

MoganBert-TR: A Turkish Encoder Foundation Model Trained from Scratch with a CLM→MLM Curriculum

Furkan Yılmaz
furkanyl509@gmail.com

Habibe Aleyna Taşdemir
aleynattasdemir@gmail.com

Muhammed Faruk Gözay
gozayfaruk@gmail.com

Abstract

Publicly available large-scale pretraining data for Turkish exists largely as a subset of multilingual dumps and involves no language-specific quality filtering; existing Turkish encoders, meanwhile, have adopted modern architectures while leaving the pretraining objective fixed. This paper introduces an encoder foundation model trained from scratch on a language-specifically filtered corpus (MoganBert-TR, 149M parameters), together with an embedding model derived from it (MoganBert-Embed).

(1) Training objective. MoganBert-TR is trained over 237.3 billion tokens with a two-stage CLM→MLM curriculum: the first portion of the run uses causal language modelling and the remainder masked language modelling, with the transition made inside the stable phase of a WSD schedule and without touching the learning rate. In a controlled ablation on the same architecture and the same corpus under an equal step budget, this design outperforms pure MLM by $2.7\text{--}3.7\times$ on Turkish MS MARCO retrieval; the measured mechanism is embedding geometry a single direction absorbs 28.1% of the variance in the pure-MLM model, against 11.9% under the curriculum. **(2) Branched annealing.** Long-context extension and learning-rate decay are split into two branches after a shared prefix; running the final portion of decay at 1024 context improves the TrGLUE average by $+0.49 \pm 0.26$ points across five paired seeds ($p \approx 0.013$) and beats a model-soup alternative by 0.75 points at $\sim 4.3\%$ additional cost. **(3) Results.** MoganBert-TR attains 78.41 on TrGLUE, the best among the Turkish ModernBERT models compared here, and 77.73 on TabiBench, second among monolingual Turkish encoders and 0.19 points behind the leader; it leads two of the eight TabiBench categories, with the largest margin on code retrieval ($+3.62$ points over TabiBERT). **(4) Embedding model.** MoganBert-Embed, produced through teacher distillation and multi-signal contrastive fine-tuning, ranks first among student models on the MTEB(Turkish) overall average with 68.30 and reaches 99.5% of its 7.57-billion-parameter teacher’s score with a $51\times$ smaller backbone. **(5) Data and tokenizer.** The Turkish portion of FineWeb2, recent months pulled from raw Common Crawl, and domain-dense text extracted from printed and institutional sources with a vision-language model were combined into a single pipeline; since no off-the-shelf Turkish quality classifier exists, the decisions of a fine-tuned Turkish BERT were distilled into fastText (94.4% agreement, $\sim 90\times$ speed-up). The accompanying 50,048-token tokenizer outperforms all compared Turkish tokenizers on both compression and fertility across two independent test sets.

Model weights, tokenizer, embedding model and evaluation code will be released openly at <https://huggingface.co/moganai>.

Keywords: Turkish natural language processing, encoder language models, pretraining objectives, tokenization, knowledge distillation, embedding models

1 Introduction

Despite the visibility of generative models, encoder-only models remain the de facto backbone of classification, sequence labelling and retrieval pipelines: their inference cost is low, they see bidirectional context in a single pass, and they are natural candidates for producing embeddings. ModernBERT [28] updated the architectural side of this family with RoPE [24], alternating local/global attention and GLU; the language-specific adaptations that followed demonstrated that the architecture transfers beyond English.

On the Turkish side this transfer has been carried out only partially. BERTurk [21] long served as the de facto standard, and models such as TabiBERT [26] and ModernBERT-TR [1] have brought the ModernBERT architecture to Turkish. What these works share, however, is the assumption that the pretraining objective is fixed: pure MLM. While the architecture was being modernised, the *training objective* itself was never systematically tested for Turkish.

This gap is not trivial. Controlled comparisons of encoder pretraining objectives [7] report that, under a fixed compute budget, a two-stage CLM→MLM design outperforms pure MLM; yet that finding was produced at English scale, and whether it holds for a morphologically rich, agglutinative language with its own tokenizer and its own corpus remains an open question. Likewise, *how* long-context extension should be carried out during the annealing phase has no established recipe: mmBERT [13] shows that altering the data mixture of the decay phase is decisive, but the role of context length within that same phase has not been measured.

This paper addresses both questions through MoganBert-TR, an encoder foundation model trained from scratch for Turkish (149M parameters). The model is trained over 237.3 billion tokens with a two-stage CLM→MLM curriculum; the contribution of that design is measured through a controlled ablation on the same architecture and the same corpus; and the annealing phase is branched in order to test the interaction between long-context extension and learning-rate decay across paired seeds. Training uses a language-specifically filtered corpus; the data pipeline and the tokenizer were produced for this purpose and are transferable contributions in their own right.

Contributions.

- **A training-objective ablation for Turkish.** Under an equal step budget, CLM→MLM is tested against pure MLM. While the two designs are practically tied on linear probing tasks, the curriculum leads by $2.7\text{--}3.7\times$ on Turkish MS MARCO retrieval. The mechanism is attributed to embedding geometry: a single direction absorbs 28.1% of the variance in the pure-MLM model, against 11.9% under the curriculum (Section 5.3).
- **Branched annealing.** Long-context extension and learning-rate decay are split into two branches after a shared prefix. Running the final portion of decay at 1024 context improves the TrGLUE average by $+0.49 \pm 0.26$ points across five paired seeds ($p \approx 0.013$; all five seeds positive) and beats a model-soup alternative by 0.75 points at $\sim 4.3\%$ additional cost (Section 6.2).
- **MoganBert-TR.** With 78.41 on TrGLUE it is the best of the Turkish ModernBERT models compared here; on TabiBench it scores 77.73, second among monolingual Turkish encoders behind ModernBERT-TR (77.92) and ahead of TabiBERT (77.58), and leads two of the eight categories (Section 6.4). The 1.46-point margin by which BERTurk leads is concentrated almost entirely in the known weak spots of an NSP-free architecture (CoLA, STS-B), analysed separately (Section 6.5).
- **MoganBert-Embed.** The embedding model produced through teacher distillation and multi-signal contrastive fine-tuning ranks first among student models on the MTEB(Turkish)

overall average with 68.30 and reaches 99.5% of its 7.57-billion-parameter teacher’s score with a $51\times$ smaller backbone (Section 7).

- **Data pipeline and tokenizer.** A distillation chain that, in a language without an off-the-shelf quality classifier, uses a slow but accurate teacher as a label generator (94.4% agreement, $\sim 90\times$ speed-up), and a 50,048-token tokenizer that outperforms all compared Turkish tokenizers across two independent test sets (Sections 3 and 4).
- **Open release.** Model weights, tokenizer, embedding model and evaluation code will be released at <https://huggingface.co/moganai>.

The remainder of the paper is organised as follows: Section 2 reviews prior work; Section 3 introduces the pretraining data and Section 4 the tokenizer; Section 5 presents the model architecture, the training-objective ablation and the full pretraining run; Section 6 reports evaluation results and Section 7 the embedding model; Section 8 discusses the findings, Section 9 lists limitations, and Section 10 concludes. Implementation details are given in the appendices.

2 Related Work

2.1 The evolution of encoder architectures

The encoder-only paradigm established by BERT [6] has been updated over subsequent years by a series of efficiency- and stability-oriented improvements. ModernBERT [28] consolidates this accumulation into a single recipe: long-context support via RoPE [24], alternating local/global attention, GLU activations, unpadding and FlashAttention-based kernels. The result is an 8192-token context and a marked inference speed-up. This architecture has also become the de facto basis for language-specific adaptations.

2.2 Turkish encoder models

BERTurk [21], the first monolingual BERT trained from scratch for Turkish, long served as the de facto standard; because it is not the product of an academic publication, its corpus composition and preprocessing decisions remain partly unspecified. TabiBERT [26] is the first work to bring the ModernBERT architecture to Turkish and to release it together with a standardised evaluation framework. ModernBERT-TR [1] uses the same architecture. On the multilingual side, mBERT [6] and XLM-R [5] include Turkish as one language among many, while mmBERT [13] scales the ModernBERT architecture to more than 1800 languages.

All of these models monolingual and multilingual alike use pure MLM as their pretraining objective. This is the point at which MoganBert-TR departs from them.

2.3 Pretraining objective and training schedule

That pure MLM is not the only option for encoder pretraining has recently been demonstrated under controlled conditions: Gisserot-Boukhlef et al. [7], using 38 models (210M–1B) and more than 15,000 fine-tuning runs, report that under a fixed compute budget a two-stage CLM→MLM design consistently outperforms pure MLM. For the 610M model, the sequence classification average is 87.00 for 100% MLM, 87.85 for 25% CLM + 75% MLM, 87.58 for 50%/50%, and 83.58 for 100% CLM. The proposed mechanism is that CLM is more data-efficient in early steps and produces representations less sensitive to the learning rate during fine-tuning, whereas MLM eventually pulls ahead on representation quality. The critical implementation detail is that the MLM phase resumes from a CLM checkpoint that has not yet entered learning-rate decay; in a WSD schedule [8] the transition is made inside the stable phase.

The final phase of the schedule (annealing/decay) is an independent design space. mm-

BERT [13] shows that data shown in the low-learning-rate region is disproportionately influential, and that adding low-resource languages only during this phase can yield large gains. The role of *context length* within that same phase, however, has not been measured; Section 6.2 fills this gap.

2.4 Data pipelines and quality filtering

The methodological choices behind the pipeline described below follow the patterns established by CCNet [29] and FineWeb/FineWeb2 [18, 19]: direct text extraction from WARC, fastText-based language identification, and the application of quality and repetition filters as a sequential chain. FineWeb-Edu [12] demonstrated that a quality classifier learned from LLM labels can be used at scale. The essential difference in adapting this to Turkish is the absence of an off-the-shelf Turkish quality classifier and the resulting need to build one from scratch (Section 3.2).

2.5 Embedding models

The raw representations of MLM-pretrained encoders are anisotropic: all vectors collect within a narrow cone, and cosine similarity buries the discriminative signal in noise. This pattern has been characterised in the literature as a “rogue dimension” [25]. Contrastive fine-tuning and teacher distillation [9] are the established ways of correcting this geometry; MTEB [15] is used as the evaluation ground.

3 Pretraining Data

3.1 Sources and processing pipeline

The corpus was compiled from three sources. The first is the Turkish portion of FineWeb2 [19]; rather than being used as-is, it was passed again through the Turkish quality filter described below. The second consists of the recent months that collection does not cover: Turkish records are first indexed via each monthly crawl’s columnar index, after which only the relevant byte ranges are fetched with HTTP Range GET the WARC files are not downloaded in full. Main text is extracted from raw WARC records with `trafilatura`; the ready-made WET text was deliberately not used because it carries boilerplate and menu residue (CCNet [29] and FineWeb [18] follow the same route).

After extraction, language verification (`fastText lid.176.ftz`), Gopher/C4-style heuristic quality filters, a Turkish boilerplate check, PII masking and learned quality classification are applied in a single pass; the output is written directly as clean JSONL. The final step is MinHash-based fuzzy deduplication [4] (n-gram 5, 14 buckets, 8 hashes per bucket), which operates incrementally across months: each month is first deduplicated within itself and then against the accumulated corpus. The processing rate measured in production is $\sim 430\text{--}590$ records/second, and the acceptance rate falls in the 29%–55% band depending on the month.

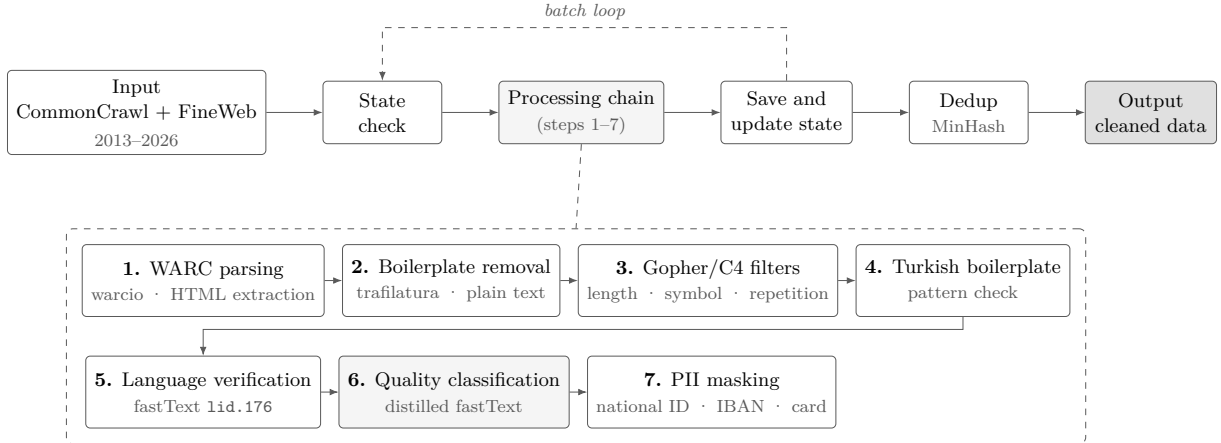


Figure 1: The data pipeline: from FineWeb2 and Common Crawl input through a single-pass processing chain to the deduplicated output. Quality classification (step 6) is not a separate pass but part of the same process.

3.2 The quality classifier: from teacher to label generator

The only Turkish-specific gap in the pipeline was the quality classifier: heuristic filters do not remove text that is grammatically valid but worthless (gambling, SEO, advertising), and no off-the-shelf Turkish quality classifier exists.

A three-stage chain was built. A fastText classifier trained with off-the-shelf word vectors failed by discarding legitimate short text as well; retrained on $\sim 6,000$ documents scored by an LLM against a 0–5 rubric (following the FineWeb-Edu [12] pattern), fastText remained at $\sim 54\%$ validation accuracy. A Turkish BERT fine-tuned on the same labels reached 70.7% exact and 93.9% ± 1 accuracy, but at ~ 96.5 documents/second it was unsuitable for production.

The solution was to use BERT not as the production classifier but as a *label generator*: a binary fastText model was trained from scratch on the $\sim 480,000$ keep/discard decisions accumulated across the processed chunks. On a chunk the teacher had never seen, the distilled model gave **94.4% agreement** and $\sim 8,880$ documents/second a $\sim 90\times$ speed-up with no loss of accuracy [9]. This pattern transfers directly to any language lacking an off-the-shelf quality classifier.

3.3 Printed and institutional sources

Alongside the web branch, a second track was added for domain-dense, editorially reviewed content: text extraction from books, theses and academic/legal publications. The decisive constraint here is not transcription accuracy but *structural classification* the cover page, copyright notice, preface and table of contents are not the content of a book and must not enter the corpus. Classical OCR tools cannot make this distinction because they decide on headings from font size and position; a vision-language model that solves both layers in a single request was therefore used instead. As the output format, line-tagged plain text was preferred over JSON; JSON produced 91 parse errors over 448 pages because of quotation marks and special characters in Turkish text. In post-processing, front-matter blocks are discarded, identical section headings reopened at page boundaries are merged, and text is chunked with a target of $\sim 6,000$ characters. Details are given in Appendix A.

4 Tokenizer

The tokenizer is a 50,048-token SentencePiece Unigram model [11]. This section presents three design decisions and the measured efficiency; implementation details such as the normalization

pipeline and code sensitivity are given in Appendix B.

4.1 Vocabulary size

Three vocabulary sizes were trained and measured on TR-MMLU (6,200 questions, 1,743,375 characters). The linguistic metrics (TR% = the proportion of valid Turkish unique tokens, Pure% = the proportion of atomic tokens) follow the definitions of Bayram et al. [3], which reports a correlation of $r = 0.90$ between TR% and MMLU score. Dilbaz (60,000+ roots, FST + rule engine) was used for morphological analysis.

Table 1: Tokenizer efficiency on TR-MMLU.

Tokenizer	Vocab	Fertility ↓	Chars/Token ↑	TR% ↑	Pure% ↑
Mogan-64K	64,000	1.5268	5.0245	84.71	40.83
Mogan-50K	50,000	1.5682	4.8918	84.42	42.67
Mürşit	59,008	1.6326	4.6989	77.00	35.81
MBERT-tr	32,000	1.7037	4.5029	78.73	35.65
BERTurk	32,000	1.8008	4.2600	81.06	43.22
TabiBERT	50,176	1.8501	4.1464	59.84	30.46
XLM-R	250,002	1.9440	3.9462	56.09	41.78

The 64K vocabulary beats 50K on every efficiency metric (fertility 1.527 vs 1.568), but this gain comes at the cost of embedding parameters: in a model with hidden size 768, 14,000 additional tokens amount to $\approx 10.7\text{M}$ additional parameters roughly 7% of the parameter budget of a 150M foundation model. Since a 2.7% fertility difference does not justify that cost, the 50K base was chosen. The final size is **50,048**: the multiple of 64 closest to 50,000; GPU tensor cores operate more efficiently at sizes that are multiples of 64.

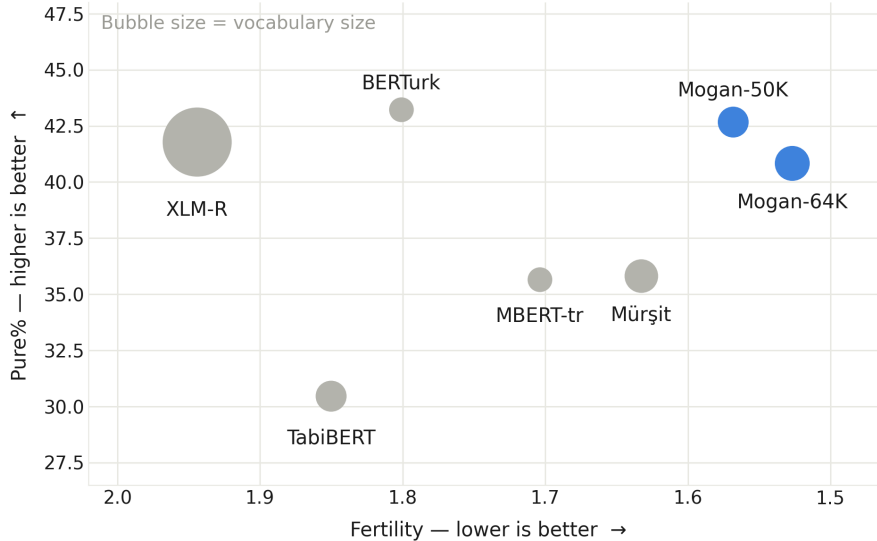


Figure 2: Fertility (x , lower is better) vs Pure% (y , higher is better); bubble size indicates vocabulary size.

4.2 Validation on independent test sets

In addition to TR-MMLU, the tokenizer was tested on two further sources entirely independent of the data collected for this project: Turkish Wikipedia (20231101.tr, 30,000 documents) and FLORES-200 [17] (tur_Latn, dev+devtest, 2,009 sentences), the academic standard for

multilingual evaluation. In both tests the same document set was applied to all tokenizers under the same seed (42).

Table 2: Tokenizer efficiency on independent test sets.

Test set	Tokenizer	Vocab	Chars/Token $\uparrow$	Fertility $\downarrow$	UNK
Wikipedia	Mogan-64K	64,000	4.705	1.640	0.00%
Wikipedia	BERTurk	32,000	4.391	1.757	0.04%
Wikipedia	Kumru-2B	50,176	4.010	1.924	0.00%
Wikipedia	TabiBERT	50,176	4.010	1.924	0.00%
FLORES-200	Mogan-64K	64,000	5.277	1.460	0.00%
FLORES-200	BERTurk	32,000	4.932	1.563	0.002%
FLORES-200	TabiBERT	50,176	4.900	1.573	0.00%
FLORES-200	Kumru-2B	50,176	4.899	1.573	0.00%

On both independent test sets the Mogan tokenizer outperforms all compared tokenizers on compression as well as on fertility.

4.3 Code sensitivity and special tokens

In code data, indentation is part of the syntax; because standard normalization removes it, different programs can collapse onto the same token sequence. With an indentation-preserving pipeline and the `allow_whitespace_only_pieces` setting, the model learned 11 multi-space tokens ranging from 3 to 16 spaces.

Table 3: Tokenizer comparison on a code corpus (18 languages, 540 files). Lossless roundtrip is the rate at which a model can reproduce its own output.

Tokenizer	Code compression	Lossless roundtrip
Mogan (final)	2.790	100%
Mürşit	2.454	96%
TabiBERT	2.196	49%
BERTurk	2.184	0%

In the placement of special tokens the single critical decision is keeping `[MASK]` in the `control_symbols` list: web data contains documents that write `[MASK]` literally, and if the other list is used that text is converted into the actual mask identifier, injecting spurious masks into the training data. The final layout and the rejected alternatives are given in Appendix B.

5 Model and Pretraining

5.1 Architecture

The model follows the ModernBERT-base configuration: 22 layers, hidden size 768, alternating local/global attention (local window 128), RoPE and GLU; 149.4M parameters in total. Since HuggingFace’s ModernBERT implementation is bidirectional only, the attention path was rewritten to support the CLM phase. This rewrite involves four pitfalls capable of silently producing an incorrect model (causal local window, boundary-aware packing, position-id resetting, document-boundary masking in the loss); these are documented together with the memory and numerical decisions in Appendix C.

5.2 Training objective: the CLM→MLM curriculum

The first portion of the run is trained with the causal language modelling (CLM) objective and the remainder with masked language modelling (MLM). The transition is made inside the stable phase of a WSD schedule [8] and without touching the learning rate; decay comes only at the very end. At the same step, the attention window is switched from causal to bidirectional mode.

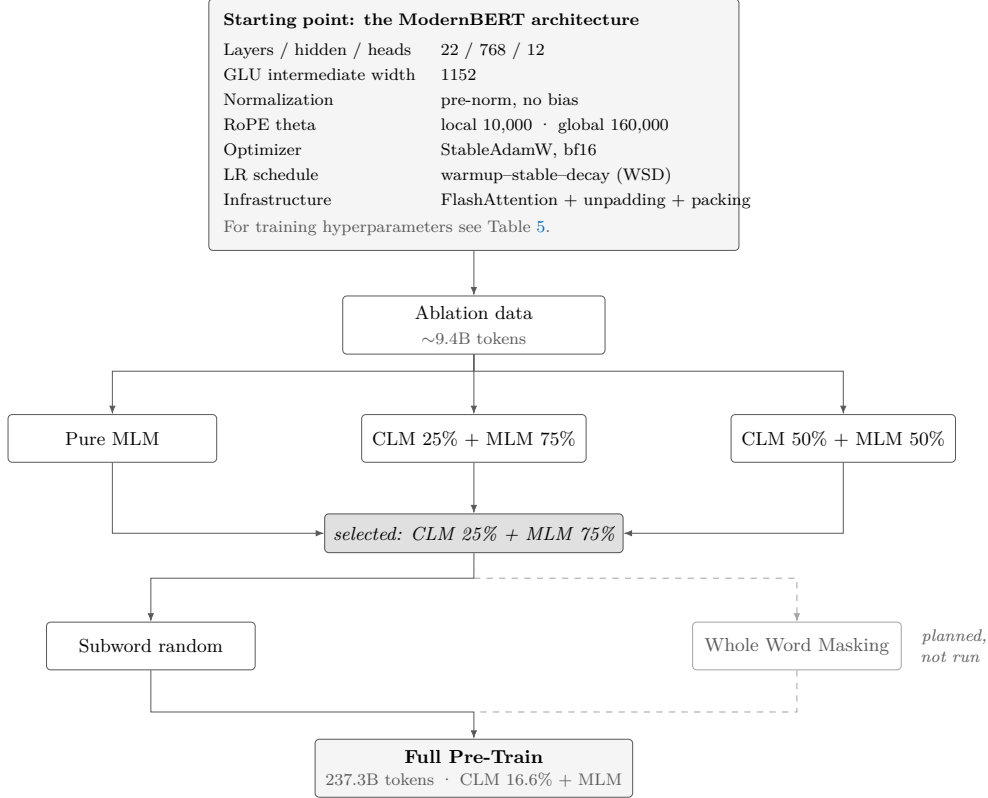


Figure 3: Decision flow for the CLM→MLM curriculum. The first ablation (§5.3) selects the training objective; the masking ablation was designed but not run on the basis of corpus statistics (dashed branch, §5.4). Training hyperparameters are given in Table 5.

5.3 Ablation: CLM→MLM vs. pure MLM

The claim reported at English scale by Gisserot-Boukhlef et al. [7] was tested on the MoganBert architecture, the MoganBert tokenizer and the Turkish corpus described above.

Setup. Both runs were trained with the same data pipeline, the same architecture and the same total number of steps (10,000): `mlm_only` runs 10,000 steps of pure MLM; `clm25_mlm75` uses CLM for the first 25% (~2,500 steps) and MLM for the remaining 75%. The ablations were run on a ~9.4-billion-token subset of the full corpus. The evaluation protocol is identical for both checkpoints (seed 1234, mask rate 0.30, `max_len` 512).

An important asymmetry. At equal total steps, `mlm_only` has seen 33% more MLM steps than `clm25_mlm75` (10,000 vs 7,500). The differences below are obtained in favour of the curriculum *despite* this handicap.

Table 4: Ablation results. The probes are linear classifiers trained on frozen representations; MS MARCO was run without contrastive training, on the raw pretraining representations. In the embedding geometry block, *lower* is better on the first two rows.

Metric	MLM-only	CLM25+MLM75	Δ
<i>Linear probe (accuracy %)</i>			
POS (BOUN), 15 classes	87.62	87.42	-0.20
News category, 5 classes	97.39	97.39	0.00
NLI, 3 classes	48.53	50.22	+1.69
Product review, 2 classes	80.75	81.80	+1.05
Average	78.57	79.21	+0.64
<i>Turkish MS MARCO (%), mean pooling</i>			
R@1	2.96	10.86	+7.90 ($\times 3.67$)
R@10	9.92	26.34	+16.42 ($\times 2.66$)
MRR	5.59	16.23	+10.64 ($\times 2.90$)
<i>Embedding geometry</i>			
Variance share of 1st component	0.2810	0.1187	-0.1623
cos_raw	0.9731	0.9595	-0.0136
STS Spearman	0.4833	0.5026	+0.0193

Finding. On the probe side the two models are practically *tied*; the +0.64-point difference in the average was measured with a single seed and cannot be separated from within-run noise. On POS the difference is 0.20 points, and on news category the two models give identical accuracy over 230 examples that task is saturated. Meaningful signal appears only on NLI and product review, both of which require sentence-level semantic representation. On the MS MARCO side, by contrast, the separation is qualitatively different: the curriculum model leads by 2.7–3.7 $\times$ on every metric, which cannot be explained by single-seed noise.

Mechanism: embedding geometry. That the probes are tied while retrieval differs threefold is no contradiction; the two metrics are sensitive to different things. A probe is a learned linear transformation and can rescale or ignore dominant directions that carry no information; cosine retrieval cannot, as it depends directly on the raw geometry. The critical row is the variance share of the first component: in the pure-MLM model a single direction absorbs 28.1% of the variance, against 11.9% in the curriculum model. Known in the literature as a “rogue dimension”, this pattern [25] compresses all vectors towards a common direction, and because cosine similarity cannot remove that common component, the discriminative signal is buried in noise. The consistent difference on STS which is likewise cosine-based is an independent confirmation of the same mechanism.

Decision. Full pretraining was carried out with the CLM $\rightarrow$ MLM curriculum: parity at worst on the probe metrics, against a clear and unidirectional advantage on all cosine-based tasks and with 33% fewer MLM steps at that.

5.4 Masking strategy

Which masking strategy to use in the MLM phase (subword random, whole word masking, or morphology-aware) was evaluated from corpus statistics without spending a training run. Three thousand documents per domain were sampled from five domains (~ 15.9 M words, 698,865 unique

types). The token-weighted rate of words split into ≥ 2 pieces is 41.1% and the type-weighted rate is 95.0%; the token shares of root and suffix candidates are 76.6% / 23.4%.

This gap is methodologically decisive: 95% of word types are multi-piece, but these are rare words; only 41% of the actual training token stream comes from multi-piece words. That number is directly the upper bound on WWM on 59% of the token stream, WWM and token-level random masking are already identical. The share that suffix-targeted masking could touch is 23.4%. **Decision:** token-level random masking was used in pretraining; a ceiling of 41% did not justify a separate ablation.

5.5 Full pretraining configuration

Table 5: Training configuration.

Item	Value
Parameter count	149.4M (22 layers, hidden 768, <code>local_attention=128</code>)
Vocabulary	50,048
Hardware	4× H100 (DDP)
Attention	FlashAttention-3 [22], varlen (<code>cu_seqlens</code>)
Precision	bf16
Optimizer	StableAdamW, betas (0.90, 0.98), eps 1e-6, weight decay 1e-5
Peak LR	8e-4 (WSD; warmup 1,431 steps, inside the CLM phase)
Global batch	2048 blocks × 1024 tokens = 2,097,152 tokens/step
Gradient accumulation	16
Gradient clipping	1.0
MFU	0.248
Throughput	836K tokens/second (4 GPUs in total)

5.6 Phase plan and branched annealing

The main run is a single stretch; the annealing phase is *split into two branches* after a shared prefix.

Table 6: Phase plan. 237.3B tokens in total for each branch.

Phase	Tokens	Steps	Ctx	Mask	LR
CLM	36.0B	1 – 17,167	1024	—	warmup + stable
MLM	180.7B	17,168 – 103,342	1024	30%	8e-4 stable
<i>Annealing — shared prefix</i>					
Anneal (shared)	10.5B	1 – 5,000	8192	10%	decay, upper half
<i>Annealing — two branches</i>					
Branch A: anneal	10.1B	5,001 – 9,810	8192	10%	decay, lower half
Branch B: anneal1k	10.1B	5,001 – 9,810	1024	10%	decay, lower half

Three things change simultaneously in the annealing phase. **(i)** Context is extended from 1024 to 8192; the global RoPE theta is scaled from 10,000 to 160,000, while the local window theta is left at 10,000 there is nothing to extend within a 129-token window. **(ii)** The mask rate is lowered from 30% to 10%; a lower mask rate produces a cleaner language modelling signal in this phase. **(iii)** The data mixture is shifted in favour of high-quality Turkish, because data shown in the low-LR region leaves a more lasting imprint on the model [13].

Rationale for branching. The alternative was to merge two different regimes with a *model soup* [30]. Branching is cheaper: the shared prefix is run once and only the final 4,810 steps are computed twice a total additional cost of $\sim 4.3\%$. A soup, by contrast, requires a separate full decay run for each regime. Both were produced and compared (Section 6.2); before the soup, the relative distance in weight space was measured ($|\text{main} - \text{anneal}|/|\text{main}| = 0.1011$ the same basin, so averaging is meaningful).

The LR decay curve ($1\text{-sqrt}, 8\text{e-}4 \rightarrow 8.36\text{e-}6$) is computed over the entire annealing run (9,810 steps) in both branches; branching changes only the *context length* of the final 4,810 steps. That is the only variable between the two branches.

5.7 Data mixture and held-out set

The mixture was applied not through probabilistic sampling during training but through a precomputed recipe file; this makes the ratios *exact* rather than statistical and makes resuming fully deterministic. The unique corpus comprises 107.96 billion tokens; the main run is two epochs over it.

Table 7: Realised group ratios in the main run. Within-step deviation is 0.055 points.

Group	Target	Realised	Δ
CommonCrawl	52.6%	52.60%	0.00
High-quality Turkish	20.7%	20.83%	+0.13
fineweb_edu_eng	13.9%	13.84%	-0.06
coding_gold	10.0%	9.97%	-0.03
eng_math_gold	2.8%	2.77%	-0.03

The language distribution is $\sim 73\%$ Turkish, $\sim 17\%$ English and $\sim 10\%$ code. The code share was deliberately kept high so that the tokenizer’s lossless-roundtrip advantage on code (Section 4.3) would be reflected in the model. For the two sources whose pool exceeds the amount to be used, a *different* slice was sampled in each epoch, so that twice as much unique data was seen at the same cost.

The evaluation set was separated from the training corpus before packing through a MinHash-based split: 6,202 documents, 136–200 per source. Candidates with a match in the training data were discarded.

The CLM ratio. The CLM share is 16.6% of the main run (36.0B / 216.7B), whereas the ratio validated in Section 5.3 was 25%. The reason for the deviation is a variable the ablation does not cover: the ablation runs were measured in a regime in which not even a single epoch was completed, whereas full pretraining was designed over two epochs. In a two-epoch design, the set seen during the CLM phase receives one MLM pass while the remainder receives two; the magnitude of this asymmetry has not been measured, and the ratio was deliberately kept low under that uncertainty. **This value has not been validated by ablation.**

6 Evaluation

6.1 Protocol

Evaluation uses two independent Turkish benchmarks. TrGLUE [2] is the primary benchmark: it is sentence-level, has a fixed five-seed protocol, and therefore supports the paired-seed comparisons on which the annealing analysis of Section 6.2 rests. TabiBench [26] is the secondary benchmark:

it is broader (28 datasets, eight categories, including retrieval and code) but single-seed, so it measures coverage rather than statistical separation.

TrGLUE. The TrGLUE evaluation was carried out entirely with the official `run_trglue.py` script; no metric was reimplemented. Hyperparameters were taken from the repository README: lr 3e-5 and batch 16 for `rte/stsb/mrpc`, lr 2e-5 and batch 128 for the remaining tasks; `max_seq_length` 128, 5 epochs, full fine-tuning. The seeds are the five official values given in the README (1, 4, 21, 40, 124), and every reported number is the average over those five. Since the repository provides no aggregation code reducing the eight tasks to a single score, the distinction is stated explicitly: for `mrpc`, `stsb` and `qqp` the script’s own `eval_combined_score` is used; `cola`, `sst2`, `qnli` and `rte` are reported with a single metric; for `mnli` the matched/mismatched average is taken. For all models, `classifier_pooling` was set to `mean`.

TabiBench. The evaluation follows the reference procedure of Türker et al. [26] without modification. Each of the 28 tasks is run in two stages. *Stage 1 — hyperparameter search:* 16 combinations over learning rate (5e-6, 1e-5, 2e-5, 3e-5), weight decay (1e-5, 1e-6) and batch size (16, 32) are tried on the *validation* split only, each for at most 10 epochs with early stopping; the winning configuration is selected on validation performance. *Stage 2 — test measurement:* a single run with the winning configuration is performed and only its *test* score is reported. The test split is not used at any point during hyperparameter selection. All 28 tasks go through this same procedure with identical early-stopping rules; there is no methodological difference between tasks. The learning-rate schedule is 6% warmup followed by linear decay to $0.02 \times$ the peak value, with mixed precision and seed 25.

Scores are aggregated as in the reference work: within a category, a *weighted average* proportional to test-set size; from categories to the overall score, a *macro average* with all eight categories weighted equally. The aggregation was verified against the reference paper’s own numbers: recomputing the weighted averages from the per-task scores and test sizes in its appendix reproduces the published *Weighted Avg* column on 40 of 40 rows to within 0.007 points, and the macro average of the categories reproduces the published *Total Avg* exactly (TabiBERT: $620.65/8 = 77.58$). Test set sizes were taken from the same appendix and agree with the Hugging Face dataset cards on 28 of 28 tasks.

Calibration check. The correctness of the protocol was tested on two independent models with published numbers: BERTurk 79.87 (published 80.07, -0.20) and ModernBERT-TR 77.64 (published 78.23, -0.59). That BERTurk holds within ± 0.2 indicates the protocol is sound; the deviation on ModernBERT-TR is most likely due to the published number having been obtained under that work’s own settings rather than the TrGLUE protocol.

6.2 Branched annealing: two branches and a model soup

The three candidates produced in Section 5.6 decay at 8192 (`anneal`), decay at 1024 (`anneal1k`), and the weight average of the main run and anneal (`soup`) were compared under the same protocol.

The effect of annealing itself. Held-out MLM loss (block 8192, 31 blocks, 254K tokens; mask rate fixed at 30% for measurement) falls monotonically throughout annealing: starting from the anchor (the final checkpoint of the main run) at 1.8026, it descends to 1.7072, 1.6801, 1.6607, 1.6474 and 1.6379 (-0.165). There is no plateau. The critical finding is that even though

training was carried out with a 10% mask rate, improvement is continuous on the harder 30% task as well lowering the mask rate did not damage performance on the harder task.

Fill-mask probe. Over 50 sentences and 39 single-token gold labels, the soup beats both of its parents (top1 71.8% vs 69.2% / 64.1%; MRR 0.787; mean $\log P$ -1.27) the classic soup result. **This gain, however, did not carry over downstream.**

Table 8: TrGLUE average of the three candidates (five official seeds).

Candidate	TrGLUE AVG	Rank (6 models)
anneal1k (decay at 1024)	78.41 $\pm$ 0.32	2
anneal (decay at 8192)	77.92 $\pm$ 0.48	3
soup	77.66 $\pm$ 0.37	5

Paired comparison. When the two branches are compared on the same seeds, seed-induced noise drops: the differences are $+0.79$, $+0.36$, $+0.65$, $+0.13$ and $+0.52$ respectively, averaging $+0.49 \pm 0.26$. Here $t = 4.28$ (df= 4), $p \approx 0.013$, and **all five of the five seeds are positive**. Running the final portion of decay at 1024 context yielded a statistically significant improvement.

Conclusion: **anneal1k** was selected as the final model. The branching approach is both cheaper ($\sim 4.3\%$ additional cost) and more successful ($+0.75$ TrGLUE points) than the soup. The soup case also concretely demonstrates the risk of selecting a model on the basis of a small-sample internal metric.

6.3 TrGLUE

Table 9: TrGLUE eight tasks, average $\pm$ standard deviation over the five official seeds. Bold: best in row.

Task	BERTurk	MBERT-TR	TabiBERT	Mogan-anneal	Mogan-anneal1k
cola	41.71 ± 1.87	26.62 ± 2.36	31.89 ± 1.76	35.92 ± 2.29	37.63 ± 1.97
sst2	87.62 ± 0.19	86.47 ± 0.44	85.72 ± 0.27	85.60 ± 0.13	85.83 ± 0.11
mrpc	71.00 ± 0.94	70.45 ± 0.63	68.91 ± 0.77	68.39 ± 1.55	69.18 ± 0.56
stsb	72.08 ± 0.60	70.51 ± 0.98	69.18 ± 5.38	68.96 ± 1.28	69.59 ± 1.42
qqp	95.04 ± 0.07	95.17 ± 0.03	95.08 ± 0.08	94.46 ± 0.06	94.54 ± 0.08
mnli	89.11 ± 0.11	88.13 ± 0.30	88.00 ± 0.21	88.10 ± 0.14	88.24 ± 0.19
qnli	90.48 ± 0.17	89.94 ± 0.18	89.89 ± 0.13	89.64 ± 0.17	89.87 ± 0.16
rte	91.94 ± 0.50	93.84 ± 0.57	93.94 ± 0.34	92.30 ± 0.75	92.42 ± 0.40
AVG	79.87 ± 0.23	77.64 ± 0.37	77.83 ± 0.57	77.92 ± 0.48	78.41 ± 0.32

MoganBert-TR (**anneal1k**) is the best of the Turkish ModernBERT models compared (78.41 vs TabiBERT 77.83, ModernBERT-TR 77.64). BERTurk leads by 1.46 points, and the source of that difference is concentrated in two tasks: CoLA (41.71 vs 37.63) and STS-B (72.08 vs 69.59) both known weak spots of an NSP-free architecture (Section 6.5).

Notes on variance. The reliable tasks are qqp (± 0.07), mnli (± 0.19), qnli (± 0.17) and sst2 (± 0.19); there, even 0.1-point differences are meaningful. TabiBERT’s ± 5.38 on STS-B is a genuine instability persisting across all five seeds almost the entirety of that model’s total deviation originates here.

6.4 TabiBench

TabiBench [26] spans 28 datasets in eight categories and reaches beyond sentence-level understanding into retrieval, code and academic text. It is therefore complementary to TrGLUE rather than a substitute: it covers more capabilities, but reports single-seed scores.

Table 10: TabiBench category scores. Metrics by category: macro-F1 (text classification, NLI, academic), micro-F1 at word level (token classification), Pearson (STS), F1 (QA), NDCG@10 (both retrieval categories). Bold: best in column among monolingual Turkish models. Reference scores for the four models below the rule are taken from Türker et al. [26], Table 4; the ModernBERT-TR row is read from the figures of its release article [1], which does not publish a numeric table. mmBERT is a 307M-parameter multilingual model trained on 3T tokens and is listed as a reference point, not as a comparable system.

Model	Text	Token	STS	NLI	QA	Acad.	IR	Code	Total
MoganBert-TR	83.71	91.20	85.43	84.14	68.37	72.62	75.78	60.57	77.73
ModernBERT-TR	85.21	90.78	86.08	84.74	67.03	71.83	77.51	60.16	77.92
TabiBERT	83.44	93.42	84.74	84.51	69.71	72.44	75.44	56.95	77.58
BERTurk	83.42	93.67	85.33	84.33	60.16	71.40	74.84	54.54	75.96
YTU-Cosmos-BERT [10]	84.25	93.60	84.68	84.16	31.50	71.78	74.29	53.80	72.26
TurkishBERTweet [16]	79.71	92.02	75.86	79.10	38.13	63.12	68.40	43.49	67.48
<i>Multilingual reference (not ranked)</i>									
mmBERT (307M)	82.54	93.81	87.05	84.38	71.47	72.65	76.20	66.02	79.26

With 77.73, MoganBert-TR places second among monolingual Turkish encoders, 0.19 points behind ModernBERT-TR (77.92) and 0.15 points ahead of TabiBERT (77.58). None of these three gaps should be read as an ordering. The benchmark’s authors state the constraint themselves: their scores come from single runs without significance testing, margins below one point “may or may not reach statistical significance”, and they present the results as indicative rather than as confirmed superiority [26]. The MoganBert-TR scores are likewise single-run, so the same caveat applies to them. The three models are best described as tied at the top of the benchmark; what is informative is the category structure underneath, where some differences are large enough to survive that caveat.

Where MoganBert-TR leads. The model ranks first among Turkish models in two categories. The one that matters is **code retrieval** (60.57). Against TabiBERT the margin is +3.62 and against BERTurk +6.03 far outside any plausible single-run noise and it is the downstream counterpart of two upstream decisions: the indentation-preserving tokenizer (Section 4.3) and the deliberately high 10% code share of the pretraining mixture. The per-task pattern is consistent with this: the gain over TabiBERT concentrates on Apps-TR (+4.90), StackoverflowQA-TR (+4.05) and CodeSearchNet-21K-TR (+2.26), all of which require matching Turkish natural language against literal source code. Against ModernBERT-TR, however, the margin is only 0.41, and that model also allocates roughly a tenth of its mixture to code; this suggests that the mixture share, rather than anything specific to MoganBert-TR, is what separates both models from TabiBERT (Section 8). The second lead, **academic understanding** (72.62), is narrow (+0.18 over TabiBERT, +0.79 over ModernBERT-TR) and driven by ThesisAbstract (+2.67), a 187-class problem over thesis abstracts.

Where MoganBert-TR trails. **Token classification** is the clearest weakness: 91.20 against 93.42 for TabiBERT and 93.67 for BERTurk, a deficit of 2.2–2.5 points that is consistent across all four datasets (WikiANN-TR −2.63, WikiNER −1.34, PosUD-IMST −1.10, PosUD-BOUN

−0.07). The uniformity of the sign across four datasets makes a measurement artefact unlikely, but the deficit cannot be attributed: ModernBERT-TR scores 90.78 in the same category with a different tokenizer, so a tokenizer-based explanation does not follow from the available evidence, and the cause is left open. **Question answering** (68.37) trails TabiBERT by 1.34 points, entirely on TQuAD (−2.35); on XQuAD MoganBert-TR leads by 12.99 points, but with 179 test examples that figure should not be over-read. In **STS**, **NLI** and **information retrieval** ModernBERT-TR is ahead by 0.65, 0.60 and 1.73 points respectively.

Scope of the comparison. The full 28-task protocol was run for MoganBert-TR only. All reference scores are taken from the publications of the models concerned rather than re-measured, because the hyperparameter search alone 16 configurations per task across 28 tasks was beyond the compute budget available to this project for six models. Consequently the MoganBert-TR numbers and the reference numbers were produced in different environments, and although the protocol is identical, environment-dependent variation cannot be excluded. This is a further reason to read the small margins at the top of the table as a tie.

Per-task scores are given in Appendix D.

6.5 CoLA: an architectural limit

CoLA is the single largest item in the gap between MoganBert-TR and BERTurk. All three ModernBERT models fall 4–15 points below BERTurk (BERTurk 41.71; MoganBert-TR 37.63; TabiBERT 31.89; ModernBERT-TR 26.62), and MoganBert-TR is the best among them.

The mechanism is architectural: in MLM, [CLS] is never masked and therefore receives no direct gradient. In BERT the NSP loss is computed over [CLS], so that [CLS] is genuinely pretrained; ModernBERT removed NSP which is why all three Turkish ModernBERT models declare `classifier_pooling: mean`.

Seven configurations were tried (lr 2e-5/3e-5/5e-5/1e-4/1e-3, batch 32/128, cls/mean pooling), and the same pattern emerged along three independent axes: *the more intensive the training, the worse the result*. The lowest LR and the shortest training give the best result; at lr 1e-3 the model collapses entirely (−0.25). Opening up 149M parameters to 7,916 examples means overfitting, and CoLA is the task most sensitive to it. The cause of the collapse is a scale mismatch: pretraining used 8e-4 but with a global batch of 2,097,152 tokens, whereas CoLA fine-tuning uses a batch of 16,384 tokens (128× smaller) with no warmup.

6.6 Raw embedding geometry

The raw representations of the pretrained model are not suitable for direct use with cosine similarity. Anisotropy measured with mean pooling over the same 400 sentences:

Table 11: Anisotropy comparison. `cos_raw` is the mean cosine similarity of random sentence pairs; proximity to 1 means all vectors are collected within a narrow cone.

Model	cos_raw	eff_rank	top1_var
MoganBert-TR	0.9841	117.9	0.0824
TabiBERT	0.9405	125.1	0.0682
ModernBERT-TR	0.9234	125.5	0.0924
BERTurk	0.7848	126.8	0.0632

Anisotropy is present in all MLM encoders (0.92–0.98), but the highest value is MoganBert-TR’s; BERTurk is markedly better, and this is plausibly a side effect of NSP (an unverified hypothesis). The direct consequence is that zero-shot retrieval performance remains low: on a

ten-set IR evaluation the raw model scores $\text{NDCG@10} = 0.2361$. This measurement is the point of departure for the embedding model work in Section 7.

6.7 RoPE theta ablation

Annealing raises the global theta to 160,000. Since TrGLUE uses `max_seq_len=128`, whether this harms short sequences was tested directly; the weights are identical and the only variable is theta in the configuration.

Table 12: Effect of changing RoPE theta at inference time.

Task	Metric	theta 160k	theta 10k	Difference
stsb	pearson	0.6933	0.6893	−0.0040
stsb	spearman	0.6710	0.6698	−0.0012
mrpc	accuracy	0.7150	0.7060	−0.0090
mrpc	f1	0.6570	0.6397	−0.0173

All four measurements are unfavourable to theta 10k. Since the model was trained with 160k, pulling it back to 10k at inference creates a frequency-base mismatch; the theoretical disadvantage of a large theta on short sequences remains small next to that mismatch. **Decision:** `global_rope_theta=160000`; the 8192-context capability is retained at no cost.

7 Embedding Model

The anisotropy measured in Section 6.6 (`cos_raw` = 0.9841) showed that the pretrained model cannot be used for retrieval in its raw form. This section presents the two-phase route followed in order to produce MoganBert-Embed, an embedding model separate from the encoder. MTEB(Turkish) [15] was used as the measurement ground throughout (26 tasks, `max_len` 2048, fp16).

7.1 Phase 1 — Teacher distillation

Qwen3-Embedding-8B [20] was used as the teacher (output dimension 3072). The corpus was given to the teacher without a task instruction. The loss has two terms:

$$L = 1.0 \cdot L_{\text{distill}} + 0.05 \cdot L_{\text{GOR}}$$

The distillation is cosine-based and the student is projected *up* from 768 to 3072 the teacher is not truncated; the student rises into its space. GOR (global orthogonal regularization [31]), meanwhile, is computed at the 768-dimensional backbone output where the anisotropy was measured. In feature distillation it is critical that the student sees the *same input format* as the teacher; if the formats diverge, the learned mapping shifts.

Table 13: Effect of Phase 1 on geometry.

Metric	Start (MLM)	After Phase 1
<code>cos_raw</code> (anisotropy)	0.9841	0.0851
<code>eff_rank</code>	117.9	157.2
<code>top1_var</code>	0.0824	0.0355
Zero-shot IR (10 sets, NDCG@10)	0.2361	0.5927

The fall of `cos_raw` from 0.98 to 0.085 means the representations have left the narrow cone

and spread out into the space; retrieval rose from 0.2361 to 0.5927 at the same step.

7.2 Phase 2 — Contrastive fine-tuning

What differs between the versions of Phase 2 is not the model architecture but which signal types enter the mixture and how each type is processed.

v1 — a single signal type. Retrieval pairs only:

$$L = 1.0 \cdot L_{\text{NCE}} + 2.0 \cdot L_{\text{distill}} + 0.1 \cdot L_{\text{GOR}}$$

The batch is split: the task half produces L_{NCE} (with hard negatives, $\tau=0.02$) and the corpus half produces L_{distill} (the Phase 1 anchor). L_{GOR} is computed *separately* over the query and document sets had it been computed jointly, it would push a query away from its own positive document and work directly against L_{NCE} . Result: IR 0.5927 $\rightarrow$ 0.6786.

v2 — negative design and two structural fixes. v2 takes the mixture beyond retrieval pairs. The decisive criterion is not the source itself but whether an *explicit* negative can be defined from that source. A **mined** negative is selected from the pool by similarity: it is topically close, and therefore hard, but carries no label guarantee. An **explicit** negative comes from the structure of the data itself and its opposition is guaranteed: the contradiction hypothesis of a premise, a low-scoring sentence pair, or a sentiment sentence expressing the same topic with the opposite polarity. When an explicit negative is available it is placed in slot zero *in place of* the mined one.

Two structural fixes were added. *False-negative masking*: since a query can have dozens of relevant documents, the other positives of that query within a batch were marked as negatives, and InfoNCE [27] was teaching the model that “this relevant document is irrelevant”; all positives of the same query were grouped and masked in the logits. *Anchor decay*: a fixed distillation weight of 2.0 tied the model to the Phase 1 geometry (the validation cosine fell from 0.8896 to only 0.883 over 8,000 steps); the weight was decayed to 0.25 but not reduced to zero the anchor is the only signal holding up quality on general Turkish text not represented in the training mixture.

v3 each signal type received its own loss. In v2 all types were fed into the same InfoNCE. In v3 three types are processed in accordance with their own nature. **(i)** CoSENT [23] for STS: by thresholding, v2 discarded the middle band ($\sim 40\%$ of the data) and ignored the ranking information, whereas the STS metric is Spearman.

$$L = \log \left(1 + \sum_{s_i > s_j} e^{(\cos_j - \cos_i)/\tau} \right)$$

(ii) In NLI, contradiction was positioned as a *hard* negative and neutral as a *soft* negative, and (hypothesis, premise) was added alongside (premise, hypothesis) pair classification is a symmetric task. **(iii)** A same-class mask was added for the label data: because the batch is task-homogeneous, one third of the in-batch negatives in a three-class set come from the same class, and at every step the model was being taught that “these two positive reviews are not similar to each other”.

In addition, a set of scientific abstracts whose labels carry rhetorical roles was removed from the mixture: two sentences with the same role are *not* semantically similar, and counting them as positives pushes the space towards clustering by discourse function. The tendency of

temperature-based sampling to repeat small sources 30–38 times was also capped (at most 12 times the number of pairs per source).

v4 and v5. Two further runs were produced from the same recipe family; they propose no new method, and their role is to serve as soup components. v4 is almost equal to v3 in overall score (66.88 vs 66.97) but has a different profile: it leads on retrieval and trails on pair classification, STS and classification. Because components that remain close in weight space while their error patterns diverge correct one another in the average, both were taken into the soup. v5 was not evaluated individually and was not used, as it lowered the result in the soup (67.85 vs 68.30).

Table 14: Phase 2 version results, MTEB(Turkish). v5 was not evaluated individually.

Version	OVERALL	Retr.	Pair	STS	Class.	Clust.	Bitext
phase1	63.53	56.13	56.76	73.95	70.61	64.35	96.55
phase2_v2	64.56	57.95	57.56	70.59	71.34	64.18	98.72
phase2_v4	66.88	58.22	70.89	80.02	73.17	62.42	95.68
phase2_v3	66.97	57.77	71.39	80.94	73.78	62.32	95.66

Two of v3’s three targeted changes clearly worked: pair classification $57.56 \rightarrow 71.39$ (+13.83) and STS $70.59 \rightarrow 80.94$ (+10.35). Classification, at $71.34 \rightarrow 73.78$ (+2.44), fell short of the target. Two areas regressed: clustering $64.18 \rightarrow 62.32$ and bitext $98.72 \rightarrow 95.66$. The soup phase exists precisely to close these two gaps.

7.3 Sentiment-contrast data

The most targeted signal of Phase 2 is a set of sentiment-contrast triples produced from scratch (40,737 triples). In each record the **positive** field expresses the same sentiment class in different words; in the **negative** field the sentiment class has been changed while topic, subject, context and length are held constant and as few words as possible are altered not an unrelated sentence but a *hard* counterexample. Neutral was defined as a genuine class: an opinion is present but measured, mixed or undecided. If there is *no* opinion at all the record is discarded entirely; texts such as news reports, definitions, recipes, announcements, legislation and technical descriptions are dropped. The data is LLM-generated and has not been human-validated (Section 9).

7.4 Model soup

A weighted average of the components was taken [30]. Because checkpoints descending from the same starting point (Phase 1) remain close to one another in weight space, averaging is meaningful.

Table 15: Selected soups. “Phase 1 share” = the ratio of the Phase 1 weight to the total weight.

Soup	Components	Weights	Phase 1 share	OVERALL
f34-heavy	phase1+v3+v4	0.4 / 1 / 1	0.167	68.30
f34-06	phase1+v3+v4	0.6 / 1 / 1	0.231	68.22
f-v3	phase1+v3	0.5 / 1	0.333	68.22
f34-05	phase1+v3+v4	0.5 / 1 / 1	0.200	68.19
345 (no Phase 1)	v3+v4+v5	equal	0.000	67.08
fisher-f34	phase1+v3+v4	0.4 / 1 / 1	—	67.19

Three findings stand out. **(1) Phase 1 must be in the mixture.** The soup consisting

only of Phase 2 versions gives the weakest result at 67.08; adding Phase 1 raises it to 68.30. Phase 1 was individually the lowest-scoring model (63.53), yet it brings clustering and bitext to the mixture (clustering 62.19 $\rightarrow$ 66.46, bitext 96.48 $\rightarrow$ 97.24) a direct compensation for the two regressions of Phase 2. **(2) The sweet spot is ~ 0.17** and the curve is flat between 0.10 and 0.25 (67.08 / 68.07 / **68.30** / 68.19 / 68.22 / 68.22); what matters is that Phase 1 is *present*, not its exact ratio. **(3) Not every component that adds diversity helps:** adding v2 drops the result to 68.10 and adding v5 to 67.85. Fisher-weighted merging [14] also fell behind the plain average in every pairing (67.19 vs 68.30); since the Fisher information matrix is estimated from a sample of 1,500 sentences, it is plausible that estimation noise at that size undermines the regularization the plain average provides.

7.5 MTEB(Turkish) results

The final soup was measured on the 26 tasks of MTEB(Turkish) under the same conditions as five external references and the teacher model: fp16, cosine similarity, at most 2048 tokens. The pooling and prompt convention of each model was read from its own configuration. Qwen3-Embedding-8B is not a competitor in this work but the teacher of the Phase 1 distillation; its score should be read as a *reference ceiling*.

Table 16: MTEB(Turkish) category averages. “Ctx” = the model’s position limit. Bold: best in column among student models. The last row is the teacher model and is not included in the ranking.

Model	Param	Ctx	OVERALL	Retr.	Pair	STS	Class.	Clust.	Bitext
MoganBert-Embed*	149M	8192	68.30	59.58	69.32	79.49	75.36	66.46	97.24
ModernBERT-TR	149M	8192	68.13	59.41	69.21	77.61	76.51	63.20	94.04
mE5-large-instruct	560M	512 [†]	67.47	61.05	63.03	81.23	73.70	61.87	98.99
BGE-M3	568M	8192	67.36	60.26	68.72	79.60	72.79	60.80	98.99
Mursit-Large-TR	404M	2048	62.00	55.70	55.19	74.60	68.66	61.53	86.72
trmteb-tr-embed	111M	512 [†]	59.71	52.54	56.81	74.96	70.62	63.17	37.75 [‡]
<i>Teacher model (reference ceiling, not included in the ranking)</i>									
Qwen3-Embedding-8B	7.57B	40960	68.66	64.46	62.76	80.04	72.95	63.01	98.18

* Weighted soup: Phase 1 + v3 + v4, ratio 0.4/1/1. [†] The position table of these two models is limited to 512; since they could not be measured at 2048, they were evaluated at 512. [‡] **trmteb-tr-embed** is monolingual; because WMT16 requires cross-lingual matching, this task falls outside its scope.

Four observations stand out.

1. The student reaches 99.5% of its teacher. The 149M-parameter model trails its 7.57-billion-parameter teacher by 0.36 points with a $51\times$ smaller backbone. The teacher’s advantage is concentrated in retrieval (64.46 vs 59.58); in return, the student surpasses its teacher on pair classification (69.32 vs 62.76), classification (75.36 vs 72.95) and clustering (66.46 vs 63.01). Distillation evidently does not transfer all capabilities in equal measure and leaves a gap on capacity-sensitive tasks such as retrieval.

2. MoganBert-Embed ranks first on the overall average among student models (68.30 vs ModernBERT-TR 68.13). The wins are scattered: across 26 tasks the soup ranks first on 9, mE5 on 8, ModernBERT-TR on 4, BGE-M3 on 3 and Mursit on 2.

3. The difference is clearest on clustering. MoganBert-Embed ranks first on both clustering tasks and leads the runner-up by 3.26 points on the category average; this is a direct consequence of the Phase 1 contribution discussed in Section 7.4.

4. Retrieval is the relative weak spot. The model ranks first on only three of the 11 retrieval tasks and trails mE5 and BGE-M3 on the category average. The share of the retrieval

signal in the Phase 2 mixture had been reduced in favour of other signal types with v3; this result is the price of that choice.

The full task-level table is given in Appendix E.

8 Discussion

The training objective is a lever independent of the architecture. The Turkish encoder literature has so far focused on updating the architecture; the measurements reported here show that, under the same architecture and the same compute budget, the *objective* alone can produce a qualitative difference. That difference, moreover, is not uniform: it is invisible on classification probes and grows threefold on cosine-based tasks. This is a concrete warning about what is missed when encoder comparisons rest on GLUE-style benchmarks alone.

The annealing phase is a cheap design space. Branching produces two different regimes for only $\sim 4.3\%$ of the total cost and, in these measurements, gives a better result than a model soup. The role of context length during decay turned out to be as decisive as that of the data mixture; this extends mmBERT’s [13] decay-phase finding along a complementary axis.

The code-retrieval gap tracks the mixture, not the objective. Code retrieval is the category in which MoganBert-TR separates most clearly from TabiBERT (+3.62) and BERTurk (+6.03), and the natural question is which design decision produced it. Two candidates are the indentation-preserving tokenizer, which reaches full lossless roundtrip on the code corpus against 49% for TabiBERT (Section 4.3), and the code share of the pretraining mixture, 10% against TabiBERT’s 6%. A third possibility that initially seems plausible is the CLM phase: source code is more strictly sequential than natural language, so a left-to-right objective might capture its structure better than masked prediction over a 30%-corrupted sequence.

The comparison with ModernBERT-TR argues against that third explanation. That model is trained with masked language modelling only, has no CLM phase, uses a different tokenizer, and allocates roughly a tenth of its mixture to code [1] and it scores 60.16, within half a point of MoganBert-TR’s 60.57, while TabiBERT with a 6% code share sits nearly four points below both. The variable that co-varies with the outcome is the mixture share, not the objective and not the tokenizer family. No code-specific benefit is therefore claimed for the curriculum; the retrieval gain measured for it (Section 5.3) was observed on natural-language retrieval, and nothing in these experiments extends it to code. Two models are a small sample and they differ in many other respects, so this is an observation rather than a controlled result; isolating the factors would require training the same mixture under two tokenizers and the same tokenizer under two objectives.

Small-sample internal metrics are misleading. Although the model soup beat both of its parents on the fill-mask probe, it ranked fifth of six models on TrGLUE. Basing checkpoint selection on small probes is common practice; this case demonstrates its cost.

There is an architectural ceiling. That BERTurk leads on CoLA and STS-B is no accident: removing the NSP loss leaves the [CLS] representation unpretrained. That all three Turkish ModernBERT models fall behind in the same direction supports this reading. A lightweight auxiliary objective providing a direct gradient to [CLS] during pretraining is an unmeasured but promising direction.

9 Limitations

- **The objective ablation is single-seeded** and rests on short runs of 10,000 steps; whether the advantage is preserved at full-pretraining scale must be verified separately. The retrieval leg rests on a single dataset, and two of the probe tasks are saturated.
- **The 16.6% CLM ratio used in full pretraining has not been validated by ablation;** the ablation result holds for 25%. The “pure 1024 annealing” branch has likewise not been measured.
- **The soup components are single-seeded**, and because the soup weights were swept on MTEB they carry some degree of overfitting; the flatness of the curve between 0.10 and 0.25 limits this risk but does not remove it.
- **TabiBench is single-seed, and only MoganBert-TR was run in this work.** Both the MoganBert-TR scores and the reference scores come from one run per task, and the reference scores were produced in a different environment; the benchmark’s authors already caution that margins below one point may not be significant [26]. The 0.19-point gap to ModernBERT-TR and the 0.15-point gap to TabiBERT are therefore not an ordering. Re-measuring all six models in one environment would settle this, but the hyperparameter search it requires was outside the available compute budget.
- **Two reference models were measured at 512 tokens** and the others at 2048; on retrieval tasks involving long documents this works against those two models, and their scores should be read as lower bounds.
- **A single teacher was used**, and the sentiment data is LLM-generated and has not been human-validated.
- **Because of licensing restrictions** the corpus cannot be released in full; the filtering decisions and the pipeline code are, however, shared for reproducibility.

10 Conclusion

This work shows that an encoder foundation model trained from scratch for Turkish from data collection through to an embedding model can be produced by a small team without institutional infrastructure. The main finding is that the choice of objective in encoder pretraining is decisive for Turkish as well: under an equal step budget, the CLM→MLM curriculum outperformed pure MLM by $2.7\text{--}3.7\times$ on cosine-based tasks, and its mechanism was measured in embedding geometry. The second finding is that the annealing phase is a cheap but effective design space: branching the decay after a shared prefix gave a result 0.75 TrGLUE points better than a model soup at $\sim 4.3\%$ additional cost.

MoganBert-TR is the best of the Turkish ModernBERT models compared, with 78.41 on TrGLUE, and reaches 77.73 on TabiBench, statistically indistinguishable from ModernBERT-TR (77.92) and TabiBERT (77.58) at the top of that benchmark; its clearest category-level advantage is code retrieval, where the mixture and tokenizer decisions of Sections 4.3 and 5.6 translate into a 3.62-point margin over TabiBERT. MoganBert-Embed, the embedding model derived from it, ranks first among student models on MTEB(Turkish) with 68.30.

Future work. There are three priority directions. The *retrieval gap* can be closed by increasing the share of the retrieval signal in the Phase 2 mixture or by producing a separate retrieval-focused soup component. Along the *scale* axis, a large variant ($\sim 400\text{M}$) and a decoder built on the same data pipeline and tokenizer would complete the family. As for *ablation gaps*, the CLM ratio used

in full pretraining and the pure 1024 annealing branch should be tested in a controlled manner, and the code-retrieval advantage should be decomposed into its tokenizer, mixture and objective components (Section 8).

Availability

Model weights, the tokenizer, the embedding model and the evaluation code will be released openly at <https://huggingface.co/moganai> upon publication of this preprint. The data pipeline code will be shared together with the filtering decisions so that the corpus can be reproduced; because of licensing restrictions on part of the corpus, the corpus itself cannot be released in full (Section 9).

References

- [1] Beshir Alkurdi, Himmet Toprak Kesgin, Muzaffer Kaan Yuce, and Mehmet Fatih Amasyali. ModernBERT-TR: A modern encoder foundation model for Turkish. *Research Square preprint*, 2026. doi: 10.21203/rs.3.rs-9500377/v1.
- [2] Duygu Altinok. Introducing TrGLUE and SentiTurca: A comprehensive benchmark for Turkish general language understanding and sentiment analysis. *arXiv preprint arXiv:2512.22100*, 2025. doi: 10.48550/arXiv.2512.22100.
- [3] M. Ali Bayram, Ali Arda Fincan, Ahmet Semih Gümüş, Sercan Karakaş, Banu Diri, and Savaş Yıldırım. Tokenization standards for linguistic integrity: Turkish as a benchmark. *arXiv preprint arXiv:2502.07057*, 2025. doi: 10.48550/arXiv.2502.07057.
- [4] Andrei Z. Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences*, 1997.
- [5] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*, 2020.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.
- [7] Hippolyte Gisserot-Boukhlef, Nicolas Boizard, Manuel Faysse, Duarte M. Alves, Emmanuel Malherbe, André F. T. Martins, Céline Hudelot, and Pierre Colombo. Should we still pretrain encoders with masked language modeling? *arXiv preprint arXiv:2507.00994*, 2025. doi: 10.48550/arXiv.2507.00994.
- [8] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In *Advances in Neural Information Processing Systems*, 2024.
- [9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [10] Himmet Toprak Kesgin, Muzaffer Kaan Yuce, and Mehmet Fatih Amasyali. Developing and evaluating tiny to medium-sized Turkish BERT models. *arXiv preprint arXiv:2307.14134*, 2023. doi: 10.48550/arXiv.2307.14134.

- [11] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP: System Demonstrations*, 2018.
- [12] Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. FineWeb-Edu: The finest collection of educational content. Hugging Face, 2024.
- [13] Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. mmBERT: A modern multilingual encoder with annealed language learning. *arXiv preprint arXiv:2509.06888*, 2025.
- [14] Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, 2022.
- [15] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of EACL*, 2023.
- [16] Ali Najafi and Onur Varol. TurkishBERTweet: Fast and reliable large language model for social media analysis. *arXiv preprint arXiv:2311.18063*, 2023. doi: 10.48550/arXiv.2311.18063.
- [17] NLLB Team. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.
- [18] Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2024.
- [19] Guilherme Penedo, Hynek Kydlíček, Vinko Šabolcec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. FineWeb2: One pipeline to scale them all adapting pre-training data processing to every language. *arXiv preprint arXiv:2506.20920*, 2025.
- [20] Qwen Team. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.
- [21] Stefan Schweter. BERTurk BERT models for Turkish. Zenodo, doi:10.5281/zenodo.3770924, 2020.
- [22] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In *Advances in Neural Information Processing Systems*, 2024.
- [23] Jianlin Su. CoSENT (i): A more effective sentence embedding scheme than Sentence-BERT. Scientific Spaces (blog), <https://kexue.fm/archives/8847>, 2022.
- [24] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [25] William Timkey and Marten van Schijndel. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of EMNLP*, 2021.

- [26] Melikşah Türker, Asude Ebrar Kızıloğlu, Onur Güngör, and Susan Üsküdarlı. TabiBERT: A large-scale ModernBERT foundation model and a unified benchmark for Turkish. *arXiv preprint arXiv:2512.23065*, 2025.
- [27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [28] Benjamin Warner, Antoine Chaffin, Benoît Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*, 2024.
- [29] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC)*, 2020.
- [30] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of ICML*, 2022.
- [31] Xu Zhang, Felix X. Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proceedings of ICCV*, 2017.

A Data Pipeline Details

Access and indexing. Unauthenticated access to Turkish content in Common Crawl is restricted at the index level; in practice the data can only be downloaded via AWS S3 (`s3://commoncrawl, us-east-1`). For each monthly crawl, Turkish content is first indexed from that month’s columnar index (Parquet) according to the `content_languages` field; this step extracts the triple `warc_filename`, `warc_record_offset` and `warc_record_length`. The download rate at production scale is $\sim 450\text{--}480$ records/second (64 concurrent workers, `c7i.8xlarge`).

Filter chain. WARC parsing, boilerplate removal, Gopher/C4 quality filters, Turkish boilerplate check, language verification, fastText quality classification and PII masking (Turkish national ID checksum verification, telephone, e-mail, IBAN, and the Luhn algorithm for credit cards).

Quality classifier details. The LLM rubric was calibrated so that gambling/casino/adult content would receive a fixed 0 and advertising/real-estate-listing/SEO content would stay below 3. The threshold choice was corrected empirically: when tested with `keep_labels={3,4,5}`, the discarded `label=2` documents turned out in large majority to be genuine news, stock-market, recipe and article content (words such as “bet” or “fiyat” were mostly false positives: “İstanbulspor”, “Betondan”), and `label=2` was therefore added to the accepted set. The distilled fastText was trained with `lr=1.0`, `epoch=25`, `wordNgrams=2`, `dim=100`, `loss=softmax`.

Resource and worker optimization. The fastText quality model (~ 2.29 GB) is loaded separately in each parallel worker. On `c7i.8xlarge` (32 vCPU, 61 GB RAM), OOM was observed at 30 and 24 workers; at 16 workers memory use remained stable ($\sim 23\text{--}30$ GB).

Ablation subset. CommonCrawl 3.63B, Code 1.70B, FineWeb-edu-eng 1.25B, Legal 780M, Articles 627M, FinePdfs 588M, Books 316M, FineMath-4Plus 303M, turkce-matematik 221M ~ 9.4 B tokens in total.

B Tokenizer Implementation Details

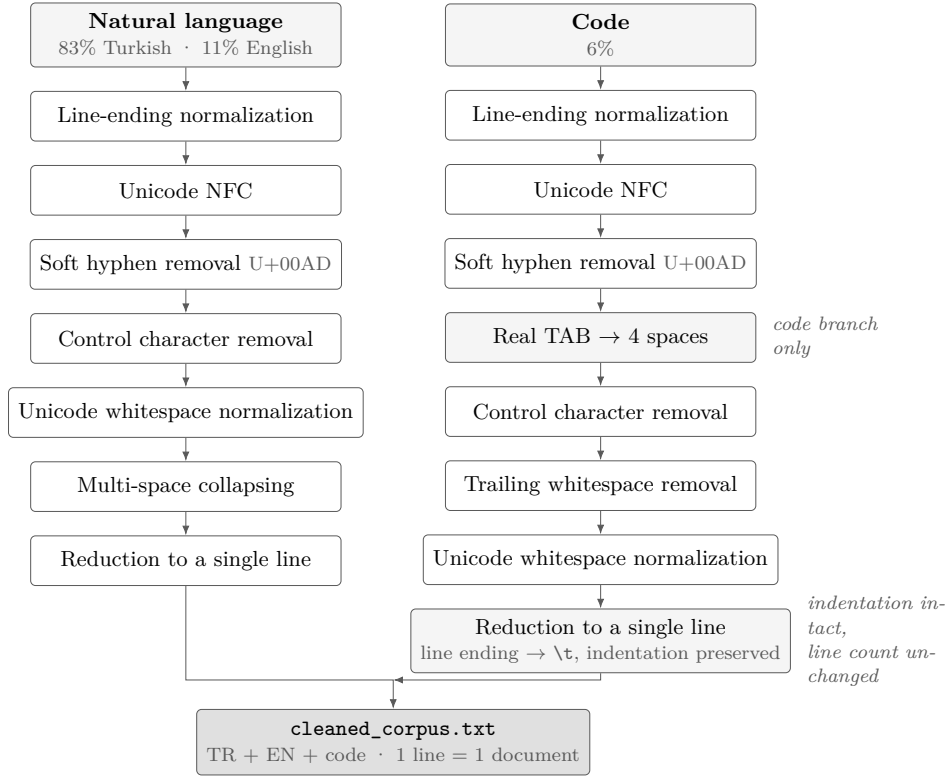


Figure 4: The pre-tokenizer stage: separate but parallel normalization chains for natural language and code data. Shaded boxes occur in the code branch only.

The shared normalization pipeline. In order: line-ending normalization ($\backslash r \backslash n, \backslash r \rightarrow \backslash n$), Unicode NFC, soft hyphen (U+00AD) removal, control character removal (Cc/Cf; excluding $\backslash n$ and $\backslash t$), Unicode whitespace normalization and multi-space collapsing. The ordering is not arbitrary: line-ending normalization must come before everything else. The rationale for Unicode whitespace normalization rests on measurement in text scraped from HTML sources, ` ` (U+00A0) was very common (26 consecutive occurrences in one legislative example; present in 99% of records in the Court of Cassation corpus). These characters are caught neither by the $[\backslash t]^+$ regex nor by Cc/Cf cleaning (they belong to the Zs category), and if left uncleaned the tokenizer learns two versions of the same word. Whitespace and punctuation splitting is not applied manually; it is left to SentencePiece’s own pre-tokenization.

The code branch. There are two constraints: indentation must be preserved and the line count must stay fixed. If indentation is removed, two different Python programs collapse onto the same token sequence. If the line endings of code are left as they are, the corpus balance also collapses: 521,000 code files amount to $\sim 15\text{--}52\text{M}$ lines, whereas all other sources together amount to $\sim 900\text{K}$ lines, and SentencePiece samples on a per-line basis. The solution is to escape the line ending to $\backslash t$; the obligatory accompanying setting is `remove_extra_whitespace=False`. This escape is a technique for tokenizer training only in the pretraining data, code is stored with real line endings.

Three silent problems encountered during training. (i) The default of `max_sentence_length` is 4192 bytes and every longer line is silently discarded; since documents had been chunked with a target of $\sim 6\text{ KB}$, $\sim 75\%$ of them were being lost (solution: 100000). (ii) Because the internal

node index is a 32-bit signed int by default, training crashed once the corpus reached 3.63 billion characters (solution: `train_extremely_large_corpus=True`). (iii) The normalizer removes consecutive spaces, and `normalization_rule_name="identity"` does not disable this (it is a separate field).

Indentation tokens. `allow_whitespace_only_pieces` is `False` by default and forbids learning pieces consisting of whitespace only. When enabled, the token count for a five-line Python function drops from 46 to 30 (the lossy setting that removes indentation gives 26); in a real file, 5,612 characters of code fell from 2,535 to 2,290 tokens.

Special token layout. `[UNK] = 0`, `[PAD] = 1`, `[CLS] = 2`, `[SEP] = 3`, `[MASK] = 4`, `[EOS] = 5`, `[unused0..15] = 6–21`, byte tokens (`byte_fallback=True`) 22–277, learned pieces 278+. The built-in BOS/EOS mechanism was disabled (`bos_id=-1`, `eos_id=-1`); otherwise the vocabulary would contain two sets of tokens for the same job.

Two rejected proposals. Adding Turkish suffixes as `user_defined_symbols`: this list does not respect word boundaries and, if *lar* were added, would forcibly cut through the middle of roots such as “Dilara”; it is moreover unnecessary, as unigram learns frequent suffixes on its own. The second is `normalization_rule_name="nmt_nfkc"`: NFC is already applied offline, and NFKC is more aggressive ($^2 \rightarrow 2$, $fi \rightarrow fi$), corrupting characters that carry meaning in a code/mathematics corpus. `split_digits` was likewise not used; its main benefit appears in generative tasks.

HuggingFace conversion. There is deliberately no pre-tokenizer had a Metaspace or Whitespace pre-tokenizer been used, consecutive spaces would be split one by one and the learned indentation tokens could never be selected during encoding. Control pieces were given a score of -1000 : since `tokenizers.Unigram` does not recognize type information, the lattice can produce identifier 4 from a literal `[MASK]` string in the text; a negative score prevents this, while atomic matching at inference is unaffected because it comes from the `AddedToken` mechanism.

C Implementation: Four Silent-Failure Pitfalls and Memory

Since HuggingFace’s ModernBERT implementation is bidirectional only, the attention path was rewritten. Four pitfalls, each capable of silently producing an incorrect model:

1. **Causal local window.** ModernBERT’s local window is symmetric ($w/2$ left + $w/2$ right). In the CLM phase a symmetric window leaks the future, and this is not visible in the loss the model silently copies. The window must be $(w-1, 0)$ in causal mode and revert to $(w/2, w/2)$ in bidirectional mode.
2. **Boundary-aware packing.** Documents are packed into fixed blocks, but attention must not cross the document boundary (`cu_seqlens`).
3. **Position-id resetting.** In a packed sequence, RoPE positions must start from zero for each document; otherwise a short document at the end of the pack receives the positional encoding of a long history it has never seen. In this project positions come from a single source (the collator’s explicit `position_ids`); FlashAttention’s own internal rotary is deliberately disabled.
4. **Document-boundary masking in the loss.** `cu_seqlens` cuts attention but does not cut the loss: in the shifted-label CLM loss, the last token of the i -th document in the pack

tries to predict the first token of the $(i+1)$ -th document. The label of each document’s final position must be -100 .

Memory. In the CLM phase, $\sim 99\%$ of positions are labelled. In a batch of $\sim 98,000$ tokens the logits tensor alone amounts to $98\text{K} \times 50,048$ in `bf16` ≈ 9.8 GB, plus 19.5 GB with `.float()` and a further 19.5 GB with `log_softmax` it does not fit even on a 96 GB card. Plain chunking is not enough, because autograd stores the logits of every chunk for backpropagation. With `torch.utils.checkpoint` the logits are recomputed during backpropagation and peak memory falls to that of a single chunk (~ 1.6 GB); equivalence was verified by test (difference $0.00\text{e}+00$).

D TabiBench Task-Level Results

Table 17: TabiBench, all 28 tasks and the eight category averages. Scores for TabiBERT, BERTurk, YTU-Cosmos-BERT, TurkishBERTweet and mmBERT are taken from Türker et al. [26], Tables 6–13. Bold: best in row among monolingual Turkish models; mmBERT is excluded from the ranking, following the source work. MB-TR denotes ModernBERT-TR, which publishes TabiBench results only at category level [1]; its per-task scores are marked “?”. Category averages are weighted by test-set size; the overall score is the unweighted macro average of the eight categories. Metrics are as listed in Table 10.

Task	Test size	Mogan	Tabi	BERTurk	YTU	TRBw	mmBERT	MB-TR
<i>Text classification</i>								
NewsCat	250	97.22	95.20	95.60	97.21	91.98	94.80	?
BilTweetNews	150	53.32	50.11	57.87	53.39	53.16	49.06	?
GenderHateSpeech	2,000	66.57	69.01	68.25	71.02	68.58	66.45	?
ProductReviews	35,275	84.72	84.32	84.30	85.04	80.37	83.51	?
<i>Weighted avg</i>		83.71	83.44	83.42	84.25	79.71	82.54	85.21
<i>Token classification</i>								
WikiNER	1,000	75.07	76.41	79.44	78.96	72.60	75.97	?
WikiANN-TR	10,000	92.71	95.34	95.37	95.41	94.05	95.85	?
PosUD-BOUN	979	90.33	90.40	89.82	89.19	89.74	90.72	?
PosUD-IMST	1,100	92.97	94.07	94.60	94.30	93.21	94.26	?
<i>Weighted avg</i>		91.20	93.42	93.67	93.60	92.02	93.81	90.78
<i>STS</i>								
SICK-TR	4,927	85.97	85.00	85.95	85.27	78.35	87.81	?
STSb-TR	1,379	83.49	83.84	83.12	82.55	66.96	84.31	?
<i>Weighted avg</i>		85.43	84.74	85.33	84.68	75.86	87.05	86.08
<i>NLI</i>								
SNLI-TR	9,824	86.18	86.47	87.21	87.18	83.05	86.44	?
MultiNLI-TR	4,923	80.06	80.60	78.57	78.15	71.21	80.28	?
<i>Weighted avg</i>		84.14	84.51	84.33	84.16	79.10	84.38	84.74
<i>Question answering</i>								
TQuAD	2,520	69.99	72.34	63.30	32.01	38.04	71.55	?
XQuAD	179	45.60	32.61	15.96	24.25	39.40	70.40	?
<i>Weighted avg</i>		68.37	69.71	60.16	31.50	38.13	71.47	67.03
<i>Academic</i>								
PubmedRCT-20K-TR	1,500	74.03	75.32	75.61	75.32	70.07	74.37	?
SciCite-TR	1,859	82.63	83.29	81.60	84.09	79.59	83.27	?
ThesisAbstract-11K	1,683	53.44	50.77	49.20	47.56	31.18	52.47	?
MedNLI-TR	1,422	80.76	80.85	79.90	80.62	72.05	80.83	?
<i>Weighted avg</i>		72.62	72.44	71.40	71.78	63.12	72.65	71.83
<i>Information retrieval</i>								
BiText	3,000	99.51	99.42	96.77	97.25	95.98	99.49	?
MsMarco-TR	31,692	83.29	83.31	81.73	81.99	74.20	83.68	?
Scifact-TR	339	79.37	74.22	78.47	79.88	68.25	80.00	?
Fiqa-TR	1,706	48.65	50.67	49.46	51.10	36.23	52.29	?
NFCorpus-TR	12,334	35.07	31.19	32.47	28.36	27.79	33.44	?
Quora-TR	15,675	90.95	92.47	92.72	92.87	86.84	92.78	?
<i>Weighted avg</i>		75.78	75.44	74.84	74.29	68.40	76.20	77.51
<i>Code retrieval</i>								
Apps-TR	3,770	22.39	17.49	18.18	14.47	4.48	32.24	?
CosQA-TR	500	88.54	88.11	85.72	82.04	82.32	89.31	?
StackoverflowQA-TR	1,994	86.90	82.85	79.93	82.65	67.01	90.43	?
CodeSearchNet-21K-TR	3,000	86.38	84.12	78.15	79.35	70.41	88.36	?
<i>Weighted avg</i>		60.57	56.95	54.54	53.80	43.49	66.02	60.16
OVERALL (macro)		77.73	77.58	75.96	72.26	67.48	79.26	77.92

Data-loading corrections. Three defects in the reference loaders were found and fixed before measurement; all three distort scores silently rather than raising errors, and are documented here for reproducibility. **(i) WikiNER label mapping.** The loader derives label identifiers from the split being loaded. When labels arrive as `ClassLabel` integers and a split does not contain every label, the mapping shifts and 0 tokens are decoded as entities. The label set was made independent of the split. **(ii) MultiNLI test split.** The loader expects an unsplit `validation_matched` (9,815 rows) and divides it itself; since the source dataset already ships it divided, the test set was halved. The two parts were merged. **(iii) Code retrieval splits.** The loader looks for a split named `dev`; the source datasets have since been re-uploaded under the name `validation`, which silently triggered a fallback branch that discarded three quarters of the training set (`apps_tr`: 308 rows instead of 1,235). The split name was corrected.

All 28 runs completed with a zero exit status, category membership matches the reference work exactly with no task counted twice, and score scales are consistent (QA reported on 0–100, all others on 0–1 and multiplied by 100 in the tables).

E MTEB(Turkish) Task-Level Results

Table 18: MTEB(Turkish), all 26 tasks. Bold: best in row among student models. The rightmost column is the teacher model.

Task	Mogan	MBERT-TR	mE5	BGE-M3	Mursit	Qwen3 (teach.)
<i>Retrieval</i>						
ArguAnaTR	51.99	49.93	49.07	50.43	45.70	61.84
CQADupstackGamingTR	54.96	56.36	61.29	58.41	53.60	67.08
FiQA2018TR	47.85	46.20	47.82	45.00	40.65	53.35
MSMarcoTRRetrieval	58.09	57.87	58.35	57.93	58.95	58.34
NFCorpusTR	9.96	9.45	9.62	9.41	9.50	12.00
QuoraRetrievalTR	95.80	95.18	95.91	95.82	95.05	95.75
SCIDOCSTR	3.30	3.37	4.18	3.69	2.70	5.78
SciFactTR	81.23	77.06	83.99	78.81	72.79	90.29
SquadTRRetrieval	71.70	75.92	77.88	76.69	62.10	79.10
TQuadRetrieval	86.43	87.13	87.25	90.52	81.56	89.30
XQuADRetrieval	94.03	95.04	96.22	96.11	90.11	96.20
<i>Pair classification</i>						
MnliTr	67.33	65.92	63.88	66.82	54.09	61.31
SnliTr	64.37	67.29	52.47	60.56	51.51	57.43
XNLI	76.26	74.43	72.75	78.78	59.98	69.52
<i>STS</i>						
STSbTR	79.49	77.61	81.23	79.60	74.60	80.04
<i>Classification</i>						
THYSentiment	69.05	68.15	66.13	67.36	57.68	66.15
TSTimelineNewsCat	64.39	67.91	65.02	64.60	63.08	63.48
Turkish75News	93.33	93.33	92.67	88.00	96.67	94.67
TurkishIrony	56.25	57.75	58.83	53.25	50.92	55.75
TurkishMovieSentiment	86.08	89.34	85.25	86.64	72.63	83.38
TurkishNewsCategory	89.36	93.64	93.80	90.00	90.00	92.60
TurkishOffensiveLang	71.72	71.51	60.68	61.97	62.06	63.36
TurkishProductSentiment	72.70	70.46	67.22	70.46	56.23	64.21
<i>Clustering</i>						
TurkishAbstractCorpus	67.93	62.02	61.94	59.39	60.20	63.73
TurkishColumnWriting	64.99	64.38	61.80	62.21	62.86	62.29
<i>Bitext</i>						
WMT16BitextMining	97.24	94.04	98.99	98.99	86.72	98.18
OVERALL (26 tasks)	68.30	68.13	67.47	67.36	62.00	68.66