ModernGBERT: German-only 1B Encoder Model Trained from Scratch

Anonymous ACL submission

Abstract

Despite the prominence of decoder-only language models, encoders remain crucial for resource-constrained applications. We introduce ModernGBERT (134M, 1B), a fully transparent family of German encoder models trained from scratch, incorporating architectural innovations from ModernBERT. To evaluate the practical trade-offs of training encoders from scratch, we also present LLäMmlein2Vec (120M, 1B, 7B), a family of encoders derived from German decoder-only models via LLM2Vec. We benchmark all models on natural language understanding, text embedding, and long-context reasoning tasks, enabling a controlled comparison between dedicated encoders and converted decoders. Our results show that ModernGBERT 1B outperforms prior state-of-the-art German encoders as well as encoders adapted via LLM2Vec, with regard to performance and parameter-efficiency. All models, training data, checkpoints and code are publicly available[1], advancing the German NLP ecosystem with transparent, high-performance encoder models.

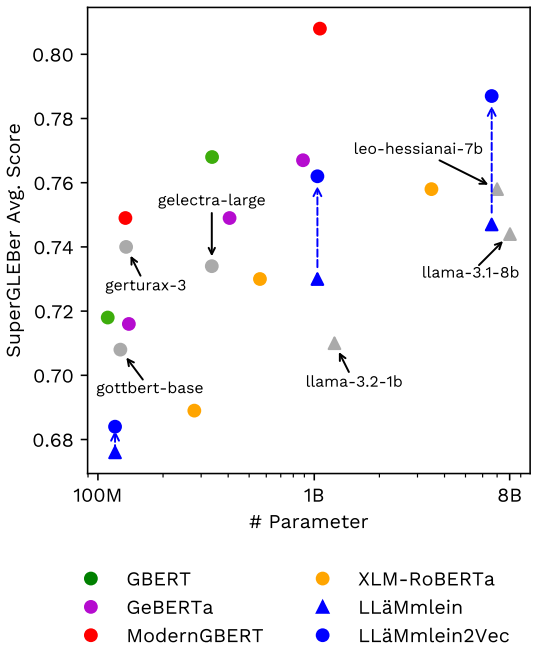

Figure 1: Performance on SuperGLEBer benchmark. • markers: encoders, ▲ markers: decoders. Dashed arrows: LLM2Vec conversion gains. Models of the same family are colored in the same color.

1 Introduction

Despite the recent dominance of decoder-only large language models (LLMs), parameter-efficient encoder models remain crucial for language technology, particularly for local deployments such as retrieval-augmented generation (RAG). Their bidirectional attention confers strong understanding capabilities with lower resource requirements, making them attractive for consumer hardware. In the German NLP landscape, $\text{GBERT}_{\text{Large}}$ (337M parameters; Chan et al., 2020) remains a popular encoder, performing competitively with much larger German-capable decoder LLMs across tasks (Pfister and Hotho, 2024), despite its modest size and limited training data (163 GB). More recently, ModernBERT (Warner et al., 2024) introduced several architectural improvements for English encoders, including enhanced relative positional embeddings and efficient attention patterns enabling long context processing. Building on this progress and inspired by the success of LLäMmlein (Pfister et al., 2024), a family of German decoder-only LLMs transparently trained on approximately 6 TB of RedPajamaV2 (Weber et al., 2024) text, we introduce ModernGBERT a family of fully open, high-performance German encoder models with 134M and 1B parameters. These models provide a foundation to explore the impact of ModernBERT's architectural innovations on German encoder performance. They also allow us to investigate how

[1] ModernGBERT and LLäMmlein2Vec, including all code, data and intermediary checkpoints will be published upon acceptance under a "research-only RAIL" license.

parameter scaling influences model quality when trained on large-scale monolingual corpora.

To better assess the practical utility and trade-offs of training encoder models from scratch, we further present LLäMmlein2Vec encoders (120M, 1B, and 7B), derived from decoder-only models using LLM2Vec (BehnamGhader et al., 2024). Since all models are based on the same training datasets, this setup provides a foundation for systematically analyzing the relationship between different architectures and training strategies.

We extensively evaluate these models during and post training via: natural language understanding (SuperGLEBer, Pfister and Hotho, 2024), embedding performance (MTEB; Enevoldsen et al., 2025; Muennighoff et al., 2023; Wehrli et al., 2023), and long-context understanding (Question Answering Needle-in-a-Haystack). Our findings reveal:

- ModernGBERT 134M and 1B are highly competitive German encoders, scaling well with size (8,192 tokens), with 1B surpassing the previous SotA $\text{GBERT}_{\text{Large}}$.
- Our LLäMmlein2Vec 7B also outperforms $\text{GBERT}_{\text{Large}}$, though dedicated encoders still outperform converted models of similar size.

Note: Throughout the paper, we highlight interesting findings and insights we gained during the process in little boxes like this one.

2 Datasets

2.1 Pre-training Dataset

We pre-trained ModernGBERT on the same data as LLäMmlein decoder models (Pfister et al., 2024), using the open-source RedPajamaV2 dataset (Weber et al., 2024).[2] This dataset comprises German CommonCrawl snapshots from 2014–2023. As we intend to keep datasets constant between ModernGBERT and LLäMmlein, we follow LLäMmlein's data pipeline and select the higher quality document-level deduplicated "head" and "middle" partitions, excluding the lower quality "tail" partition. For our 134M model, we only selected the head partition. We used the same processing pipeline as Pfister et al. (2024): First, paragraph-level deduplication using a Bloom filter to remove redundant content like GDPR notices and web boilerplate, improving data diversity. Then, a token-to-word ratio filter to further improve text quality. The final dataset is approximately 6 TB, consisting of $\approx$ 2 TB from head and $\approx$ 4 TB from middle. Using a $\text{GBERT}_{\text{Large}}$ tokenizer, this results in about 1.27T tokens.

Dataset	# tokens	# sequences	median length
LONG-Head (ext1)	52B	6,813,019	7,755
LONG-Head/Middle (ext1)	90B	11,785,941	8,013
HQ (ext2)	14.4B	43,191,271	199
Fineweb2	7,640M	42,319,173	194
OpenLegalData	407M	53,798	7,583
Wikipedia	143M	19,004	7,515
Fineweb2-long	6,211M	799,296	7,902

Table 1: Composition of the context extension datasets.

2.2 Context Extension Dataset

ModernBERT enhances its context capacity from 1,024 to 8,192 by finetuning in two phases: on $\approx$ 250B-token subsample of 8,192-token sequences from its original pre-training dataset (ext1), followed by a curated $\approx$ 50B-token dataset with mixed sequence lengths, including short and long ones (ext2, up to 8,192 tokens) (Gao et al., 2025).

Following this setup, we proceed to construct our own German context extension datasets for the two phases (Table 1): for the first phase (ext1), we take the same approach and subsample long sequences from our pre-training datasets (resulting in "LONG-Head" from the head partition for our 134M model, and "LONG-Head/Middle" from the head and middle partition for our 1B model). For the second phase (ext2) on a high-quality dataset we coin "HQ" in Table 1, we use the German portion of the Fineweb2 dataset (Penedo et al., 2024)[3]. Aiming for a similar distribution, we first take a randomized sample of Fineweb2, and added a separate sample from Fineweb2, selecting long documents with $\geq$8,192 tokens, splitting them into sequences of $\approx$ 8,192 tokens ("Fineweb2-long"). Furthermore, we add additional long documents by including the 2023 German Wikipedia[4] and the 2022 OpenLegalData dump,[5] also split to sequences of up to 8,192 tokens. The entire HQ dataset consists of 14.4B tokens. Table 1 summarizes these three resulting datasets.

[2] Common Crawl Foundation Terms of Use

[3] Open Data Commons Attribution License (ODC-By) v1.0

[4] Creative Commons Attribution-ShareAlike 3.0

[5] https://de.openlegaldata.io/; database licensed under Open Database License (ODbL v1.0), cases are exempt from copyright law

3 Methodology

3.1 ModernGBERT: Scaling SotA Encoders

The ModernGBERT models adapt the ModernBERT architecture and training strategy for German. ModernGBERT 134M matches the base ModernBERT model size (22 layers, 768 hidden units, but 16M fewer parameters due to a smaller vocabulary size), while ModernGBERT 1B consists of 28 layers and a hidden size of 2,048. Full architectural details can be found in Table 4.

Both models follow the ModernBERT pretraining recipe: masked language modeling (MLM) with no next-sentence-prediction, a 30% masking rate, and sequences up to 1,024 tokens (10,000 RoPE theta). For training, we use our German pre-training corpus (Section 2.1): ModernGBERT 1B trains on the head then middle partitions for a total of 1.27T tokens; ModernGBERT 134M is trained only on the head partition (0.47T tokens), as downstream evaluation showed early saturation (Section 5.1). Details of the training procedure are shown in Table 5.

After MLM, we extend context length in two phases, following ModernBERT, raising the RoPE theta to 160,000 and training on longer sequences. In the first extension phase (ext1), models are trained on the LONG-Head for the 134M model or LONG-Head/Middle for the 1B model. In ext2, both models are trained on the HQ dataset. As we did not intend to develop a novel German tokenizer, we utilized the original BERT-style tokenizer from $\text{GBERT}_{\text{Large}}$ (resulting in a 31,168-word embedding layer - a multiple of 64 for compute efficiency). While LLäMmlein (Pfister et al., 2024) provides a dedicated German Llama-style tokenizer, our preliminary ablations consistently showed degraded downstream performance. This degradation is consistent with results by Warner et al. (2024), who observed similar behavior with (English) Llama-style tokenizers during the development of the original ModernBERT. We therefore retained the $\text{GBERT}_{\text{Large}}$ tokenizer.

Throughout the training checkpoints are saved and evaluated, and all are released publicly to support further research. In addition, inspired by Pythia (Biderman et al., 2023) we provide full training provenance by logging and releasing the order of data points seen during training; thus, all checkpoints can be linked with the exact data points seen up to that checkpoint.

3.2 LLM2Vec: Turning Decoders to Encoders

LLM2Vec (BehnamGhader et al., 2024) proposes a method to convert decoder-only LLMs into effective text encoders through the following steps: First, the causal attention mask is replaced with a full attention mask, enabling bidirectional attention across tokens. Second, the model is trained using a masked next token prediction (MNTP) objective. Third, unsupervised contrastive learning (SimCSE) is applied, improving embedding quality by maximizing agreement between differently dropped-out versions of the same input. However, we intend to remain closely aligned with ModernGBERT's training objectives and not reproduce LLM2Vec results. For this reason, we trained our models exclusively using the MNTP objective, which is most similar to MLM. ModernGBERT performs two context extension phases, each using two datasets per model. Similarly, we train all three LLäMmlein models using the same respective two datasets as employed by ModernGBERT's context extensions (Section 2.2): the LLäMmlein2Vec 120M model variant follows ModernGBERT 134M (LONG-Head for phase one, HQ for phase two); the LLäMmlein2Vec 1B and 7B models follow ModernGBERT 1B (LONG-Head/Middle for phase one, HQ tokens for phase two). For each model, we apply MNTP training separately on each respective dataset, resulting in two distinct adapter modules—one per phase. We evaluate both individual adapters (ext1 & ext2) as well as a merged model (ext1+2) where both adapters are combined. Notably, the models achieve comparable results even without seeing the full training data as exemplary shown in Table 7 - this is comparable to the observations in Pfister et al. (2024), indicating possibilities of reducing compute in future trainings. However, for consistency and comparability, we report results using the fully trained models throughout the paper[6]. We also increased the sequence length to 8,192 and set RoPE theta to 160,000, otherwise, we follow the default LLM2Vec parameters (see Table 6 for more details).

[6] except for the LLäMmlein2Vec 7B trained on the LONG-Head/Middle model, which we trained on 64 nodes with 4 H200 each for 14 hours, before stopping the training due to compute constraints (Table 6)

4 Evaluation Setup

4.1 SuperGLEBer

We assess our final models using the German SuperGLEBer benchmark (Pfister and Hotho, 2024), which includes 29 tasks across text classification, sequence tagging, question answering, and sentence similarity. These tasks cover diverse domains such as news, legal texts, and consumer reviews. For each task, models are fine-tuned with QLoRA (Dettmers et al., 2023) by default, or LoRA as fallback. In addition to evaluating final checkpoints, we follow LLäMmlein (Pfister et al., 2024) and evaluate intermediate checkpoints on the same representative SuperGLEBer subset as selected by Pfister et al.: the classification tasks NLI (Conneau et al., 2018), FactClaiming Comments (Risch et al., 2021), DB Aspect (Wojatzki et al., 2017), and WebCAGe (Henrich et al., 2012), the sequence tagging task EuroParl (Faruqui and Padó, 2010), and the sentence similarity task PAWSX (Liang et al., 2020).

4.2 Massive Text Embedding Benchmark

We further evaluate the models on the German subset of the Massive Text Embedding Benchmark *MTEB(deu,v1)* (Enevoldsen et al., 2025). The specific tasks can be found in Table 8. In addition to text pair classification and semantic textual similarity—already covered by the SuperGLEBer benchmark—MTEB includes clustering (Wehrli et al., 2023), as well as reranking and retrieval tasks.

These latter tasks provide a more comprehensive assessment of general-purpose sentence embeddings, focusing on the models' ability to produce robust semantic representations.

To adapt the base models for embedding tasks, we fine-tune them using the Sentence-Transformer framework (Reimers and Gurevych, 2019) in a supervised setup. Fine-tuning employs 10,000 samples from the German portion of the machine-translated multilingual mMARCO passage ranking dataset (Bonifacio et al., 2022), maximizing similarity between query and positive passages, while minimizing similarity to negative passages. Sentence embeddings are obtained by mean pooling over the final token representations. We use InfoNCE loss with a batch size of 128 and a learning rate of 5×10^{-5}. We apply QLoRA for efficient training (falling back to LoRA for the GBERT family, where quantization is not supported).

4.3 Long-Context Understanding

Evaluating long-context capabilities in German is hindered by the scarcity of native high-quality datasets, with translations from English often introducing artifacts.

To address this, we construct a Question-Answering Needle-In-a-Haystack (QA-NIAH) evaluation (Ivgi et al., 2023; Hsieh et al., 2024) based on the human-annotated GermanQuAD dataset (Möller et al., 2021). Given a question, the goal is to extract the answer span from a long document. We adapt GermanQuAD to a QA-NIAH setup as follows: for each question–paragraph ("needle") pair, we sample up to 3 distractor paragraphs and shuffle them with the needle, forming a "haystack" document of up to 1,024 tokens. The answer always appears only in the needle paragraph. For evaluation, we increase distractors to up to 20, yielding documents up to 8,192 tokens. This yields 11,518 training and 2,204 test question–haystack pairs. We fine-tune models on the QA-NIAH training set using QLoRA, and evaluate on the test set, following the SuperGLEBer benchmark procedure. Training uses sequences up to 1,024 tokens; evaluation uses up to 8,192 tokens, assessing generalization to longer contexts.

5 Evaluation Results

5.1 Intermediate Model Evaluation

To track pre-training progress, we evaluated intermediate checkpoints on six representative SuperGLEBer tasks, following Pfister et al. (2024) (see Section 4.1). All checkpoints will be released for future analysis. Figure 3 shows that ModernGBERT 1B's average performance steadily improves throughout training, while ModernGBERT 134M quickly saturates.

To quantify these trends, we evaluated the full SuperGLEBer suite on four checkpoints for ModernGBERT 1B and three for 134M (comparison done using Wilcoxon signed-rank tests): ModernGBERT 134M plateaued after 72B tokens (15% of data), with no further significant improvements. In contrast, ModernGBERT 1B showed significant gains over the same dataset portion ($p < 0.0001$), and additional gains during training on the middle partition ($p < 0.00052$). Performance then plateaus after 864B tokens (67% of entire pre-training dataset), with the SuperGLEBer score increasing only slightly from 0.777 to 0.791 despite 406B more tokens.

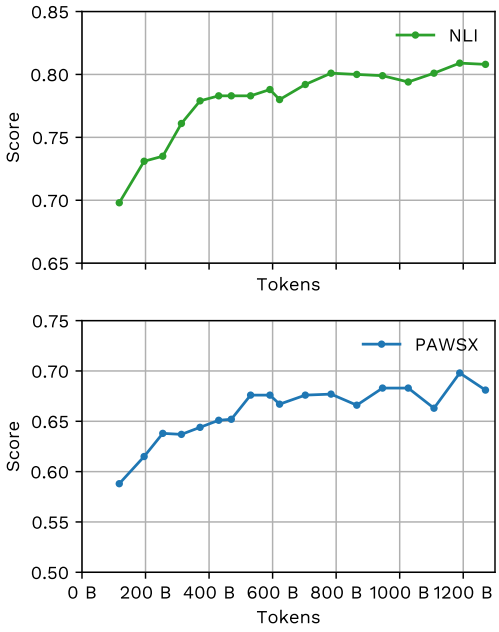

Figure 2: Intermediate checkpoint evaluation of ModernGBERT 1B during pre-training, on the SuperGLEBer tasks: NLI (top) and PAWSX (bottom). To improve trend visibility, we adjusted the y-axis scale and plotted every second checkpoint.

Analyzing on the six selected subtasks that were run on every checkpoint (Section 5.1), for the 134M variant, only PAWSX showed a significant positive Spearman rank correlation between training token count and performance ($r = 0.655; p < 0.003$), while the others did not. For 1B, all but EuroParl showed significant positive correlations ($r > 0.57; p < 0.00014$). In particular, even though the aggregate score remains largely stable in the final third of the pre-training, on complex tasks such as NLI and PAWSX, we still see slight improvements with increased training (Figure 2).

These saturation patterns, including per-task trends and overall performance plateaus, are consistent with findings by Pfister et al. (2024) for decoder models, and by Antoun et al. (2024) for their ModernBERT variant ModernCamemBERT (with 136M parameters) trained on French. Our results confirm that although small ModernBERT models saturate quickly, larger models benefit from additional data. Extrapolating the observed scaling behavior between ModernGBERT 134M and 1B, we hypothesize that training a larger, 7B-sized encoder could make further use of an extensive monolingual datasets, improving performance beyond ModernGBERT 1B.

Model	Avg	Class.	NER	PAWSX	QA
$\text{GBERT}_{\text{Base}}$	0.718	0.723	0.786	0.561	0.803
$\text{GBERT}_{\text{Large}}$	0.768	0.785	0.799	0.654	0.832
$\text{GeBERTa}_{\text{Base}}$	0.716	0.715	0.778	0.559	0.813
$\text{GeBERTa}_{\text{Large}}$	0.749	0.743	0.791	0.619	0.844
$\text{GeBERTa}_{\text{XLarge}}$	0.767	0.770	0.807	0.643	0.848
$\text{XLM-RoBERTa}_{\text{Base}}$	0.689	0.693	0.754	0.505	0.802
$\text{XLM-RoBERTa}_{\text{Large}}$	0.730	0.714	0.787	0.583	0.837
$\text{XLM-RoBERTa}_{\text{XLarge}}$	0.758	0.750	0.802	0.656	0.822
LLäMmlein 120M	0.676	0.702	0.712	0.477	0.812
LLäMmlein2Vec 120M	0.684	0.703	0.741	0.472	0.819
LLäMmlein 1B	0.733	0.781	0.773	0.548	0.828
LLäMmlein2Vec 1B	0.762	0.776	0.812	0.615	0.843
LLäMmlein 7B	0.747	0.810	0.805	0.524	0.851
LLäMmlein2Vec 7B	0.787	0.799	0.838	**0.670**	0.842
ModernGBERT 134M$^{\diamond}$	0.730	0.716	0.782	0.589	0.833
ModernGBERT 134M	0.749	0.735	0.805	0.612	0.836
ModernGBERT 1B$^{\diamond}$	0.800	0.806	0.839	0.681	0.874
ModernGBERT 1B	**0.808**	**0.812**	**0.845**	0.699	**0.876**

Table 2: Performance comparison on SuperGLEBer. $\diamond$ indicates ModernGBERT without context extension.

Confirmation: Our findings corroborate Antoun et al. (2024) and Pfister et al. (2024): small ModernGBERT models reach saturation early, while scaling model and dataset size enables improvements.

5.2 Final Model Evaluation

Natural Language Understanding We evaluate all final models on the full SuperGLEBer benchmark. Table 2 averages the scores per task type, while Table 7 provides more fine-grained scores. We compare our models to established encoders: GBERT (Chan et al., 2020), GeBERTa (Dada et al., 2023), and XLM-RoBERTa (Conneau et al., 2020; Goyal et al., 2021).

Our ModernGBERT consistently outperforms comparable and larger models. The 134M variant achieves an average score of 0.749, surpassing all similar-sized baselines, including $\text{GBERT}_{\text{Base}}$ (0.718), $\text{XLM-RoBERTa}_{\text{Base}}$ (0.689), $\text{GeBERTa}_{\text{Base}}$ (0.716), and even $\text{XLM-RoBERTa}_{\text{Large}}$ (0.730), as well as LLäMmlein 1B (0.733). The ModernGBERT 1B variant achieves a new state-of-the-art average score across the entire SuperGLEBer of 0.808, outperforming $\text{GBERT}_{\text{Large}}$ (0.768) by 4% and beating the seven times larger LLäMmlein2Vec 7B (0.787). It leads in three of four evaluation categories, including classification (0.812), NER (0.845), and QA (0.876). Only on sentence similarity (0.699), our seven times larger LLäMmlein2Vec 7B achieves better results. ModernGBERT scales well, with performance improving for larger model sizes, again suggesting that scaling ModernBERT-style

encoders can leverage large monolingual corpora effectively. In the SuperGLEBer setting, adding context extension improved ModernGBERT's average by 1.9% for the 134M model (from 0.730 to 0.749) and by 0.8% for the 1B variant (from 0.800 to 0.808). No large improvements were to be expected from our context extension, as SuperGLEBer tasks do not make use of long contexts.

Adaptation via LLM2Vec yields consistent gains across models. Thus, our first LLM2Vec tuning (analogous to ext1, Section 2.2) showed the most prominent positive effect, while the second finetune using the ext2 datasets showed only marginal increase, or even a decrease in performance. The same holds for a mixture of the two LLM2Vec adapters (ext1+2). The LLäMmlein2Vec 7B achieves the strongest results among the LLM2Vec models (0.787). Conversion of LLäMmlein 120M, 1B, 7B improved the average score by 0.8%, 2.9%, and 4.0% respectively. This effect is especially pronounced in PAWSX, with scores increasing by up to 14.6% for LLäMmlein 7B and 6.7% for LLäMmlein 1B.

Observation: LLM2Vec yields the best improvement on similarity-related tasks.

Comparing the LLäMmlein2Vec with the ModernGBERT family, we find that on similarly sized models, ModernGBERT always outperforms the transformed decoders by a large margin. Only the much larger LLäMmlein2Vec 7B approaches the performance of ModernGBERT 1B.

Observation: With similar data and model sizes, training encoders from scratch outperforms LLM2Vec converted models.

Text Embedding We evaluate models on the MTEB benchmark, which covers six task categories: classification, pair classification, clustering, reranking, retrieval, and short text similarity (STS) tasks. While Table 3 summarizes the outcomes, all results are presented in Table 9. In general, supervised fine-tuning on mMARCO yields consistent improvements across all model types. While classification performance sometimes declines, substantial gains can be observed in other areas: 25% on average for reranking, 26% for retrieval and 25% for STS.

Model	Avg	Clustering	Reranking	Retrieval
$\text{GBERT}_{\text{Base}}$	0.360	0.274	0.118	0.226
$\text{GBERT}_{\text{Base}}{}^{\dagger}$	0.500	0.318	0.374	0.461
$\text{GBERT}_{\text{Large}}$	0.412	0.336	0.206	0.297
$\text{GBERT}_{\text{Large}}{}^{\dagger}$	0.521	0.334	0.389	0.493
$\text{XLM-RoBERTa}_{\text{Base}}$	0.248	0.173	0.024	0.008
$\text{XLM-RoBERTa}_{\text{Base}}{}^{\dagger}$	0.403	0.247	0.247	0.299
$\text{XLM-RoBERTa}_{\text{Large}}$	0.264	0.172	0.048	0.026
$\text{XLM-RoBERTa}_{\text{Large}}{}^{\dagger}$	0.460	0.259	0.343	0.416
$\text{XLM-RoBERTa}_{\text{XLarge}}$	0.301	0.225	0.090	0.142
$\text{XLM-RoBERTa}_{\text{XLarge}}{}^{\dagger}$	0.479	0.342	0.362	0.407
LLäMmlein2Vec 120M	0.315	0.261	0.139	0.224
LLäMmlein2Vec 120M†	0.471	0.308	0.325	0.425
LLäMmlein2Vec 1B	0.399	0.308	0.183	0.276
LLäMmlein2Vec 1B†	0.540	**0.343**	0.433	0.511
LLäMmlein2Vec 7B	0.376	0.249	0.169	0.266
LLäMmlein2Vec 7B†	**0.557**	0.339	**0.477**	**0.522**
ModernGBERT 134M$^{\diamond}$	0.383	0.293	0.139	0.241
ModernGBERT 134M$^{\diamond\dagger}$	0.485	0.303	0.364	0.432
ModernGBERT 134M	0.376	0.296	0.120	0.213
ModernGBERT 134M†	0.501	0.312	0.404	0.446
ModernGBERT 1B$^{\diamond}$	0.374	0.318	0.097	0.199
ModernGBERT 1B$^{\diamond\dagger}$	0.549	0.339	0.463	0.511
ModernGBERT 1B	0.366	0.307	0.088	0.191
ModernGBERT 1B†	0.551	0.338	0.459	0.512

Table 3: Performance comparison on MTEB. "Avg" refers to the average over all six task groups, not only the ones shown here. ◇ indicates ModernGBERT without context extension, while † marks the variant with additional training.

Observation: Fine-tuning yields the largest gains in reranking, retrieval, and STS tasks.

The best overall average performance is achieved by the fine-tuned LLäMmlein2Vec 7B (0.557), closely followed by the fine-tuned ModernGBERT 1B (0.551), despite the latter being significantly smaller. LLäMmlein2Vec models generally show strong performance after fine-tuning, particularly when trained with the extension dataset of the first phase (ext1). Using the second extension phase (ext2) or combining both adapters into the base model (ext1+2) harms the performance. Interestingly, the latter shows the largest fine-tuning gains among the three variants.

The ModernGBERT models perform competitively to similarly sized models. Before fine-tuning, ModernGBERT 1B (avg. 0.366) already outperforms most encoder-only models, such as $\text{GeBERTa}_{\text{XLarge}}$ (0.325) or $\text{XLM-RoBERTa}_{\text{XLarge}}$ (0.301), but not $\text{GBERT}_{\text{Large}}$ (0.412). However, after fine-tuning, it demonstrates clear superiority among native encoder-only models by at least 3% on the average score. As with our observations on the SuperGLEBer benchmark, ModernGBERT's context extension did not show significant improvements here.

Comparing ModernGBERT and LLäMmlein2Vec, we find that before fine-tuning, the LLäMmlein2Vec 1B and 7B models produce better representations than ModernGBERT 1B. However, after fine-tuning, ModernGBERT 1B surpasses the 1B variant of LLäMmlein2Vec on average and closely aligns with the larger 7B model.

Long-Context Understanding Table 10 reports results on our German Question-Answering Needle-in-a-Haystack benchmark. Next to the overall accuracy, we also present accuracy on subsets of the test dataset, consisting only of short (<1,024), medium (1,024 to 4,095), resp. long (4,096 to 8,192) sequences. The evaluation focuses on LLMs supporting up to 8,192 tokens: ModernGBERT, the encoder-converted LLäMmlein2Vec, as well as their original decoder counterparts. Notably, LLäMmlein models were pre-trained with a maximum context of 2,048 tokens. ModernGBERT 1B demonstrates strong long-context performance across all lengths, outperforming all encoders. The first extension phase during ModernGBERT training yielded strong improvements, increasing accuracy by approximately factor 3, but the final extension phase on the HQ dataset slightly decreased performance by few percentage points, particularly for the 134M variant.

Regarding LLM2Vec, a sufficiently long conversion improved long-context understanding. Conversion of LLäMmlein 120M and 1B decoders (with native context length of 2,048) improved accuracy by factor 1.3 resp. 2, both not as pronounced in comparison to the ModernGBERT encoders. For LLäMmlein2Vec 7B however (with LLM2Vec training on approximately half of our ext1 dataset), it decreased by 51%, with no correct answers on haystacks of >4,096 tokens. Given the intensive compute requirements, we did not explore further optimizations regarding context extension of the LLäMmlein2Vec 7B model.

Observation: On small training datasets, LLM2Vec tuning limits the understanding of long-context samples.

5.3 Inference Efficiency

We evaluate inference efficiency across varying sequence lengths using four synthetic datasets, each containing 8,192 documents composed of random tokens. Following Warner et al. (2024), we created two datasets using fixed length sequences of either 512 or 8,192 tokens. The other two datasets feature normal distributed sequence lengths (either mean 256, variance 64; or mean 4,096, variance 1,024) to better simulate real-world conditions. Our ModernGBERT models adopt ModernBERT's unpadding approach: padding tokens are removed and sequences in a batch are concatenated, allowing Flash Attention to handle variable-length attention masks. The computational equivalence is facilitated by carefully crafting an appropriate attention mask. In contrast, all other models rely on conventional padding.

Table 11 summarizes our findings. Among smaller models (134M resp. 120M), ModernGBERT and LLäMmlein2Vec achieve comparable efficiency on fixed-length data, both only surpassed by $\text{GBERT}_{\text{Base}}$ and $\text{XLM-RoBERTa}_{\text{Base}}$ in terms of efficiency on short sequences.

For the 1B variants, ModernGBERT consistently outperforms LLäMmlein2Vec 1B and 7B variations in inference speed, likely due to its architectural decisions optimized for efficiency, such as ensuring that weight matrices have dimension of multiples of 64, and are divisible into 128×256 block for efficient tiling on the GPU. Gains are most pronounced for variable-length datasets, where ModernGBERT's unpadding yields clear benefits: the 134M ModernGBERT is the most efficient model on variable length, and the 1B variant substantially outpaces its LLäMmlein2Vec counterpart. Furthermore, given the comparable task performance for e.g. ModernGBERT 1B and LLäMmlein2Vec 7B on MTEB (see Table 9), the ModernGBERT model is 10 times as fast on variable length long context documents. The same trend is even more pronounced for ModernGBERT 1B, compared to its 1B LLäMmlein2Vec counterpart, where LLäMmlein2Vec is consistently outperformed by it similarly sized ModernGBERT version, which on top is twice as efficient on these long documents.

Observation: When considering the trade-off between computational efficiency and downstream performance metrics, ModernGBERT consistently emerges as the optimal solution—frequently outperforming LLäMmlein2Vec on both dimensions simultaneously.

6 Related Work

Next-Generation Encoders Several recent efforts have extended ModernBERT to new lan-

guages and domains, including adaptations for French (Antoun et al., 2024) and Japanese (Sugiura et al., 2025).

Concurrent to our work, several alternative encoder architectures have been proposed. Breton et al. (2025) introduced NeoBERT, an English encoder scaled to 250M, incorporating similar architectural innovations like ModernBERT, but scaling up layers rather than hidden dimension, switching from GeLU to SwiGLU activation, and using a modified training scheme (Cosine scheduler, reduced masking). Their model surpasses ModernBERT-large on GLUE and MTEB with 100M fewer parameters, although its scalability with model size remains unexplored.

Likewise, Boizard et al. (2025) recently presented EuroBERT (210M, 610M, 2.1B), a multilingual encoder family featuring architectural changes similar to those of ModernBERT, but retaining some architectural details (RMSNorm layer normalization, SiLU activation function, Llama-style tokenizer) from the Llama family, resembling our LLäMmlein2Vec architecture.

Antoun et al. (2025) compared French ModernBERT and DeBERTaV3 under controlled conditions, finding DeBERTaV3 to be superior on downstream tasks but significantly slower in training and inference.

Tuning decoder-only LLMs into Encoders Few works have investigated converting decoder-only LLMs into encoders, besides LLM2Vec (Section 3.2). Recent studies predominantly address either distilling text embedders (Li and Li, 2024; Lee et al., 2025, 2024; Ma et al., 2025) or fine-tuning LLMs as bidirectional encoders for specific tasks (Li et al., 2023; Dukić and Snajder, 2024), with evaluation typically focused on English or multilingual settings.

Concurrently, MAGNET (Khosla et al., 2025) was proposed for converting decoder LLMs into foundational encoders, similarly to LLM2Vec. Unlike LLM2Vec, MAGNET employs both bidirectional and causal attention and adds a missing-span generation objective.

7 Conclusion

We have demonstrated that both architectural advances in ModernBERT and the LLM2Vec decoder transformation method yield strong German encoder models. The proposed ModernGBERT family, especially the 1B variant, sets a new state-of-the-art for German encoders, outperforming previous models while remaining suitable for practical deployment as a drop-in replacement for GBERT, capable of handing sequences of up to 8,192 tokens. Our learning dynamics analysis confirms that larger encoder architectures can effectively exploit terabyte-scale German monolingual corpora, with performance consistently improving with increased model size and data. These trends suggest that even larger encoder models could yield further gains, which we leave to future work.

A comparison of ModernGBERT, and LLäMmlein2Vec (derived from LLäMmlein) both based on the same dataset, shows that dedicated encoder training yields superior results, justifying its computational expense when parameter efficiency is essential. By releasing ModernGBERT, along with full training transparency, intermediate checkpoints, and detailed documentation, we aim to facilitate further development and understanding within the German LLM community.

Limitations

Despite the ModernGBERT models being a notable advancement in the German NLP landscape, several limitations persist: **1) Monolingual focus.** Although the focus on German is a strength for this specific context, ModernGBERT is unable to utilize multilingual contexts or perform cross-lingual tasks, hindering applicability in some scenarios. **2) Limited coding capabilities.** High-quality German resources for coding are rare, and no code is included in the training dataset. This restricts its capabilities in code retrieval applications. **3) Evaluation scope.** While we rigorously evaluated our models on the German SuperGLEBer and MTEB benchmarks, these benchmarks are limited in their domain, and other domains such as literature, medical domains, or technical subjects were not tested. Furthermore, our benchmarks do not strictly probe for "German factual knowledge", for instance, knowledge about German geography, or common German TV shows. **4) No custom tokenizer.** We utilized the original BERT-style GBERT tokenizer due to its availability and persistent usage. However, we did not invest in developing a custom tokenizer, like the BPE-style OLMo tokenizer used in ModernBERT. Consequently, ModernGBERT's tokenizer cannot, e.g., differentiate between various whitespace characters or encode emoji.

5) Evaluation of long-context understanding. Due to the absence of high-quality native German evaluation datasets, we had to rely on non-natural QA-NIAH sequences, only broadly testing for long–context understanding. Contrast this with English benchmarks such as ∞Bench-MC (Zhang et al., 2024) or LongBench-v2 (Bai et al., 2025), which include full novels along with questions that require attention to many information scattered throughout the novel. In future work, we plan on developing a dedicated high-quality non-synthetic German long-context evaluation benchmark.

References

Wissam Antoun, Francis Kulumba, Rian Touchent, Éric de la Clergerie, Benoît Sagot, and Djamé Seddah. 2024. Camembert 2.0: A smarter french language model aged to perfection.

Wissam Antoun, Benoît Sagot, and Djamé Seddah. 2025. ModernBERT or DeBERTaV3? examining architecture and data influence on transformer encoder models performance. ArXiv:2504.08716.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. ArXiv:2412.15204.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*, Philadelphia, PA, USA.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte M. Alves, André Martins, Ayoub Hammal, Caio Corro, Céline Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El-Haddad, Manuel Faysse, Maxime Peyrard, Nuno M. Guerreiro, Patrick Fernandes, Ricardo Rei, and Pierre Colombo. 2025. EuroBERT: Scaling multilingual encoders for european languages. ArXiv:2503.05500.

Hamed Bonab, Mohammad Aliannejadi, Ali Vardasbi, Evangelos Kanoulas, and James Allan. 2021. Cross-market product recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, page 110–119, New York, NY, USA. Association for Computing Machinery.

Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. mMARCO: A multilingual version of the MS MARCO passage ranking dataset. ArXiv:2108.13897.

Lola Le Breton, Quentin Fournier, Mariam El Mezouar, and Sarath Chandar. 2025. NeoBERT: A next-generation BERT. ArXiv:2502.19587.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott A. Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1094–1106, Seattle, United States. Association for Computational Linguistics.

Aaron Chibb. 2022. German-english false friends in multilingual transformer models: An evaluation on robustness and word-to-word fine-tuning. `huggingface:aari1995/false_friends_en_de`.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Amin Dada, Aokun Chen, Cheng Peng, Kaleb Smith, Ahmad Idrissi-Yaghir, Constantin Seibold, Jianning Li, Lars Heiliger, Christoph Friedrich, Daniel Truhn, Jan Egger, Jiang Bian, Jens Kleesiek, and Yonghui Wu. 2023. On the impact of cross-domain data on German language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13801–13813, Singapore. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized llms. ArXiv:2305.14314.

David Dukić and Jan Snajder. 2024. Looking right is sometimes right: Investigating the capabilities of decoder-only LLMs for sequence labeling. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14168–14181, Bangkok, Thailand. Association for Computational Linguistics.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, Ömer Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Mariya Hendriksen, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Šuppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Michael Günther, Mengzhou Xia, Weijia Shi, Xing Han Lù, Jordan Clive, Gayatri Krishnakumar, Anna Maksimova, Silvan Wehrli, Maria Tikhonova, Henil Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide, Lester James Miranda, Alena Fenogenova, Guangyu Song, Ruqiya Bin Safi, Wen-Ding Li, Alessia Borghini, Federico Cassano, Hongjin Su, Jimmy Lin, Howard Yen, Lasse Hansen, Sara Hooker, Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. 2025. MMTEB: Massive multilingual text embedding benchmark. ArXiv:2502.13595.

Manaal Faruqui and Sebastian Padó. 2010. Training and evaluating a german named entity recognizer with semantic generalization. In *Conference on Natural Language Processing*.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2023. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2025. How to train long-context language models (effectively). ArXiv:2410.02660.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 29–33, Online. Association for Computational Linguistics.

Verena Henrich, Erhard Hinrichs, and Tatiana Vodolazova. 2012. WebCAGe – a web-harvested corpus annotated with GermaNet senses. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 387–396, Avignon, France. Association for Computational Linguistics.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? ArXiv:2404.06654.

Maor Ivgi, Uri Shaham, and Jonathan Berant. 2023. Efficient long-text understanding with short-text models. *Transactions of the Association for Computational Linguistics*, 11:284–299.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual amazon reviews corpus. ArXiv:2010.02573.

Savya Khosla, Aditi Tiwari, Kushal Kafle, Simon Jenni, Handong Zhao, John Collomosse, and Jing Shi. 2025. MAGNET: Augmenting generative decoders with representation learning and infilling capabilities. ArXiv:2501.08648.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. NV-embed: Improved techniques for training llms as generalist embedding models. ArXiv:2405.17428.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. 2024. Gecko: Versatile text embeddings distilled from large language models. ArXiv:2403.20327.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented

semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.

Xianming Li and Jing Li. 2024. BeLLM: Backward dependency enhanced large language model for sentence embeddings. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 792–804, Mexico City, Mexico. Association for Computational Linguistics.

Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu lee Wang, Qing Li, and Xiaoqin Zhong. 2023. Label supervised LLaMA finetuning. ArXiv:2310.01208.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Xueguang Ma, Xi Victoria Lin, Barlas Oguz, Jimmy Lin, Wen tau Yih, and Xilun Chen. 2025. DRAMA: Diverse augmentation from large language models to smaller dense retrievers. ArXiv:2502.18460.

Philip May. 2021. Machine translated multilingual sts benchmark dataset. `huggingface:PhilipMay/stsb_multi_mt`.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. GermanQuAD and GermanDPR: Improving non-English question answering and passage retrieval. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 42–50, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. ArXiv:2210.07316.

James O'Neill, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. 2021. I wish i would have loved this one, but i didn't – a multilingual dataset for counterfactual detection in product reviews. ArXiv:2104.06893.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Martin Jaggi, Leandro von Werra, and Thomas Wolf. 2024. Fineweb2: A sparkling update with 1000s of languages. Huggingface:HuggingFaceFW/fineweb-2.

Jan Pfister and Andreas Hotho. 2024. SuperGLEBer: German language understanding evaluation benchmark. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7904–7923, Mexico City, Mexico. Association for Computational Linguistics.

Jan Pfister, Julia Wunderle, and Andreas Hotho. 2024. LLäMmlein: Compact and competitive german-only language models from scratch. ArXiv:2411.11171.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Julian Risch, Anke Stoll, Lena Wilms, and Michael Wiegand. 2021. Overview of the GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments*, pages 1–12, Duesseldorf, Germany. Association for Computational Linguistics.

Issa Sugiura, Kouta Nakayama, and Yusuke Oda. 2025. llm-jp-modernbert: A ModernBERT model trained on a large-scale japanese corpus with long context length. ArXiv:2504.15544.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. ArXiv:2412.13663.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. RedPajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 116462–116492. Curran Associates, Inc.

Silvan Wehrli, Bert Arnrich, and Christopher Irrgang. 2023. German text embedding clustering benchmark. In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, pages 187–201, Ingolstadt, Germany. Association for Computational Lingustics.

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. Germeval

2017: Shared task on aspect-based sentiment in social media customer feedback. In *Proceedings of the GermEval 2017 – Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, pages 1–12, Berlin, Germany.

Marco Wrzalik and Dirk Krechel. 2021. GerDaLIR: A German dataset for legal information retrieval. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 123–128, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. MIRACL: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

Appendix

A Model Architecture and Training Details

Table 4 provides an overview of the model architectures for the ModernGBERT (134M, 1B) and LLäMmlein2Vec (120M, 1B, 7B) model families. Detailed training settings regarding the pretraining phase, context extension phase one and two, for ModernGBERT are listed in Table 5, and those for LLäMmlein2Vec, covering MNTP training, can be found in Table 6.

B Evaluation Results

In the following we present the full evaluation results on SuperGLEBer, MTEB, NIAH, and our efficiency benchmarks for German-capable encoder models, specifically our ModernGBERT (134M, 1B) and LLäMmlein2Vec (120M, 1B, 7B).

B.1 SuperGLEBer

In Figure 3 we illustrate the training progress, evaluating several intermediate checkpoints of ModernGBERT 134M and 1B. Notably, while the smaller model did not show significant improvements after approximately 15% of it's training data, the 1B model improves performance until 67%.

Table 7 presents the results on the full SuperGLEBer benchmark, comparing various German-capable encoder models. Notably, ModernGBERT 1B sets a new state of the art, surpassing the previously leading encoder model $\text{GBERT}_{\text{Large}}$ as well as the seven times larger LLäMmlein2Vec 7B. Transforming the LLäMmlein decoders into encoders (in particular, the +ext1 variant) improves the average score and yields notable gains on similarity and sequence tagging tasks.

B.2 Massive Text Embedding Benchmark (MTEB)

We summarize all tasks included in the *MTEB(deu,,v1)* benchmark in Table 8 and report the corresponding results in Table 9. In addition to the base model outcomes, we also present scores for models after supervised training on the mMARCO dataset. Notably, the fine-tuned versions consistently outperform their base counterparts, with particularly strong improvements in reranking, retrieval, and s2s tasks. LLäMmlein2Vec 7B achieves the best results closely followed by ModernGBERT 1B.

B.3 Needle-in-a-Haystack

The results on the Question-Answering Needle-in-a-Haystack test are presented in Table 10. ModernGBERT 1B performs strongly across sequence lengths, surpassed only by the eight-times larger LLaMA 3.2. For LLäMmlein 120M and 1B, MNTP training on the LONG-Head or LONG-Head/Middle datasets improves performance on longer contexts.

B.4 Efficiency

Finally, we depict the outcomes of our model efficiency tests in Table 11. The smaller ModernGBERT model is the most efficient on variable-length input, while the 1B variant substantially outperforms its LLäMmlein2Vec counterpart.

	ModernGBERT		LLäMmlein2Vec		
Parameters	134M	1B	120M	1B	7B
Vocabulary	31,168	31,168	32,064	32,064	32,064
Unused Tokens	66	66	54	54	54
Layers	22	28	12	22	32
Hidden Size	768	2,048	768	2,048	4,096
Transformer Block	Pre-Norm	Pre-Norm	Post-Norm	Post-Norm	Post-Norm
Activation Function	GeLU	GeLU	SiLU	SiLU	SiLU
Attention Heads	12	32	12	32	32
Head Size	64	64	64	64	128
Intermediate Size	1,152	3,072	2,048	5,632	11,008
Normalization	LayerNorm	LayerNorm	RMSNorm	RMSNorm	RMSNorm
Norm Epsilon	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
RoPE theta	160,000	160,000	160,000	160,000	160,000
Global Attention	Every three layers	Every three layers	Every layer	Every layer	Every layer
Local Attention Window	128	128	—	—	—
Local Attn RoPE theta	10,000	10,000	—	—	—

Table 4: Model design of the ModernGBERT and LLäMmlein2Vec model family.

	Pretraining Phase		Context Extension: Phase One		Context Extension: Phase Two	
	134M	1B	134M	1B	134M	1B
Training Tokens	0.47T	1.27T	52B	90B	14.4B	
Max Sequence Length	1,024		8,192		8,192	
RoPE Theta	10,000		160,000		160,000	
Batch Size	4,608	4,928	96	96	96	96
Warmup (tokens)	3×10^9		—	—	—	—
Microbatch Size	96	28	8	3	8	3
Learning Rate	8×10^{-4}	5×10^{-5}	3×10^{-4}	5×10^{-5}	3×10^{-4}	5×10^{-6}
Schedule	Trapezoidal		—	—	1-sqrt	
Warmup (tokens)	15×10^9		—	—	—	—
Decay (tokens)	—	—	—	—	12.8×10^9	
Weight Decay	1×10^{-5}		1×10^{-5}	1×10^{-6}	1×10^{-5}	1×10^{-6}
Training Time (hours)	31.3	446.1	5.9	42.1	2.0	8.3
Model Initialization	Megatron	Megatron	—	—	—	—
Dropout (attn out)	0.1					
Dropout (all other layers)	0.0					
Optimizer	StableAdamW					
Betas	(0.90, 0.98)					
Epsilon	1×10^{-6}					
Training Hardware	16× H100					
Training Strategy	Distributed DataParallel, bfloat16					

Table 5: ModernGBERT training settings. Dropout and below are shared across all phases.

	LLäMmlein2Vec 120M		LLäMmlein2Vec 1B		LLäMmlein2Vec 7B	
	Ext1	Ext2	Ext1	Ext2	Ext1	Ext2
Training Tokens	52B	14.4B	90B	14.4B	90B	14.4B
Max Sequence Length	8,192		8,192		8,192	
RoPE theta	160,000		160,000		160,000	
Batch Size	32	32	32	32	16	16
Training Hardware	64× H200		64× H200		256 × H200	128 × H200
Training Duration	10h41	3h40	37h24	6h40	14h25†	9h39

Table 6: LLäMmlein2Vec training settings. Due to limited resources we had to terminate the 7B model training on the first extension dataset early †.

type	model	params	classification						tagging			similarity	QA	Average
			tox. macro F1	sent. micro F1	match ACC	WSD micro F1	other mixed	avg mixed	NER micro F1	other micro F1	avg micro F1	pearson corr	m. t. F1	mixed
enc	$GBERT_{Base}$	111M	0.537	0.620	0.738	0.814	0.749	0.723	0.705	0.806	0.786	0.561	0.803	0.718
enc	$GBERT_{Large}$	337M	0.604	0.673	0.810	0.837	0.816	0.785	0.744	0.813	0.799	0.654	0.832	0.768
enc	gerturax-3	135M	0.561	0.617	0.788	0.804	0.762	0.736	0.707	0.810	0.789	0.603	0.832	0.740
enc	$GeBERTa_{Base}$ †	139M	0.530	0.619	0.766	0.789	0.737	0.715	0.687	0.801	0.778	0.559	0.813	0.716
enc	$GeBERTa_{Large}$ †	406M	0.551	0.623	0.783	0.795	0.775	0.743	0.736	0.804	0.791	0.619	0.844	0.749
enc	$GeBERTa_{XLarge}$ †	887M	0.606	0.671	0.796	0.830	0.795	0.770	0.763	0.818	0.807	0.643	0.848	0.767
enc	$XLM\text{-}RoBERTa_{Base}$	279M	0.530	0.546	0.741	0.790	0.717	0.693	0.648	0.780	0.754	0.505	0.802	0.689
enc	$XLM\text{-}RoBERTa_{Large}$	561M	0.512	0.559	0.795	0.814	0.739	0.714	0.720	0.804	0.787	0.583	0.837	0.730
enc	$XLM\text{-}RoBERTa_{XLarge}$	3.48B	0.558	0.612	0.815	0.820	0.778	0.750	0.746	0.816	0.802	0.656	0.822	0.758
dec	Llama 3.2	1B	0.519	0.629	0.768	0.808	0.768	0.737	0.648	0.733	0.716	0.551	0.835	0.710
dec	Llama 3.1	8B	0.586	0.713	0.819	0.849	0.817	0.790	0.708	0.753	0.744	0.573	0.868	0.744
dec	LLäMmlein	120M	0.510	0.580	0.710	0.798	0.732	0.702	0.613	0.737	0.712	0.477	0.812	0.676
enc	LLäMmlein2Vec (1/5 ext1)	120M	0.480	0.571	0.696	0.793	0.738	0.700	0.627	0.754	0.729	0.471	0.821	0.680
enc	LLäMmlein2Vec (ext1)	120M	0.491	0.564	0.711	0.797	0.739	0.703	0.660	0.761	0.741	0.472	0.819	0.684
enc	LLäMmlein2Vec (ext2)	120M	0.471	0.568	0.686	0.772	0.739	0.697	0.620	0.757	0.730	0.455	0.818	0.675
enc	LLäMmlein2Vec (ext1+2)	120M	0.448	0.490	0.581	0.749	0.708	0.657	0.505	0.728	0.683	0.439	0.730	0.627
dec	LLäMmlein	1B	0.603	0.710	0.790	0.839	0.806	0.781	0.728	0.785	0.773	0.548	0.828	0.733
enc	LLäMmlein2Vec (1/5 ext1)	1B	0.587	0.694	0.827	0.820	0.806	0.779	0.795	0.815	0.811	0.615	0.842	0.762
enc	LLäMmlein2Vec (ext1)	1B	0.575	0.702	0.775	0.835	0.808	0.776	0.790	0.818	0.812	0.615	0.843	0.762
enc	LLäMmlein2Vec (ext2)	1B	0.528	0.688	0.754	0.815	0.789	0.755	0.775	0.812	0.804	0.580	0.836	0.744
enc	LLäMmlein2Vec (ext1+2)	1B	0.503	0.566	0.692	0.787	0.732	0.697	0.727	0.799	0.784	0.557	0.828	0.717
dec	LLäMmlein	7B	0.632	0.739	0.821	0.873	0.835	0.810	0.796	0.808	0.805	0.524	0.851	0.747
enc	LLäMmlein2Vec (ext1)	7B	0.633	0.742	0.813	0.839	0.822	0.799	0.851	0.835	0.838	**0.670**	0.842	0.787
enc	LLäMmlein2Vec (ext2)	7B	0.612	0.743	0.808	0.848	0.822	0.797	0.848	0.834	0.837	0.657	0.842	0.783
enc	LLäMmlein2Vec (ext1+2)	7B	0.554	0.647	0.610	0.786	0.744	0.709	0.747	0.804	0.793	0.639	0.821	0.740
enc	ModernGBERT	134M	0.503	0.617	0.769	0.780	0.744	0.716	0.734	0.794	0.782	0.589	0.833	0.730
enc	ModernGBERT + ext1+2	134M	0.526	0.647	0.779	0.806	0.760	0.735	0.791	0.809	0.805	0.621	0.836	0.749
enc	ModernGBERT	1B	0.610	**0.746**	**0.827**	0.858	0.833	0.806	0.849	0.837	0.839	0.681	0.874	0.800
enc	ModernGBERT + ext1+2	1B	**0.635**	0.745	0.826	**0.876**	**0.836**	**0.812**	**0.868**	**0.840**	**0.845**	0.699	**0.876**	**0.808**

Table 7: SuperGLEBer results, averaged at varying levels of granularity, following (Pfister and Hotho, 2024). The columns reading "Average" have been averaged across the averages of the respective task types, in order to not overweight any task type for which more datasets exist. † indicates that these models were evaluated without fp16. ModernGBERT models marked with "w/o ext." refer to the model checkpoint after pre-training but before the context extension phases. We also exemplary evaluated LLM2Vec after 20% of LONG Head or Head/Middle training data ("+ 1/5 ext1").

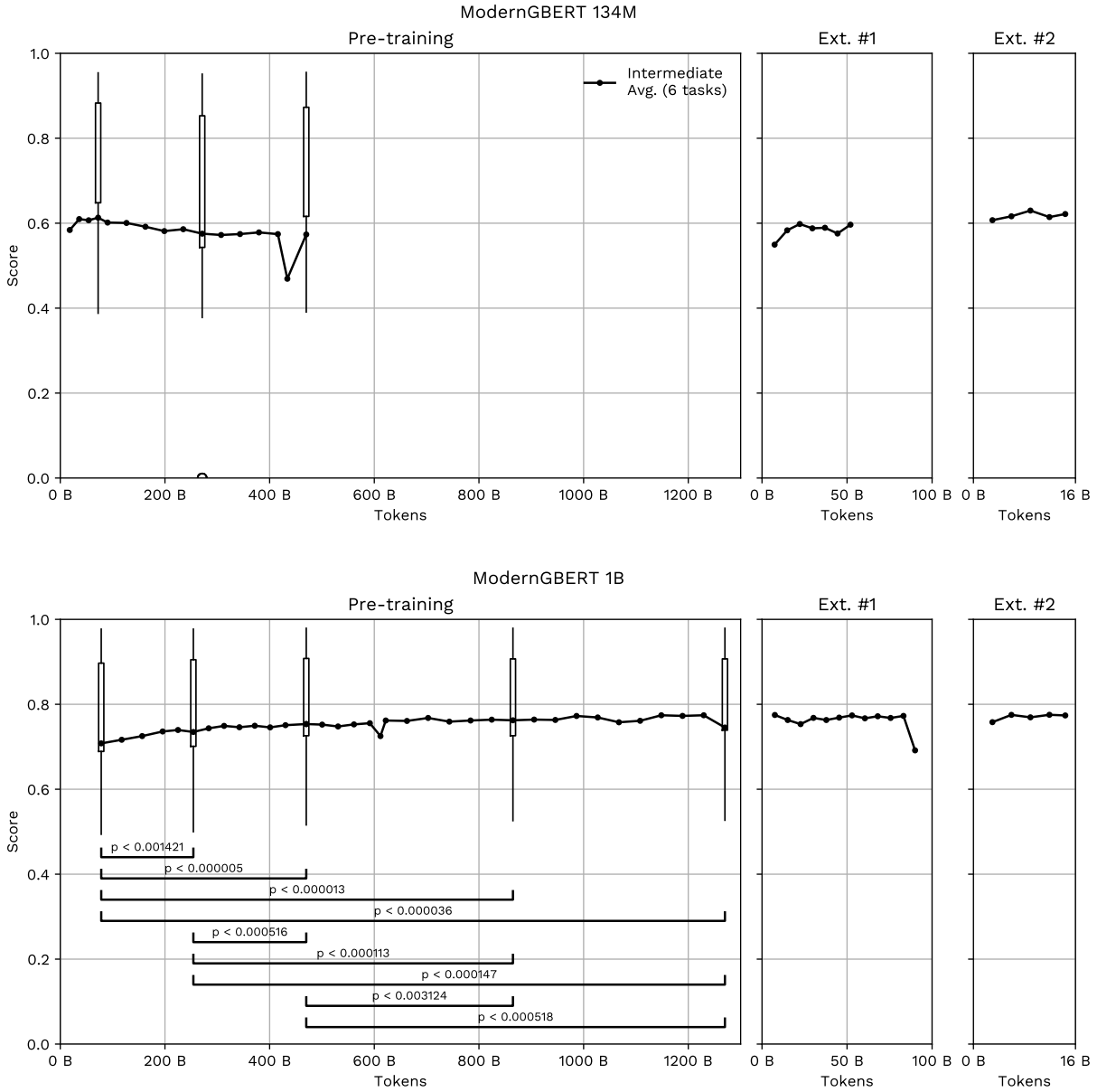

Figure 3: Intermediate Checkpoint Evaluation. Note that the solid line represents the mean of the six tasks selected for the intermediate checkpoint evaluation (NLI, FactClaiming Comments, DB Aspect, WebCAGe, EuroParl, PAWSX Similarity). The box plots show the distribution of scores for those checkpoints evaluated across all 29 SuperGLEBer tasks. We compared each pair of these checkpoints using Wilcoxon signed-rank tests, and highlighted significant increases with brackets. Brackets of pairs without significant increases are not displayed. (Accordingly, all pairs of 134M checkpoints show no significant increase.) Four checkpoints failed to converge during fine-tuning on some task, leading to the visible outliers. Similar behavior has been observed by Antoun et al. (2025).

Category	Task	Metric	Reference
Classification	AmazonCounterfactual	Accuracy	O'Neill et al. (2021)
	AmazonReviews	Accuracy	Keung et al. (2020)
	MTOPDomain	Accuracy	Li et al. (2021)
	MTOPIntent	Accuracy	Li et al. (2021)
	MassiveIntent	Accuracy	FitzGerald et al. (2023)
	MassiveScenario	Accuracy	FitzGerald et al. (2023)
PairClassification	FalseFriendsGermanEnglish	Average Precision	Chibb (2022)
	PawsXPairClassification	Average Precision	Yang et al. (2019)
Clustering	BlurbsClusteringP2P	V-measure	Wehrli et al. (2023)
	BlurbsClusteringS2S	V-measure	Wehrli et al. (2023)
	TenKGnadClusteringP2P	V-measure	Wehrli et al. (2023)
	TenKGnadClusteringS2S	V-measure	Wehrli et al. (2023)
Reranking	MIRACLReranking	nDCG@10	Zhang et al. (2023)
Retrieval	GermanQuAD-Retrieval	MRR@5	Möller et al. (2021)
	GermanDPR	DCG@10	Möller et al. (2021)
	Xmarket	nDCG@10	Bonab et al. (2021)
	GerDaLIR	nDCG@10	Wrzalik and Krechel (2021)
STS	GermanSTSBenchmark	Spearman	May (2021); Cer et al. (2017)
	STS22	Spearman	Chen et al. (2022)

Table 8: Overview of tasks included in the German *MTEB(deu, v1)* benchmark, grouped by six categories: classification, pairclassification, clustering, reranking, retrieval and STS.

Model	Params	Classification Average	PairClassification Average	Clustering Average	Reranking Average	Retrieval Average	STS Average	Average
$GBERT_{Base}$	111M	0.634	0.504	0.274	0.118	0.226	0.402	0.360
$GBERT_{Base}$ †		0.632	0.601	0.318	0.374	0.461	0.613	0.500
$GBERT_{Large}$	337M	0.649	0.544	0.336	0.206	0.297	0.438	0.412
$GBERT_{Large}$ †		0.646	**0.662**	0.334	0.389	0.493	0.603	0.521
gerturax-3	135M	0.623	0.554	0.269	0.141	0.187	0.375	0.358
gerturax-3†		0.620	0.614	0.328	0.346	0.389	0.538	0.472
$GeBERTa_{Base}$	139M	0.632	0.535	0.312	0.174	0.213	0.429	0.382
$GeBERTa_{Base}$ †		0.611	0.613	0.318	0.374	0.430	0.611	0.493
$GeBERTa_{Large}$	406M	0.642	0.533	0.287	0.223	0.274	0.424	0.397
$GeBERTa_{Large}$ †		0.618	0.611	0.311	0.374	0.432	0.616	0.494
$GeBERTa_{XLarge}$	887M	0.626	0.536	0.278	0.108	0.058	0.342	0.325
$GeBERTa_{XLarge}$ †		0.638	0.631	0.323	0.414	0.462	0.655	0.521
$XLM\text{-}RoBERTa_{Base}$	279M	0.442	0.506	0.173	0.024	0.008	0.333	0.248
$XLM\text{-}RoBERTa_{Base}$ †		0.555	0.529	0.247	0.247	0.299	0.539	0.403
$XLM\text{-}RoBERTa_{Large}$	561M	0.510	0.510	0.172	0.048	0.026	0.320	0.264
$XLM\text{-}RoBERTa_{Large}$ †		0.576	0.574	0.259	0.343	0.416	0.593	0.460
$XLM\text{-}RoBERTa_{XLarge}$	3.48B	0.456	0.519	0.225	0.090	0.142	0.372	0.301
$XLM\text{-}RoBERTa_{XLarge}$ †		0.609	0.564	0.342	0.362	0.407	0.590	0.479
LLäMmlein2Vec (ext1)	120M	0.546	0.529	0.261	0.139	0.224	0.188	0.315
LLäMmlein2Vec † (ext1)	120M	0.599	0.575	0.308	0.325	0.425	0.592	0.471
LLäMmlein2Vec (ext2)	120M	0.457	0.525	0.202	0.118	0.117	0.205	0.271
LLäMmlein2Vec † (ext2)	120M	0.607	0.588	0.295	0.305	0.339	0.498	0.439
LLäMmlein2Vec (ext1+2)	120M	0.339	0.530	0.098	0.046	0.009	0.107	0.188
LLäMmlein2Vec † (ext1+2)	120M	0.517	0.563	0.263	0.251	0.355	0.554	0.417
LLäMmlein2Vec (ext1)	1B	0.641	0.542	0.308	0.183	0.276	0.442	0.399
LLäMmlein2Vec † (ext1)	1B	0.670	0.625	**0.343**	0.433	0.511	0.660	0.540
LLäMmlein2Vec (ext2)	1B	0.617	0.541	0.299	0.189	0.280	0.431	0.393
LLäMmlein2Vec † (ext2)	1B	0.666	0.622	0.330	0.433	0.499	0.644	0.532
LLäMmlein2Vec (ext1+2)	1B	0.337	0.538	0.075	0.062	0.010	0.217	0.206
LLäMmlein2Vec † (ext1+2)	1B	0.647	0.611	0.325	0.421	0.481	0.640	0.521
LLäMmlein2Vec (ext1)	7B	0.683	0.558	0.249	0.169	0.266	0.333	0.376
LLäMmlein2Vec † (ext1)	7B	**0.687**	0.636	0.339	**0.477**	**0.522**	**0.682**	**0.557**
LLäMmlein2Vec (ext2)	7B	0.679	0.555	0.323	0.187	0.309	0.462	0.419
LLäMmlein2Vec † (ext2)	7B	0.683	0.628	0.337	0.471	0.517	0.680	0.553
LLäMmlein2Vec (ext1+2)	7B	0.349	0.525	0.072	0.047	0.005	0.182	0.197
LLäMmlein2Vec † (ext1+2)	7B	0.677	0.615	0.327	0.460	0.506	0.663	0.541
ModernGBERT	134M	0.639	0.537	0.293	0.139	0.241	0.449	0.383
ModernGBERT †		0.602	0.606	0.303	0.364	0.432	0.602	0.485
ModernGBERT + ext1+2	134M	0.642	0.536	0.296	0.120	0.213	0.445	0.376
ModernGBERT † + ext1+2		0.629	0.612	0.312	0.404	0.446	0.606	0.501
ModernGBERT	1B	0.665	0.544	0.318	0.097	0.199	0.418	0.374
ModernGBERT †		0.659	0.641	0.339	0.463	0. 511	0.681	0.549
ModernGBERT + ext1+2	1B	0.659	0.540	0.307	0.088	0.191	0.410	0.366
ModernGBERT † + ext1+2		0.659	0.654	0.338	0.459	0.513	**0.682**	0.551

Table 9: Results on *MTEB(deu, v1)* of the German MTEB Benchmark. For each task type, scores were averaged across respective unique tasks. We provide results for basis models as well as after supervised training on mMARCO †. In all cases, evaluation was done in a zero-shot fashion without further finetuning on the above tasks. Best scores are indicated in bold.

Model	Params	<1,024 tok.	1,024 to 4,095 tok.	4,096 to 8,192 tok.	Overall
LLäMmlein	120M	0.286	0.124	0.049	0.091
LLäMmlein	1B	0.517	0.230	0.088	0.165
LLäMmlein	7B	0.529	0.310	0.122	0.216
LLäMmlein2Vec (ext1)	120M	0.315	0.206	0.044	0.120
LLäMmlein2Vec (ext2)	120M	0.252	0.047	0.000	0.031
LLäMmlein2Vec (ext1+2)	120M	0.055	0.001	0.000	0.003
LLäMmlein2Vec (ext1)	1B	0.588	0.448	0.232	0.333
LLäMmlein2Vec (ext2)	1B	0.555	0.297	0.003	0.144
LLäMmlein2Vec (ext1+2)	1B	0.462	0.209	0.033	0.123
LLäMmlein2Vec (ext1)	7B	0.597	0.207	0.000	0.111
LLäMmlein2Vec (ext2)	7B	0.605	0.176	0.000	0.099
LLäMmlein2Vec (ext1+2)	7B	0.580	0.327	0.000	0.156
ModernGBERT	134M	0.552	0.168	0.013	0.105
ModernGBERT + ext1	134M	0.536	0.410	0.238	0.323
ModernGBERT + ext1+2	134M	0.540	0.393	0.201	0.296
ModernGBERT	1B	0.556	0.233	0.023	0.136
ModernGBERT + ext1	1B	**0.617**	0.506	**0.406**	**0.457**
ModernGBERT + ext1+2	1B	0.601	**0.526**	0.383	0.451

Table 10: QA-NIAH results. Metric is Exact Match. All tokens are counted per the model's respective tokenizer. ModernGBERT models marked with "w/o ext" refer to the model checkpoint after pre-training but before the context extension phases, those marked with "w/o ext2" to the models after extension phase one, but before phase two.

		Short		Long	
Model	Params	Fixed Length	Variable Length	Fixed Length	Variable Length
$\text{GBERT}_{\text{Base}}$	111M	2.33 ± 0.13	5.25 ± 0.27	–	–
$\text{GBERT}_{\text{Large}}$	337M	7.25 ± 0.77	15.70 ± 1.66	–	–
gerturax-3	135M	4.13 ± 0.40	8.26 ± 0.81	–	–
$\text{GeBERTa}_{\text{Base}}$ †	139M	9.79 ± 0.04	19.40 ± 0.08	–	–
$\text{GeBERTa}_{\text{Large}}$ †	406M	27.30 ± 0.09	54.10 ± 0.40	–	–
$\text{GeBERTa}_{\text{XLarge}}$ †	887M	42.20 ± 0.36	83.80 ± 0.70	–	–
$\text{XLM-RoBERTa}_{\text{Base}}$	279M	2.28 ± 0.08	5.05 ± 0.19	–	–
$\text{XLM-RoBERTa}_{\text{Large}}$ †	561M	7.27 ± 0.53	15.90 ± 1.04	–	–
$\text{XLM-RoBERTa}_{\text{XLarge}}$ †	3.48B	57.70 ± 0.37	123.00 ± 0.71	–	–
LLäMmlein2Vec	120M	3.74 ± 0.75	7.17 ± 0.53	6.69 ± 0.14	8.39 ± 0.35
LLäMmlein2Vec	1B	27.30 ± 0.16	53.90 ± 0.37	42.70 ± 0.12	59.70 ± 0.30
LLäMmlein2Vec	7B	143.00 ± 0.22	288.00 ± 0.52	180.00 ± 0.19	304.00 ± 0.41
ModernGBERT	134M	3.60 ± 0.29	3.70 ± 0.74	5.42 ± 0.33	4.71 ± 0.75
ModernGBERT	1B	22.60 ± 0.40	22.50 ± 0.18	28.70 ± 0.31	26.20 ± 0.36

Table 11: Model Throughput. Numbers are seconds per million tokens. All models were run on an RTX A6000 with Bfloat16 data type and with Flash Attention 2, except models with †, which did not implement Flash Attention 2. Reported uncertainty is the empirical standard deviation on 10 repetitions.