IN5640: Large Language Models Development
LLM Training III:
Mid-training, Continual Training, Context Extension

Stephan Oepen, Andrey Kutuzov

Language Technology Group, University of Oslo

October 12, 2026



Contents

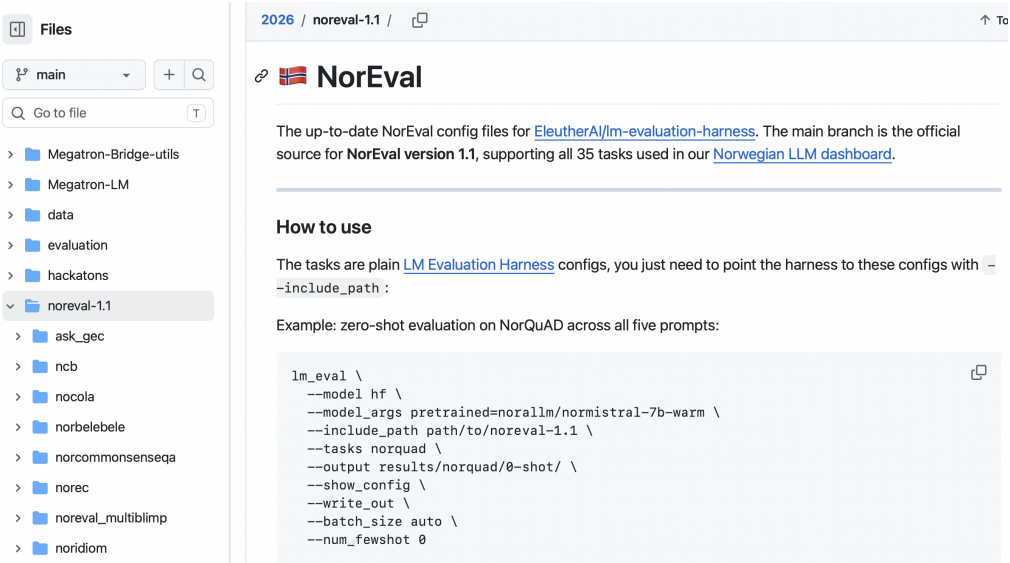
1. Mid-Way Internship Survey
2. Reflections from Team Consultations
3. Mid-Training, Adaptive Annealing
4. Continual Pre-training
   - Three-stage continual training (warm-starting)
5. Context Extension
6. Summary & Follow-Up Reading

A Preliminary Summary from our Point of View

Contents

1. Mid-Way Internship Survey
2. Reflections from Team Consultations
3. Mid-Training, Adaptive Annealing
4. Continual Pre-training
   - Three-stage continual training (warm-starting)
5. Context Extension
6. Summary & Follow-Up Reading

Ongoing Work in Refining and Extending NorEval



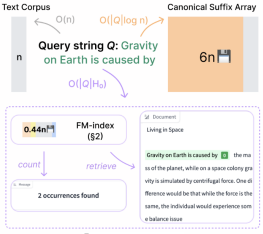
Decontamination Matters (If Maybe Less So to Us)

INFINI-GRAM MINI: Exact n-gram Search at the Internet Scale with FM-Index

Hao Xu[♡] **Jiacheng Liu**[♡♠] **Yejin Choi**[♢] **Noah A. Smith**[♡♠] **Hannaneh Hajishirzi**[♡♠]
[♡]Paul G. Allen School of Computer Science & Engineering, University of Washington
[♠]Allen Institute for AI [♢]Stanford University

Abstract

Language models are trained mainly on massive text data from the Internet, and it becomes increasingly important to understand this data source. Exact-match search engines enable searching in large text corpora – counting string appearances and retrieving the enclosing documents – yet the high storage overhead hinders their application on Internet-scale data. We present INFINI-GRAM MINI, an efficient and scalable system that can make petabyte-level text corpora searchable. Based on the FM-index data structure (Ferragina and Manzini, 2000), which simultaneously indexes and compresses text, our system creates indexes with



INFINI-GRAM MINI

	Test Size	Pile train	DCLM baseline	CC 2025-05	CC 2025-08	CC 2025-13	CC 2025-18	CC 2025-21	CC 2025-26
Knowledge and Reasoning									
MMLU	1000	13.20	28.40	13.50	9.00	12.10	11.50	11.70	9.20
MMLU-Pro	1000	5.50	16.20	7.10	5.40	6.00	6.30	7.40	6.90
BigBenchHard	1000	0.00	0.10	1.40	1.40	3.20	2.30	1.80	1.70
AGIEval	1000	0.80	3.10	2.70	3.60	3.00	7.00	9.40	4.60
GPQA	448	0.00	0.00	0.90	2.00	1.30	0.70	0.90	2.70
HLE	881	0.00	0.30	0.10	0.00	0.10	0.00	0.00	0.00
Math									
AIME-2024	30	0.00	0.00	10.00	3.30	6.70	40.00	40.00	13.30
GSM8K	1000	0.00	5.00	5.00	0.80	6.90	0.70	74.20	7.30
MATH-500	500	0.60	3.20	0.60	7.80	0.80	0.80	0.80	8.20
MGSM	250	0.00	0.00	5.60	1.60	35.60	0.80	72.80	6.00
Code									
HumanEval	164	0.00	0.00	0.00	0.60	0.60	0.60	0.00	0.00
HumanEval+	164	0.00	0.00	0.00	0.60	0.60	0.60	0.00	0.00
LiveCodeBench	880	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SWE-bench	500	0.00	0.00	0.20	0.20	0.00	0.00	0.00	0.00
MBPP	500	0.00	0.40	1.00	1.40	1.20	1.80	1.00	1.40
Commonsense Understanding									
ARC-Challenge	1000	1.80	34.10	11.90	4.00	3.10	3.80	4.20	4.80
ARC-Easy	1000	1.30	31.70	5.40	9.50	5.50	5.50	6.10	6.20
CSQA	1000	0.10	1.00	0.10	0.10	0.20	0.10	0.00	0.10
HellaSwag	1000	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10
OpenbookQA	500	10.80	15.60	14.60	30.20	13.20	13.40	13.20	12.20
Social IQa	1000	0.00	0.50	0.20	4.40	0.20	0.30	0.20	0.10
WinoGrande	1000	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Reading Comprehension									
CoQA	500	8.00	18.40	7.40	8.80	8.60	7.20	7.60	8.80
SQuAD	1000	2.80	40.10	2.70	33.00	10.10	1.50	2.00	8.50

Personally Identifiable Information (PII)

PRIVACY REQUIREMENTS IN NATIONAL AND EUROPEAN REGULATIONS

Personally Identifiable Information (PII)

PRIVACY REQUIREMENTS IN NATIONAL AND EUROPEAN REGULATIONS

CREDIT CARD NUMBERS

AUTHENTICATION TOKENS

INTERNET AND WEB ADDRESSES

(PRIVATE) EMAIL ADDRESSES

(PRIVATE) PHONE NUMBERS

(RESIDENTIAL) STREET ADDRESSES

PERSONAL NAMES

...

Personally Identifiable Information (PII)

PRIVACY REQUIREMENTS IN NATIONAL AND EUROPEAN REGULATIONS

CREDIT CARD NUMBERS

AUTHENTICATION TOKENS

INTERNET AND WEB ADDRESSES

(PRIVATE) EMAIL ADDRESSES

(PRIVATE) PHONE NUMBERS

(RESIDENTIAL) STREET ADDRESSES

PERSONAL NAMES

...

REGULAR EXPRESSIONS OR SEQUENCE LABELING

CHALLENGES: MULTILINGUALITY AND FALSE POSITIVES

Personally Identifiable Information (PII)

PRIVACY REQUIREMENTS IN NATIONAL AND EUROPEAN REGULATIONS

CREDIT CARD NUMBERS

AUTHENTICATION TOKENS

INTERNET AND WEB ADDRESSES

(PRIVATE) EMAIL ADDRESSES

(PRIVATE) PHONE NUMBERS

(RESIDENTIAL) STREET ADDRESSES

PERSONAL NAMES

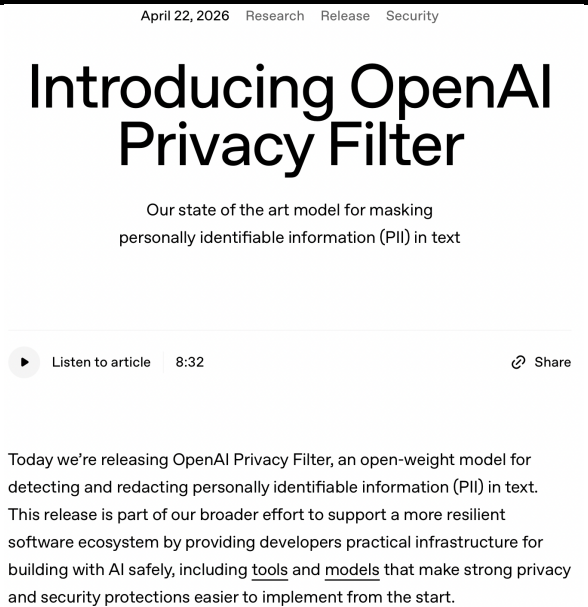
...

REGULAR EXPRESSIONS OR SEQUENCE LABELING

CHALLENGES: MULTILINGUALITY AND FALSE POSITIVES

TWO STRATEGY CHOICES: ANNOTATION AND MASKING

For Example: A 1.5B MOE (50M Active Parameters)



April 22, 2026 Research Release Security

Introducing OpenAI Privacy Filter

Our state of the art model for masking personally identifiable information (PII) in text

Listen to article 8:32 Share

Today we're releasing OpenAI Privacy Filter, an open-weight model for detecting and redacting personally identifiable information (PII) in text. This release is part of our broader effort to support a more resilient software ecosystem by providing developers practical infrastructure for building with AI safely, including tools and models that make strong privacy and security protections easier to implement from the start.

Contents

1. Mid-Way Internship Survey
2. Reflections from Team Consultations
3. Mid-Training, Adaptive Annealing
4. Continual Pre-training
   - Three-stage continual training (warm-starting)
5. Context Extension
6. Summary & Follow-Up Reading

Goals for Today: Beyond Pretraining

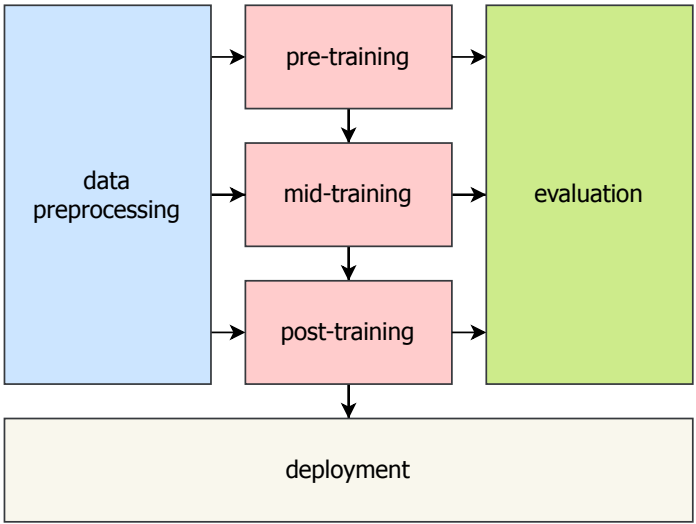
Mid-Training

Modern LLM pretraining typically involves multiple stages with different data mixtures, often followed by a final phase to extend context length. For example, Qwen3 (A. Yang, Li, et al., 2025) uses a three-stage approach: a general stage on 30T tokens at 4k context length, a reasoning stage with 5T higher-quality tokens emphasising STEM and coding, and finally a long context stage on hundreds of billions of tokens at 32k context length. SmolLM3 follows a similar philosophy, with planned interventions to introduce higher-quality datasets and extend context alongside reactive adjustments based on performance monitoring.

As we explained in the previous chapter, the data mixture doesn't have to stay fixed throughout training. Multi-stage training allows us to strategically shift dataset proportions as training progresses. Some interventions are planned from the start: For SmolLM3, we knew we'd introduce higher-quality FineMath4+ and Stack-Edu in stage 2, then add curated Q&A and reasoning data during the final decay phase. Other interventions are reactive, driven by monitoring performance during training. For example, in SmolLM2, when we found math and code performance lagging behind our targets, we curated entirely new datasets (FineMath and Stack-Edu) and introduced them mid-training. This flexibility—whether following a planned curriculum or adapting to emerging gaps—is what allows us to maximize the value of our compute budget.

`https://huggingface.co/spaces/HuggingFaceTB/smol-training-playbook#mid-training`

Recall: The LLM Development Life Cycle



Somewhat More Specifically: Olmo 3, for Example

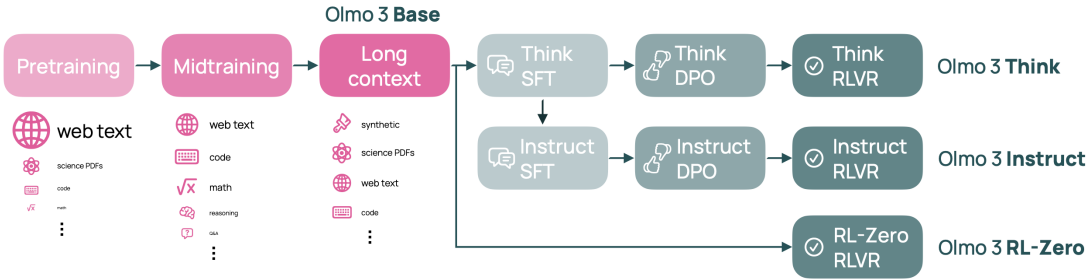
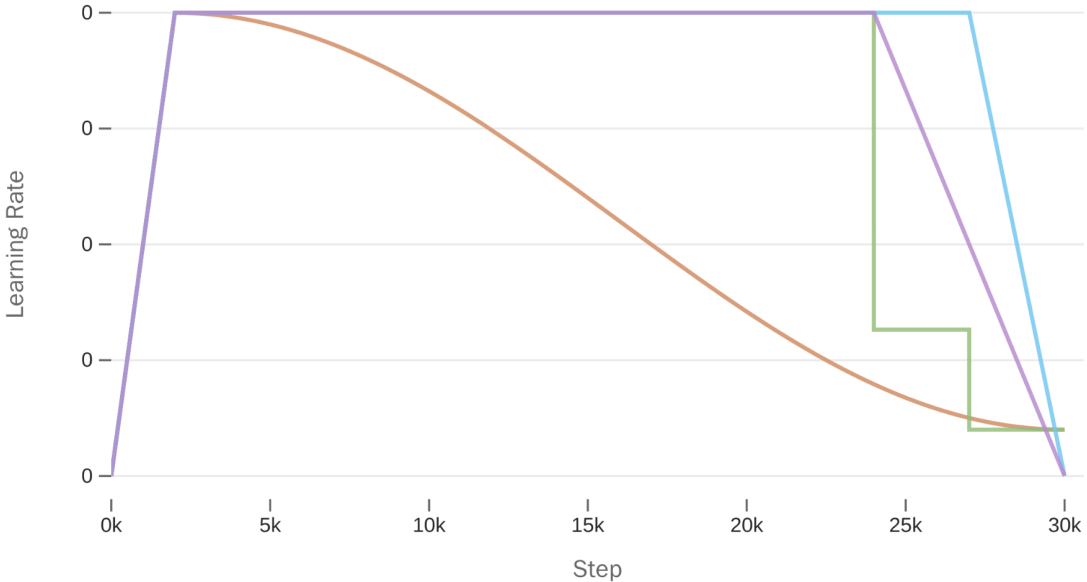


Figure 2 Depiction of model flow for Olmo 3. Development is divided into major **base model training (left)** and **post-training (right)** stages, each further divided into sub-stages with their own recipes (i.e., training data and method).

https://arxiv.org/abs/2512.13961

Recall: The WSD Learning Rate Schedule



Purely Metaphorically: Completing High School

CONSOLIDATE (MOST) RELEVANT KNOWLEDGE

PREPARE FOR SPECIALIZATION

IF NEED BE, "REPAIR"

MUCH SMALLER TOKEN BUDGET: SELECTIVITY

Purely Metaphorically: Completing High School

CONSOLIDATE (MOST) RELEVANT KNOWLEDGE

PREPARE FOR SPECIALIZATION

IF NEED BE, "REPAIR"

MUCH SMALLER TOKEN BUDGET: SELECTIVITY

BACK TO THE MOST FUNDAMENTAL QUESTIONS:

WHICH CAPABILITIES ARE IMPORTANT?

WHERE DO THINGS STAND?

Purely Metaphorically: Completing High School

CONSOLIDATE (MOST) RELEVANT KNOWLEDGE

PREPARE FOR SPECIALIZATION

IF NEED BE, "REPAIR"

MUCH SMALLER TOKEN BUDGET: SELECTIVITY

BACK TO THE MOST FUNDAMENTAL QUESTIONS:

WHICH CAPABILITIES ARE IMPORTANT?

WHERE DO THINGS STAND?

CHANGE IN DATA MIX IS BALANCING ACT (AGAIN):

SET DIRECTION, LIMIT CONFUSION

KEY CONSIDERATIONS: REPLAY, FRONT-LOADING

Somewhat More Specifically: Olmo 3, for Example

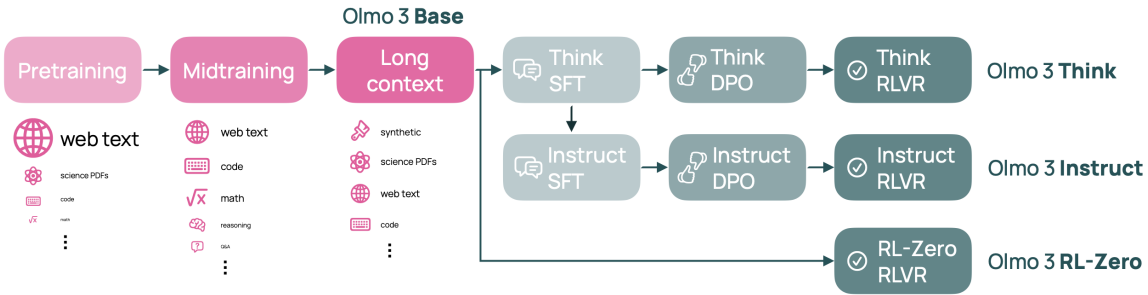
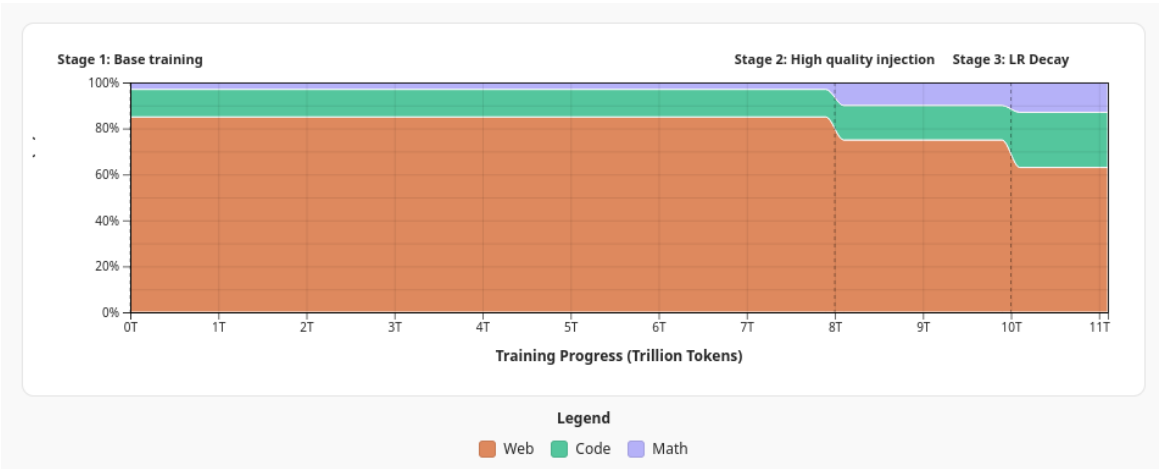


Figure 2 Depiction of model flow for Olmo 3. Development is divided into major **base model training (left)** and **post-training (right)** stages, each further divided into sub-stages with their own recipes (i.e., training data and method).

Abstractly Very Similiar: SmolLM3



The Boundaries are Increasingly Blurry

NVIDIA

2025-09-26

Front-Loading Reasoning: The Synergy between Pretraining and Post-Training Data

Syeda Nahida Akter[2 1], **Shrimai Prabhumoye**[1,3], **Eric Nyberg**[2], **Mostofa Patwary**[1], **Mohammad Shoeybi**[1], **Yejin Choi**[1,4], **Bryan Catanzaro**[1]
NVIDIA[1], Carnegie Mellon University[2], Boston University[3], Stanford University[4]
sakter@andrew.cmu.edu, sprabhumoye@nvidia.com

Abstract

The prevailing paradigm for enhancing the reasoning abilities of Large Language Models (LLMs) revolves around post-training on high-quality, reasoning-intensive data. While emerging literature suggests that reasoning data is increasingly incorporated also during the mid-training stage—a practice that is relatively more proprietary and less openly characterized—the role of such data in pretraining remains unclear. In particular, due to the opaqueness of pretraining corpora in most frontier models, the effect of reasoning data introduced at different phases of pre- and/or post-training is relatively less reported in

Annealing Ablations can be Comparatively Cheap (1 of 2)

OPENEUROLLM PRELUDE: 9B DENSE, 10T TOKENS;

EUROPE-CENTRIC MULTILINGUALIUTY: 36 LANGUAGES (INCLUDING NORWEGIAN)

"ENGLISH-FIRST" PRETRAINING: 80% ENGLISH, CODE, MATH

Annealing Ablations can be Comparatively Cheap (1 of 2)

OpenEuroLLM prelude: 9B dense, 10T tokens;

Europe-centric multilingualiuty: 36 languages (including Norwegian)

"English-first" pretraining: 80% English, code, math

vary "dominant" and "multilingual" components separately

three variants each: less web text; higher multilingual quality

micro-annealing after 8T tokens (inverse sqrt decay):

four scales: 20B, 50B, 100B, (and 300B) tokens

two sets of benmarks: `dclm-core-22`, `oellm-multilingual-eu`:

Annealing Ablations can be Comparatively Cheap (2 of 2)

mix	dclm 20bt	eu 20bt	dclm 50bt	eu 50bt	dclm 100bt	eu 100bt
baseline	0.303	-0.295	0.527	0.053	0.434	-0.228
dominant1	0.355	0.448	0.464	0.844	0.559	0.622
dominant2	0.401	0.826	0.623	0.625	0.694	0.742
dominant3	0.112	0.699	0.539	0.611	0.361	0.998
baseline	0.303	-0.295	0.527	0.053	0.434	-0.228
multilingual1	0.358	-0.485	0.564	-0.061	0.720	0.206
multilingual2	0.285	0.019	0.209	0.167	0.391	0.210
multilingual3	-0.123	0.044	0.205	0.387	0.498	0.111

WSM: Checkpoint Merging instead of Annealing (1 of 3)

WSM: Decay-Free Learning Rate Schedule via Checkpoint Merging for LLM Pre-training

Changxin Tian[1]*, **Jiapeng Wang**[1,2,3*‡], **Qian Zhao**[1], **Kunlong Chen**[1], **Jia Liu**[1], **Ziqi Liu**[1], **Jiaxin Mao**[2,3], **Wayne Xin Zhao**[2,3‡], **Zhiqiang Zhang**[1†], **Jun Zhou**[1]
[1]Ling Team, Ant Group [2]Gaoling School of Artificial Intelligence, Renmin University of China
[3]Beijing Key Laboratory of Research on Large Models and Intelligent Governance
tianchangxin.tcx@antgroup.com, wangjp1010@ruc.edu.cn

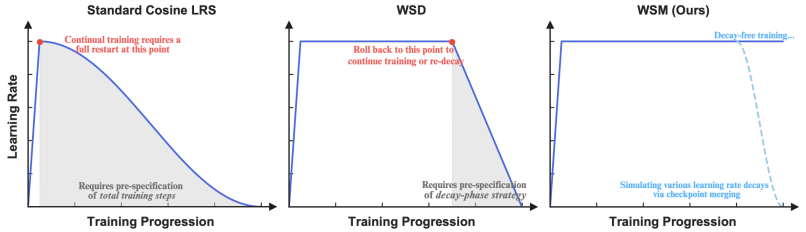
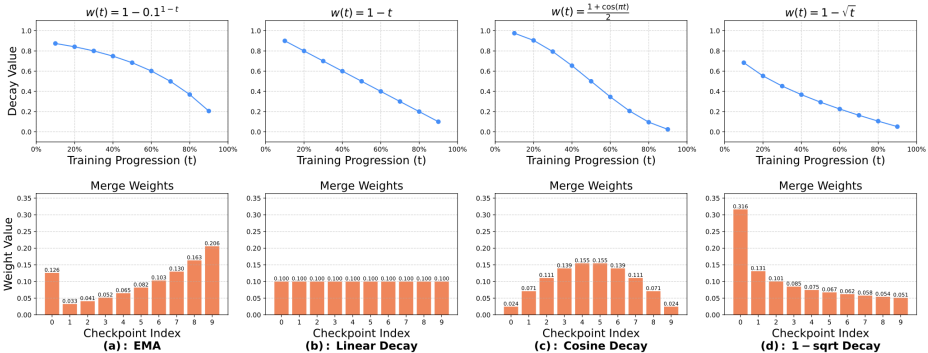


Figure 1: Comparison between our proposed WSM (Warmup-Stable and Merge) and mainstream learning rate scheduling strategies, Cosine (Loshchilov & Hutter, 2016) and WSD (Hu et al., 2024).

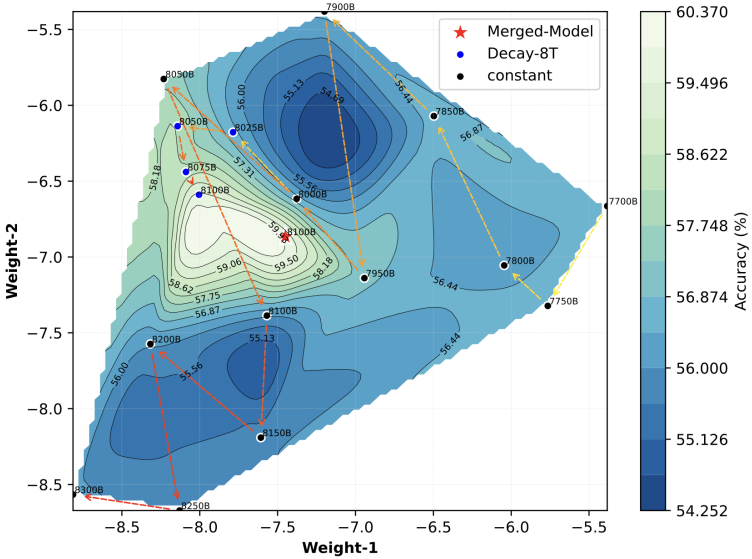
WSM: Checkpoint Merging instead of Annealing (2 of 3)

The core idea of checkpoint merging in this work is to take an ordered list of checkpoints, $[\theta_n, \theta_{n+1}, \ldots, \theta_{n+k}]$, and apply a merge function to generate a single model $\hat{\theta}_{n+k}$. Here, $\theta_i \in \mathbb{R}^d$ represents the model's parameter vector at the i-th training iteration. The most general form is a weighted average of the checkpoints:

$$\hat{\theta}_{n+k} = \sum_{j=0}^{k} c_j \theta_{n+j} \tag{1}$$



WSM: Checkpoint Merging instead of Annealing (3 of 3)



Contents

1 Mid-Way Internship Survey

2 Reflections from Team Consultations

3 Mid-Training, Adaptive Annealing

4 Continual Pre-training
- Three-stage continual training (warm-starting)

5 Context Extension

6 Summary & Follow-Up Reading

Small languages, big models: NorMistral-11B

What is the problem we are solving?

- Problem: not enough training data for lower-resource languages and/or domains
- Solution: continual pre-training (a.k.a. warm-starting) of strong models pre-trained on data in other languages/domains (usually English or multilingual) (Gururangan et al., 2020)
- Let's look at NorMistral-11B model (`https://hf.co/ltg/normistral-11b-warm`)
  - released two years ago, but the fundamentals are the same
  - more modern work in progress: `https://hf.co/HPLT/NorOLMo-13B`

Small languages, big models: NorMistral-11B

What is the problem we are solving?

- Problem: not enough training data for lower-resource languages and/or domains
- Solution: continual pre-training (a.k.a. warm-starting) of strong models pre-trained on data in other languages/domains (usually English or multilingual) (Gururangan et al., 2020)
- Let's look at NorMistral-11B model (`https://hf.co/ltg/normistral-11b-warm`)
  - released two years ago, but the fundamentals are the same
  - more modern work in progress: `https://hf.co/HPLT/NorOLMo-13B`

- Three orders of magnitude less data than English for Norwegian languages
  - In 2024, 24 billion words available for Norwegian (Bokmål and Nynorsk) and just 40 million words for Sámi
  - Scaling laws: this is enough only for a 1B size model! (Hoffmann et al., 2022)
  - *(in 2026, it can be 35 billion; does not change the picture)*

Small languages, big models: NorMistral-11B

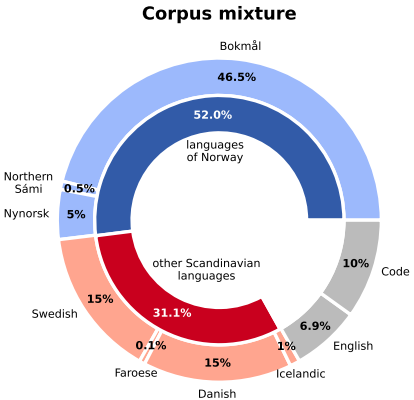
What is the problem we are solving?

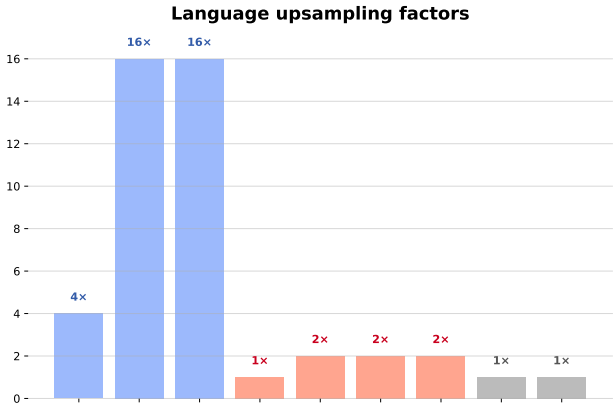
- Problem: not enough training data for lower-resource languages and/or domains
- Solution: continual pre-training (a.k.a. warm-starting) of strong models pre-trained on data in other languages/domains (usually English or multilingual) (Gururangan et al., 2020)
- Let's look at NorMistral-11B model (`https://hf.co/ltg/normistral-11b-warm`)
  - released two years ago, but the fundamentals are the same
  - more modern work in progress: `https://hf.co/HPLT/NorOLMo-13B`

- Three orders of magnitude less data than English for Norwegian languages
  - In 2024, 24 billion words available for Norwegian (Bokmål and Nynorsk) and just 40 million words for Sámi
  - Scaling laws: this is enough only for a 1B size model! (Hoffmann et al., 2022)
  - *(in 2026, it can be 35 billion; does not change the picture)*
- NorMistral 11B solved this by:
  1. continual pre-training: knowledge transfer from an existing models,
  2. data augmentation with texts from related languages,
  3. targeted upsampling, following (Muennighoff et al., 2023)

Continual Pre-training

Overview of NorMistral-11B pre-training data:





The resulting corpus of 250B non-unique tokens allows to train an 11B size model

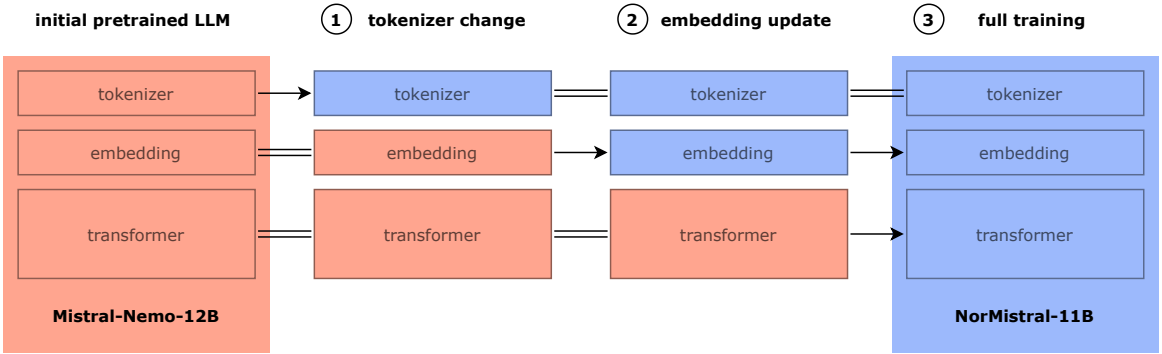
(Samuel et al., 2025)

Continual Pre-training

Training data composition

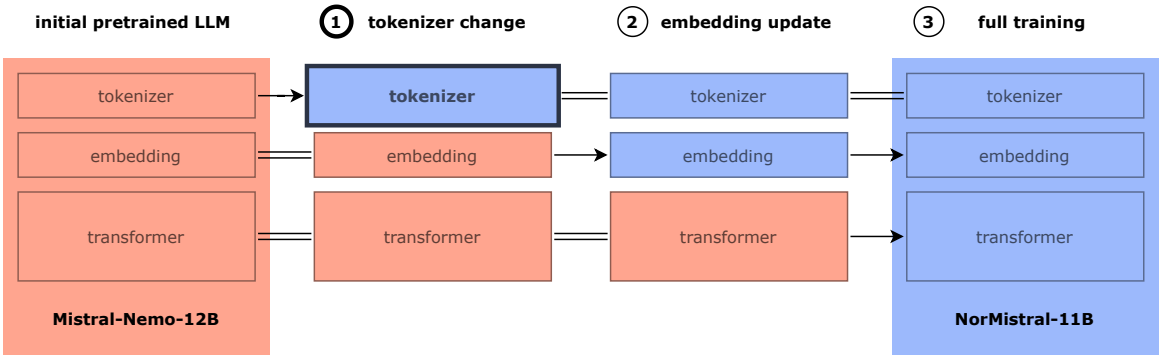
1. Mímir Core corpus (based on the Norwegian Colossal Corpus, NCC) (de la Rosa et al., 2025)
2. CulturaX (Nguyen et al., 2023) (not used much any more)
3. HPLT v1.2 (de Gibert et al., 2024) (**deprecated now, do not use!**)
4. high-quality English from FineWeb-edu (Lozhkov et al., 2024a)
5. code from Stack v2 (Lozhkov et al., 2024b)
6. Faroese and Northern Sámi from Glot500 (Imani et al., 2023)
7. Northern Sámi from the SIKOR free corpus (Giellatekno and Divvun, 2016)
8. own web crawl by LTG for Northern Sámi (about 13M words)

Three-stage continual training (warm-starting)



The original `Mistral-Nemo-Base-2407` is a somewhat-multilingual `Llama`-like model by (Jiang et al., 2023). How does one make it a Norwegian LLM?

Three-stage continual training (warm-starting)



Stage 1: Tokenizer change

Three-stage continual training (warm-starting)

Stage 1: Tokenizer change

- Non-optimal tokenizer irreversibly hurts training and inference efficiency
  - **Mistral-Nemo-12B** tokens: `_O la _og _Kari _sp iste _brun ost _på _h yt ta _i _fj orden`
  - **NorMistral-11B** tokens: `_Ola _og _Kari _spiste _brun ost _på _hytta _i _fjorden`

Three-stage continual training (warm-starting)

Stage 1: Tokenizer change

- Non-optimal tokenizer irreversibly hurts training and inference efficiency
  - **Mistral-Nemo-12B** tokens: `_O la _og _Kari _sp iste _brun ost _på _h yt ta _i _fj orden`
  - **NorMistral-11B** tokens: `_Ola _og _Kari _spiste _brun ost _på _hytta _i _fjorden`

Tokenizer	# subword tokens	NOB	NNO	SME
Mistral-Nemo-12B	131 072	1.79	1.87	2.63
NorMistral-11B	51 200	1.22	1.28	1.82

***Tokenizer statistics**: The vocabulary size and fertility of tokenizers for Bokmål (NOB), Nynorsk (NNO), and Northern Sámi (SME).*

Three-stage continual training (warm-starting)

Stage 1: Tokenizer change

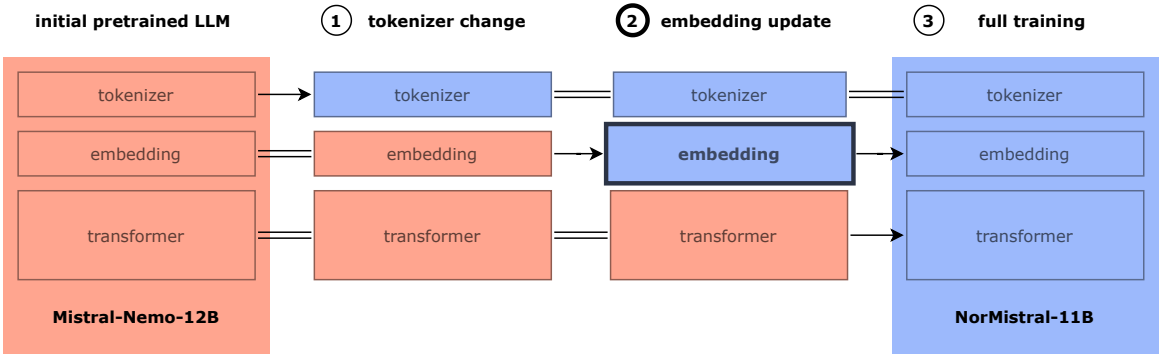
- Non-optimal tokenizer irreversibly hurts training and inference efficiency
  - **Mistral-Nemo-12B** tokens: `_O la _og _Kari _sp iste _brun ost _på _h yt ta _i _fj orden`
  - **NorMistral-11B** tokens: `_Ola _og _Kari _spiste _brun ost _på _hytta _i _fjorden`

Tokenizer	# subword tokens	NOB	NNO	SME
Mistral-Nemo-12B	131 072	1.79	1.87	2.63
NorMistral-11B	51 200	1.22	1.28	1.82

***Tokenizer statistics**: The vocabulary size and fertility of tokenizers for Bokmål (NOB), Nynorsk (NNO), and Northern Sámi (SME).*

- The first step is thus training a brand new tokenizer for the target language(s)
- The whole procedure is inspired by (de Vries and Nissim, 2021)

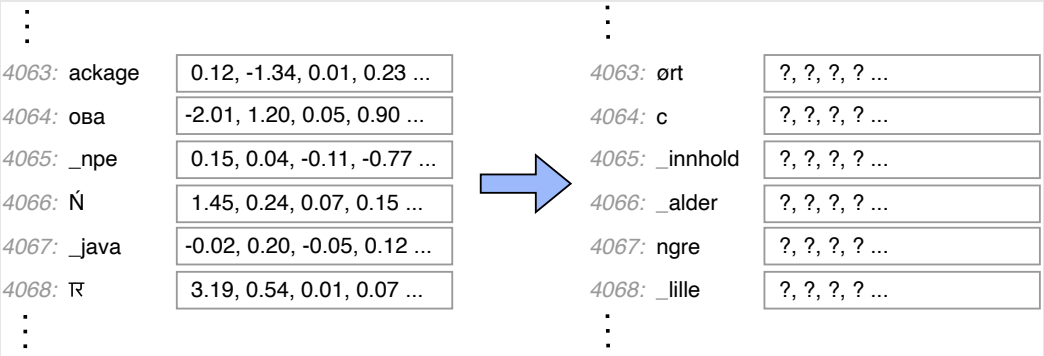
Three-stage continual training (warm-starting)



Three-stage continual training (warm-starting)

Stage 2: Embedding update

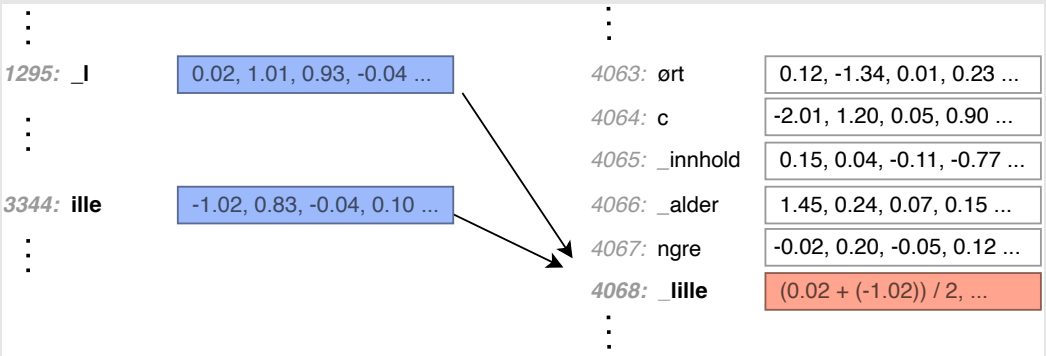
- Replacing the tokenizer completely ruins the subword embeddings, we need to re-align them for the new subwords



Three-stage continual training (warm-starting)

Stage 2: Embedding update

- Replacing the tokenizer completely ruins the subword embeddings, we need to re-align them for the new subwords
- First, initialization from the original embeddings:

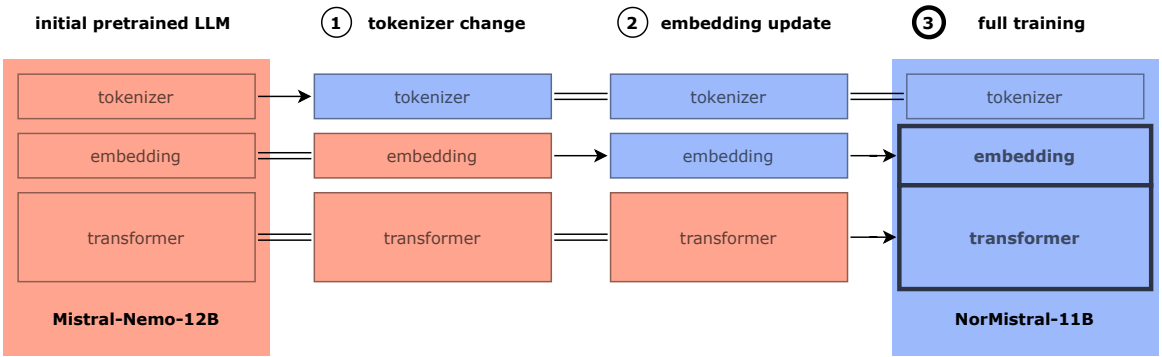


Three-stage continual training (warm-starting)

Stage 2: Embedding update

- Replacing the tokenizer completely ruins the subword embeddings, we need to re-align them for the new subwords
- First, initialization from the original embeddings
- Then we continually train the model with all parameters frozen – except for the input/output embeddings
- Now all subwords are in a familiar embedding space, which prevents a sudden drop of performance during the third stage

Three-stage continual training (warm-starting)



Three-stage continual training (warm-starting)

Stage 3: Full training

- Finally, after updating the tokenizer and embeddings, full continual pre-training starts, re-using the non-embedding weights of `Mistral`
- Trained on 250 billion tokens (60 000 steps of $1\,024 \times 4\,096$ tokens)
- WSD learning-rate schedule, 1 000 warm-up steps and 10 000 decay steps
- AdamW optimizer, weight decay of 0.1, no dropout
- `bfloat16` precision
- Training cost: $1.7 \cdot 10^{22}$ FLOPs, with an average of 38% model FLOP/s utilization (MFU)
- Pre-training took place on LUMI HPC cluster: 256 AMD MI250X GPUs
  - used 55 000 GPU hours in total (8.5 days of runtime)

Three-stage continual training (warm-starting)

Stage 3: Full training

- Finally, after updating the tokenizer and embeddings, full continual pre-training starts, re-using the non-embedding weights of `Mistral`
- Trained on 250 billion tokens (60 000 steps of $1\,024 \times 4\,096$ tokens)
- WSD learning-rate schedule, 1 000 warm-up steps and 10 000 decay steps
- AdamW optimizer, weight decay of 0.1, no dropout
- `bfloat16` precision
- Training cost: $1.7 \cdot 10^{22}$ FLOPs, with an average of 38% model FLOP/s utilization (MFU)
- Pre-training took place on LUMI HPC cluster: 256 AMD MI250X GPUs
  - used 55 000 GPU hours in total (8.5 days of runtime)

This is a ready-to-use recipe for continual pre-training (CPT).

Continual Pre-training

How does it compare to training from scratch in the final performance?

Training method	NorQuAD 1-shot	NoReC 16-shot	NRK-Quiz-QA 0-shot	Tatoeba EN-NB 16-shot
TRANSFORMER ARCHITECTURE				
BLOOM	43.6	67.6	44.6	52.2
Llama / Mistral	**43.7**	**80.3**	**48.2**	**53.4**
CONTINUAL TRAINING				
init. from scratch	43.7	80.3	48.2	53.4
three-stage continual	**64.8**	**84.9**	**57.9**	**57.2**
HYBRID TRAINING OBJECTIVE				
causal-only	67.0	86.0	**59.0**	**58.8**
hybrid masked-causal	**69.3**	**87.5**	55.4	58.2
TRAINING STEPS				
0 steps (base model)	76.5	86.9	47.4	49.6
0 steps (adapted tokenizer)	73.5	89.4	44.2	51.4
10,000 steps	69.3	87.5	55.4	58.2
20,000 steps	70.5	89.2	57.7	58.8
30,000 steps	66.2	82.3	59.0	58.5
40,000 steps	68.5	87.0	61.1	**58.9**
50,000 steps	70.4	88.7	60.2	58.7
60,000 steps	**76.7**	**90.5**	**63.7**	58.8

(Samuel et al., 2025)

Continual Pre-training

Replaying some of the previous data during CPT helps to prevent catastrophic forgetting:

Rules of thumb for continual pre-training

Caveat—The following guidelines are written to the best of our *current knowledge*.

Learning rate schedule:

- If the learning rate was cosine-decayed from a large value η_{max} to a small value η_{min} during pre-training on the initial dataset, the following guidelines can help to continually pre-train your model:
  - Re-warming and re-decaying the learning rate from $\mathcal{O}(\eta_{max})$ to $\mathcal{O}(\eta_{min})$ improves adaptation to a new dataset, e.g. compared to continuing from small learning rates $\mathcal{O}(\eta_{min})$.
  - Decreasing the schedule's maximum learning rate can help reduce forgetting, whereas increasing it can improve adaptation.
- Infinite LR schedules are promising alternatives to cosine decay schedules. They transition into a high constant learning rate across tasks, helping prevent optimization-related forgetting by avoiding re-warming the LR between tasks. They also avoid committing to a specific budget of tokens as a final exponential decay can be used to train the model to convergence at any point during training.

Replay:

- We find that even small amounts of replay are good at mitigating forgetting. We recommend experimenting with different replay fractions since relative differences between them appear very early during training. For example, one may experiment with different replay fractions for a limited token budget, using evaluations relevant to their use case, to find a sweet spot between adapting to the new data and mitigating performance loss due to the distribution shift.

(Ibrahim et al., 2024)

Contents

1. Mid-Way Internship Survey
2. Reflections from Team Consultations
3. Mid-Training, Adaptive Annealing
4. Continual Pre-training
   - Three-stage continual training (warm-starting)
5. Context Extension
6. Summary & Follow-Up Reading

Context Extension



Recall from the previous lecture:

- An optional 4th stage of LLM pre-training is long context extension
- Happens right before supervised fine-tuning (SFT)

Context Extension



Recall from the previous lecture:

- An optional 4th stage of LLM pre-training is long context extension
- Happens right before supervised fine-tuning (SFT)
- A 'standard' context length for base-level LLMs nowadays is 4096 tokens
  - i.e., cannot efficiently process sequences longer than that

Context Extension



Recall from the previous lecture:

- An optional 4th stage of LLM pre-training is long context extension
- Happens right before supervised fine-tuning (SFT)
- A 'standard' context length for base-level LLMs nowadays is 4096 tokens
  - i.e., cannot efficiently process sequences longer than that
- But processing long documents, large code bases and lengthy conversations requires more!
- Context size of more than 100K or even 1M tokens.

Context Extension



Recall from the previous lecture:

- An optional 4th stage of LLM pre-training is long context extension
- Happens right before supervised fine-tuning (SFT)
- A 'standard' context length for base-level LLMs nowadays is 4096 tokens
  - i.e., cannot efficiently process sequences longer than that
- But processing long documents, large code bases and lengthy conversations requires more!
- Context size of more than 100K or even 1M tokens.

How do we achieve that?

Context Extension

- To be able to process long documents, LLM must be trained on long token sequences
- But it is expensive to train on long-context spans from the very beginning
  - quadratic cost of attention with sequence length!
- also, the model must first learn short-length relationships between words

Context Extension

- To be able to process long documents, LLM must be trained on long token sequences
- But it is expensive to train on long-context spans from the very beginning
  - quadratic cost of attention with sequence length!
- also, the model must first learn short-length relationships between words
- Luckily, it is enough to do a short longer-context run late in the training (Gao et al., 2025)
- E.g., you mostly train on 4K contexts, and then do a short 'boost' of 64K contexts.
- ...or do sequential scaling from 4K to 32K to 64k, as in SmoLM.

Context Extension

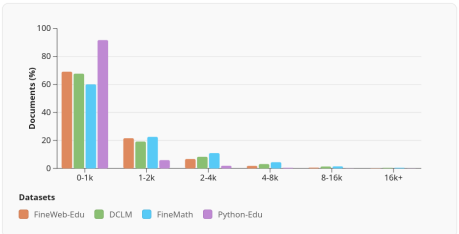
Long Context Data

- You'll need documents which are actually long
  - most documents in web-crawled datasets are less than 2K tokens in size

Context Extension

Long Context Data

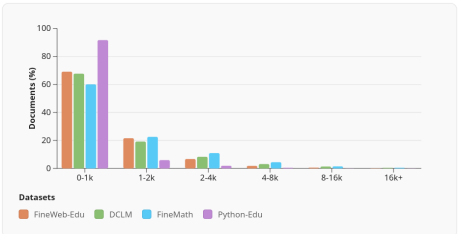
- You'll need documents which are actually long
  - most documents in web-crawled datasets are less than 2K tokens in size
- Upsampling of books, long manuals, code repos
- Synthetically generated long documents
- Mix long documents with short ones! (60/40%)



Context Extension

Long Context Data

- You'll need documents which are actually long
  - most documents in web-crawled datasets are less than 2K tokens in size
- Upsampling of books, long manuals, code repos
- Synthetically generated long documents
- Mix long documents with short ones! (60/40%)



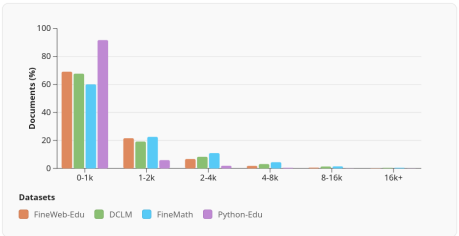
Model Config for Long Context

- It is not only data: you need to increase max sequence length in the model itself
- E.g., `NorOLMo-13B`: increase from 4096 to 16384 tokens in Stage 3.

Context Extension

Long Context Data

- You'll need documents which are actually long
  - most documents in web-crawled datasets are less than 2K tokens in size
- Upsampling of books, long manuals, code repos
- Synthetically generated long documents
- Mix long documents with short ones! (60/40%)



Model Config for Long Context

- It is not only data: you need to increase max sequence length in the model itself
- E.g., `NorOLMo-13B`: increase from 4096 to 16384 tokens in Stage 3.
- It entailed decreasing batch size from 2048 to 512 (preserving number of tokens per step)...
- ...and increasing RoPe base frequency from 500K to 2M, so that attention weights tokens far apart do not decay too rapidly (see more on positional encoding in Lecture 4).

Context Extension

You'll need benchmarks to determine the best long context extension approach. Measuring perplexity is not enough.
There are some well-established approaches:

Long Context Evaluations

- Needle in a haystack (NIAH)
  - 'recall a text span in a large corpus with distractors'
  - largely saturated

Context Extension

You'll need benchmarks to determine the best long context extension approach. Measuring perplexity is not enough.
There are some well-established approaches:

Long Context Evaluations

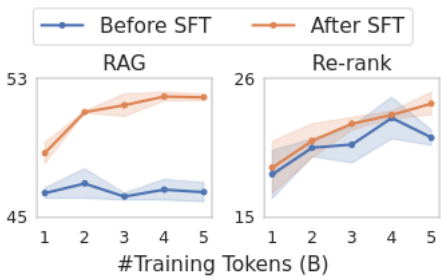
- Needle in a haystack (NIAH)
  - 'recall a text span in a large corpus with distractors'
  - largely saturated
- RULER (Hsieh et al., 2024)
  - synthetic; NIAH with diverse types and quantities of needles; proxies for real NLP tasks
- HELMET (Yen et al., 2025)
  - a set of natural tasks: RAG, QA, summarization, etc
  - model-based evaluations (LLM-as-a-judge)

Task	Configuration	Example
Single NIAH (S-NIAH)	type_key = word type_value = number type_haystack = essay size_haystack = context length	One of the special magic numbers for long-context is: 12345. What is the special magic number for long-context mentioned in the provided text? Answer: 12345
Multi-keys NIAH (MK-NIAH)	num_keys = 2 type_key = word type_value = number type_haystack = essay size_haystack = context length	One of the special magic numbers for long-context is: 12345. One of the special magic numbers for long-context is: 54321. What is the special magic number for long-context mentioned in the provided text? Answer: 12345
Multi-values NIAH (MV-NIAH)	num_values = 2 type_key = word type_value = number type_haystack = essay size_haystack = context length	One of the special magic numbers for long-context is: 12345. One of the special magic numbers for long-context is: 54321. What are all the special magic numbers for long-context mentioned in the provided text? Answer: 12345 54321

		HELMET		
Models	NIAH	Recall	RAG	Re-rank
Fu et al. (2024)	100	95.8	52.1	23.1
Llama-3.1-8B	100	99.4	56.3	37.0
Llama-3.1-70B	100	100	62.1	49.2

Tips and tricks

- NB: for some long-context tasks, performance improvements are visible only after SFT ! (Gao et al., 2025)
- Document masking is important: we don't want cross-document attention.



Contents

1. Mid-Way Internship Survey
2. Reflections from Team Consultations
3. Mid-Training, Adaptive Annealing
4. Continual Pre-training
   - Three-stage continual training (warm-starting)
5. Context Extension
6. Summary & Follow-Up Reading

Summary: Key Concepts from the Lecture Today

- Decontamination
- Personally identifiable information (PII)
- Mid-training
- Front-loading
- Checkpoint merging
- Continual pre-training
- Data replay
- Long-context extension

Follow-Up Reading: Ibrahim et al.(2024)

Published in Transactions on Machine Learning Research (06/2024)

Simple and Scalable Strategies to Continually Pre-train Large Language Models

Adam Ibrahim[*†‡] *research@adamibrahim.fr*
Benjamin Thérien[*†‡] *benjamin.therien@mila.quebec*
Kshitij Gupta[*†‡] *kshitij.gupta@mila.quebec*
Mats L. Richter[†‡] *mats.richter@mila.quebec*
Quentin Anthony[◇†§] *qubitquentin@gmail.com*
Timothée Lesort[†§] *t.lesort@gmail.com*
Eugene Belilovsky[‡§] *eugene.belilovsky@concordia.ca*
Irina Rish[†§] *irina.rish@mila.quebec*

Department of Computer Science and Operation Research, Université de Montréal, Montréal, Canada †
Department of Computer Science and Software Engineering, Concordia University, Montréal, Canada ‡
Mila, Montréal, Canada §
EleutherAI ◇

Reviewed on OpenReview: *https://openreview.net/forum?id=DimPeeCxKO*

Abstract

Large language models (LLMs) are routinely pre-trained on billions of tokens, only to start the process over again once new data becomes available. A much more efficient solution is to continually pre-train these models—saving significant compute compared to re-training. However, the distribution shift induced by new data typically results in degraded performance on previous data or poor adaptation to the new data. In this work, we show that a simple and scalable combination of learning rate (LR) re-warming, LR re-decaying, and replay of previous data is sufficient to match the performance of fully re-training from scratch on all available data, as measured by the final loss and the average score on several language model (LM) evaluation benchmarks. Specifically, we show this for a weak but realistic distribution shift between two commonly used LLM pre-training datasets (English→English) and a stronger distribution shift (English→German) at the 405M parameter model scale with large dataset sizes (hundreds of billions of tokens). Selecting the weak but realistic shift for larger-scale experiments, we also find that our continual learning strategies match the re-training baseline for a 10B parameter LLM. Our results demonstrate that autoregressive transformer-based LLMs can be successfully updated via simple and scalable continual learning strategies, matching the re-training baseline using only a fraction of the compute. Finally, inspired by previous work, we propose alternatives to the cosine learning rate schedule that help circumvent forgetting induced by LR re-warming and that are not bound to a fixed token budget.

https://openreview.net/forum?id=DimPeeCxKO

Follow-Up Reading: Samuel et al. (2025)

Small Languages, Big Models:
A Study of Continual Training on Languages of Norway

David Samuel Vladislav Mikhailov Erik Velldal Lilja Øvrelid
Lucas Charpentier Andrey Kutuzov Stephan Oepen
University of Oslo, Language Technology Group
davisamu@ifi.uio.no

Abstract

Training large language models requires vast amounts of data, posing a challenge for less widely spoken languages like Norwegian and even more so for truly low-resource languages like Northern Sámi. To address this issue, we present a novel three-stage continual training approach that substantially improves the downstream performance together with the inference efficiency for the target languages. Based on our findings, we train, evaluate, and openly release a new generative language model for Norwegian Bokmål, Nynorsk, and Northern Sámi with 11.4 billion parameters: *NorMistral-11B*.

1 Introduction

The development of large language models typically requires massive amounts of training data, which benefits wide-spread languages such as English, but poses a significant challenge for less

To address this data scarcity, we propose a novel approach combining three key elements: knowledge transfer from existing models, data augmentation with related languages, and targeted upsampling. This method enables us to train an 11.4B-parameter model that achieves state-of-the-art performance across Norwegian language tasks while obtaining strong capabilities in Northern Sámi. The three main research contributions of this paper can be summarized as follows:

1. **Novel training method for data-constrained language models** We propose a three-stage training method for efficient adaptation of existing language models to lower-resource languages. Our results demonstrate that this approach works well for adapting a Mistral model to Bokmål, Nynorsk and Northern Sámi. Our model achieves the *state-of-the-art* performance on tasks requiring deep linguistic understanding and world knowledge in Norwegian contexts – while being more than 30% faster than the original Mistral model on Norwegian inputs.

2. **Flexible masked-causal model** We train a

https://aclanthology.org/2025.nodalida-1.61/

Follow-Up Reading: Yen et al. (2025)

Published as a conference paper at ICLR 2025

HELMET: How to Evaluate Long-Context Language Models Effectively and Thoroughly

Howard Yen[P] **Tianyu Gao**[P] **Minmin Hou**[I] **Ke Ding**[I]
Daniel Fleischer[I] **Peter Izsak**[I] **Moshe Wasserblat**[I] **Danqi Chen**[P]
[P]Princeton Language and Intelligence, Princeton University [I]Intel
{hyen,tianyug,danqic}@cs.princeton.edu
https://princeton-nlp.github.io/HELMET

Abstract

Many benchmarks exist for evaluating long-context language models (LCLMs), yet developers often rely on synthetic tasks such as needle-in-a-haystack (NIAH) or an arbitrary subset of tasks. However, it remains unclear whether these benchmarks reflect the diverse downstream applications of LCLMs, and such inconsistencies further complicate model comparison. We investigate the underlying reasons behind these practices and find that existing benchmarks often provide noisy signals due to limited coverage of applications, insufficient context lengths, unreliable metrics, and incompatibility with base models. In this work, we introduce **HELMET** (**H**ow to **E**valuate **L**ong-context **M**odels **E**ffectively and **T**horoughly), a comprehensive benchmark encompassing seven diverse, application-centric categories. We also address several issues in previous benchmarks by adding controllable lengths up to 128K tokens, model-based evaluation for reliable metrics, and few-shot prompting for robustly evaluating base models. Consequently, we demonstrate that HELMET offers more reliable and consistent rankings of frontier LCLMs. Through a comprehensive study of 59 LCLMs, we find that (1) synthetic tasks like NIAH do not reliably predict downstream performance; (2) the diverse categories in HELMET exhibit distinct trends and low correlations with each other; and (3) while most LCLMs achieve perfect NIAH scores, open-source models significantly lag behind closed ones when tasks require full-context reasoning or following complex instructions—the gap widens as length increases. Finally, we recommend using our RAG tasks for fast model development, as they are easy to run and better predict other downstream performance; ultimately, we advocate for a holistic evaluation across diverse tasks.[1]

https://proceedings.iclr.cc/paper_files/paper/2025/hash/f5332c8273d02729730a9c24dec2135e-Abstract-Conference.html

References I

de Gibert, O., Nail, G., Arefyev, N., Bañón, M., van der Linde, J., Ji, S., Zaragoza-Bernabeu, J., Aulamo, M., Ramírez-Sánchez, G., Kutuzov, A., Pyysalo, S., Oepen, S., and Tiedemann, J. (2024). A new massive multilingual dataset for high-performance language technologies. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 1116–1128, Torino, Italia. ELRA and ICCL.

de la Rosa, J., Mikhailov, V., Zhang, L., Wetjen, F., Samuel, D., Liu, P., Braaten, R.-A., Mæhlum, P., Birkenes, M. B., Kutuzov, A., Enstad, T., Farsethås, H. C., Brygfjeld, S. A., Gulla, J. A., Oepen, S., Velldal, E., Østgulen, W., Øvrelid, L., and Myhre, A. S. (2025). The impact of copyrighted material on large language models: A Norwegian perspective. In Johansson, R. and Stymne, S., editors, Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025), pages 544–560, Tallinn, Estonia. University of Tartu Library.

References II

de Vries, W. and Nissim, M. (2021). As good as new. how to successfully recycle English GPT-2 to make models for other languages. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 836–846, Online. Association for Computational Linguistics.

Gao, T., Wettig, A., Yen, H., and Chen, D. (2025). How to train long-context language models (effectively). In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T., editors, Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7376–7399, Vienna, Austria. Association for Computational Linguistics.

Giellatekno and Divvun (2016). SIKOR North Saami corpus.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8342–8360, Online. Association for Computational Linguistics.

References III

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J. W., and Sifre, L. (2022). An empirical analysis of compute-optimal large language model training. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, Advances in Neural Information Processing Systems.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., and Ginsburg, B. (2024). RULER: What's the real context size of your long-context language models? In First Conference on Language Modeling.

Ibrahim, A., Thérien, B., Gupta, K., Richter, M. L., Anthony, Q. G., Belilovsky, E., Lesort, T., and Rish, I. (2024). Simple and scalable strategies to continually pre-train large language models. Transactions on Machine Learning Research.

References IV

Imani, A., Lin, P., Kargaran, A. H., Severini, S., Jalili Sabet, M., Kassner, N., Ma, C., Schmid, H., Martins, A., Yvon, F., and Schütze, H. (2023). Glot500: Scaling multilingual corpora and language models to 500 languages. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. (2023). Mistral 7b.

Lozhkov, A., Ben Allal, L., von Werra, L., and Wolf, T. (2024a). Fineweb-edu: the finest collection of educational content.

References V

Lozhkov, A., Li, R., Allal, L. B., Cassano, F., Lamy-Poirier, J., Tazi, N., Tang, A., Pykhtar, D., Liu, J., Wei, Y., Liu, T., Tian, M., Kocetkov, D., Zucker, A., Belkada, Y., Wang, Z., Liu, Q., Abulkhanov, D., Paul, I., Li, Z., Li, W.-D., Risdal, M., Li, J., Zhu, J., Zhuo, T. Y., Zheltonozhskii, E., Dade, N. O. O., Yu, W., Krauß, L., Jain, N., Su, Y., He, X., Dey, M., Abati, E., Chai, Y., Muennighoff, N., Tang, X., Oblokulov, M., Akiki, C., Marone, M., Mou, C., Mishra, M., Gu, A., Hui, B., Dao, T., Zebaze, A., Dehaene, O., Patry, N., Xu, C., McAuley, J., Hu, H., Scholak, T., Paquet, S., Robinson, J., Anderson, C. J., Chapados, N., Patwary, M., Tajbakhsh, N., Jernite, Y., Ferrandis, C. M., Zhang, L., Hughes, S., Wolf, T., Guha, A., von Werra, L., and de Vries, H. (2024b). StarCoder 2 and The Stack v2: The next generation.

Muennighoff, N., Rush, A. M., Barak, B., Scao, T. L., Tazi, N., Piktus, A., Pyysalo, S., Wolf, T., and Raffel, C. (2023). Scaling data-constrained language models. In Thirty-seventh Conference on Neural Information Processing Systems.

References VI

Nguyen, T., Nguyen, C. V., Lai, V. D., Man, H., Ngo, N. T., Dernoncourt, F., Rossi, R. A., and Nguyen, T. H. (2023). CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages.

Samuel, D., Mikhailov, V., Velldal, E., Øvrelid, L., Charpentier, L. G. G., Kutuzov, A., and Oepen, S. (2025). Small languages, big models: A study of continual training on languages of Norway. In Johansson, R. and Stymne, S., editors, Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025), pages 573–608, Tallinn, Estonia. University of Tartu Library.

Yen, H., Gao, T., Hou, M., Ding, K., Fleischer, D., Izsak, P., Wasserblat, M., and Chen, D. (2025). Helmet: How to evaluate long-context models effectively and thoroughly. In Yue, Y., Garg, A., Peng, N., Sha, F., and Yu, R., editors, International Conference on Learning Representations, volume 2025, pages 98914–98965.