	50.9	79.4	71.5	71.3	79.5	68.9	68.3
500k + 5k cooldown	44.9	49.6	79.5	71.3	70.8	80.4	68.9	68.0
590k	**46.3**	50.9	**80.0**	**72.4**	71.7	80.6	**69.6**	68.9
OPT-175B	41.2	–	75.1	–	78.3	**81.2**	69.0	–
GPT-3 175B	–	**51.4**	–	68.8	**78.9**	81.0	–	**70.0**

Table 2. Extended zero-shot evaluation results. Wiki BPB is Wikitext bits-per-byte. LAMBADA uses the OpenAI variant (`lambada_openai`). GPT-3 baselines are from (Brown et al., 2020).

Step	Wiki BPB ↓	LAMBADA acc ↑	NQ EM ↑
100k	0.777	55.9	4.6
200k	0.756	57.3	5.7
300k	0.745	58.6	5.9
400k	0.739	58.8	6.9
500k	0.734	59.6	7.5
590k	**0.723**	**61.1**	**7.8**
GPT-3 175B	–	**76.2**	**14.6**

Table 3. MMLU results at the final checkpoint.

	MMLU		MMLU (continuation)	
	0-shot	few-shot	0-shot	few-shot
Humanities	35.4	35.7	30.1	30.2
Social Sciences	42.1	47.1	40.8	42.4
STEM	37.2	39.0	35.5	35.8
Other	45.2	49.1	45.5	46.9
Overall	39.4	**41.9**	37.1	37.8
OPT-175B	–	31.8	–	–
GPT-3 175B	–	**43.9**	–	–

Table 4. End-to-end run cost for the 309.3B-token training run.

Scenario	GPU-hours	Rate	Cost
Standard	1,536	$5.70 / GPU-h	$8,755.20
Spot	1,536	$3.00 / GPU-h	$4,608.00

engineering can compress pretraining cost significantly. In future work, we plan to extend our study to larger model sizes, and potentially different architectures.

Acknowledgements

The authors would like Verda Cloud for computational support, and in particular Paul Chang for his consistent, prompt and generous help throughout the project. We also thank Jen Iofinova for performing basic safety testing on our model. Andrei Panferov and Erik Schultheis are supported in part by the BilAI (Biliteral AI) Austrian Center of Excellence in Artificial Intelligence, while Georgios Vlassis is supported in part by SwissAI.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Ainslie, J., Lee-Thorp, J., De Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on*

Empirical Methods in Natural Language Processing, pp. 4895–4901, 2023.

Allal, L. B., Lozhkov, A., Bakouch, E., et al. SmolLM – blazingly fast and remarkably powerful. Hugging Face blog, 2024. URL https://huggingface.co/blog/smollm.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Cook, J., Guo, J., Xiao, G., Lin, Y., and Ha, S. Four over six: More accurate NVFP4 quantization with adaptive block scaling. *arXiv preprint arXiv:2512.02010*, 2025.

Diao, S., Yang, Y., Fu, Y., Dong, X., Su, D., Kliegl, M., Chen, Z., Belcak, P., Suhara, Y., Yin, H., Patwary, M., Lin, C., Kautz, J., and Molchanov, P. CLIMB: CLustering-based iterative data mixture bootstrapping for language model pre-training. *arXiv preprint arXiv:2504.13161*, 2025.

Forsythe, A. TokenMonster: Ungreedy subword tokenizer and vocabulary trainer for python, go & javascript. GitHub repository, 2023. URL https://github.com/alasdairforsythe/tokenmonster.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation. Zenodo, 2024. URL https://zenodo.org/records/10256836.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2021.

Panferov, A., Schultheis, E., Tabesh, S., and Alistarh, D. Quartet II: Accurate LLM pre-training in NVFP4 by improved unbiased gradient estimation. *arXiv preprint arXiv:2601.22813*, 2026.

Su, D., Kong, K., Lin, Y., Jennings, J., Norick, B., Kliegl, M., Patwary, M., Shoeybi, M., and Catanzaro, B. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2459–2475, 2025.

Team OLMo, Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Vargaftik, S., Ben Basat, R., Portnoy, A., Mendelson, G., Ben Itzhak, Y., and Mitzenmacher, M. Eden: Communication-efficient and robust distributed mean estimation for federated learning. In *International Conference on Machine Learning*, 2022.

Verda. NVIDIA® B300 SXM6 product page. Product page, 2026. URL https://verda.com/b300. Accessed March 2026.

Vlassis, G., Ashkboos, S., Volkova, A., Hoefler, T., and Alistarh, D. Beyond outliers: A study of optimizers under quantization. *arXiv preprint arXiv:2509.23500*, 2025.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.