ColBERT-Zero: To Pre-train Or Not To Pre-train ColBERT

Antoine Chaffin [1] Luca Arnaboldi [2] Amélie Chatelain [1] Florent Krzakala [2]

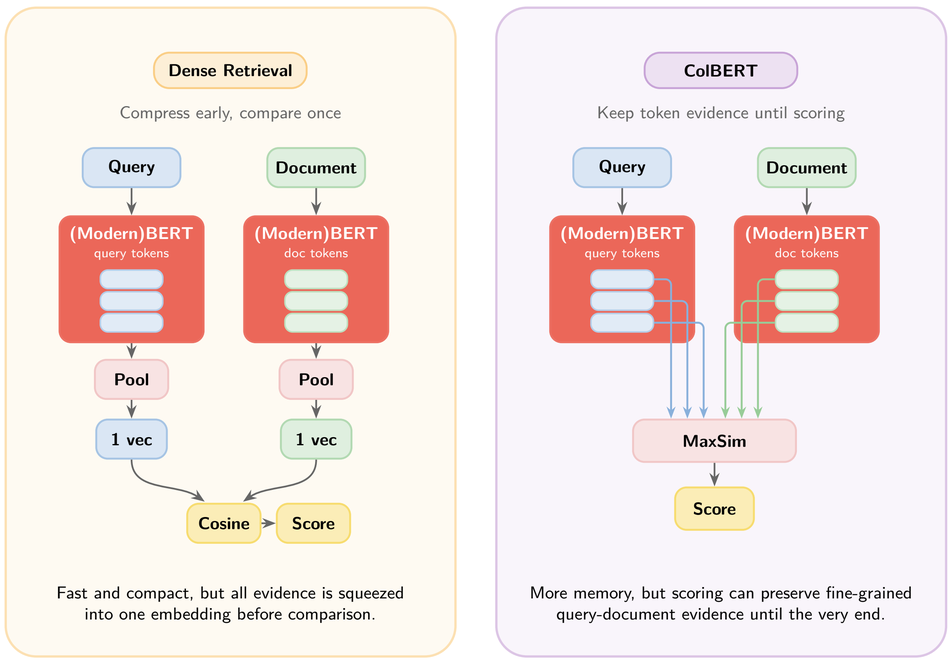
[1]LightOn, France [2]EPFL, IdePHICS Lab, Switzerland

Core Question

Can a strong dense encoder plus knowledge distillation recover most of ColBERT's gains, or is large-scale multi-vector pre-training truly essential?

Two Retrieval Philosophies

Dense retrieval compresses evidence **before** comparison. ColBERT delays compression and scores with **token-level evidence** still intact.

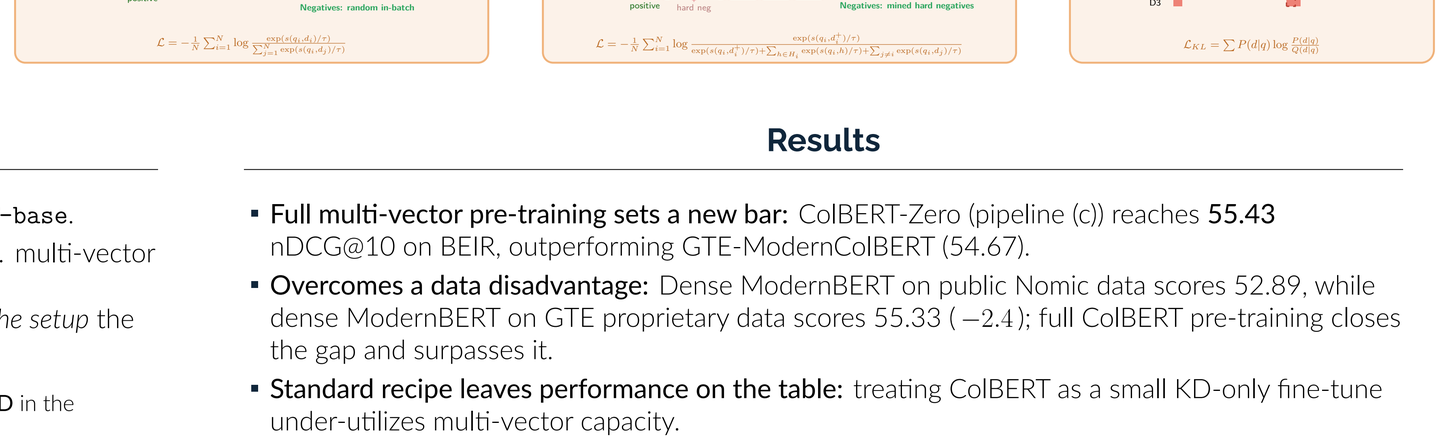


Fast and compact, but all evidence is squeezed into one embedding before comparison.

More memory, but scoring can preserve fine-grained query-document evidence until the very end.

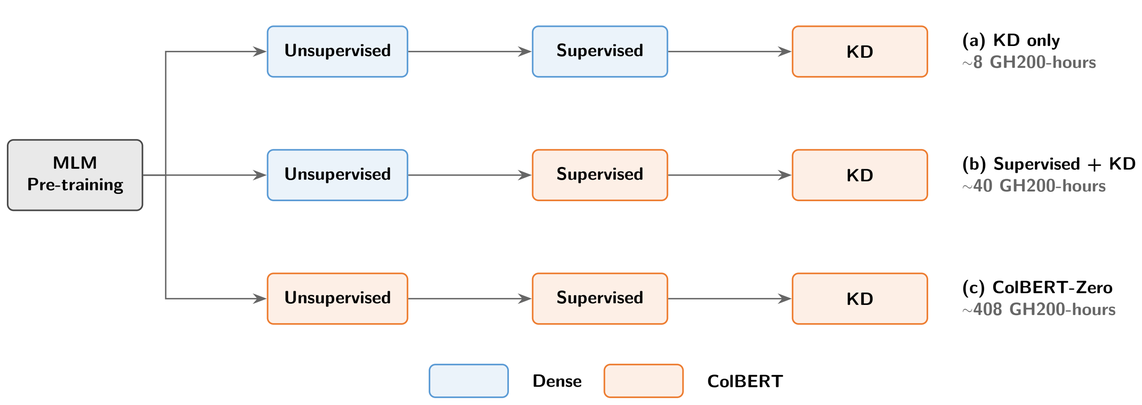
Training Story

Standard practice is KL distillation only from a strong dense model; we ask whether full multi-vector training is better.



Methodology

- **Base model:** Dense and multi-vector variants share the same backbone: `ModernBERT-base`.
- **Training Data: Nomic Embed** public datasets (chosen to enable a controlled dense *vs.* multi-vector comparison: same data mixture, same base model, same pipeline).
- **Pipelines compared:** To isolate the effect of **multi-vector pre-training**, we vary only *the setup* the contrastive phases happen in (dense or multi-vector).
  - (a) **KD-only:** Distill a multi-vector student from a dense pre-trained model.
  - (b) **Hybrid:** Start from Nomic's `modernbert-embed-unsupervised` (dense), then run **supervised + KD** in the multi-vector setting.
  - (c) **ColBERT-Zero:** Perform all pre-training phases directly as a **multi-vector** model.
- **PyLate Library:** Custom training framework for large-scale multi-vector distillation
  - **GradCache**: Gradient caching compatible with contrastive (scale BS without VRAM constraints)
  - **Multi-GPU gathering**: Use samples already computed on the other GPUs
  - **Single-source batches**: Prevent shortcut learning



The critical role of Prompt Alignment

- **Misalignment is silent poison:** Stripping pre-training prompts hurts performance; adding prompts to models not pre-trained with them also hurts.
- **Prompts provide intrinsic benefit:** Full pre-training with prompts (55.43) **vs.** without (54.61) = **0.82-point gap** with perfect alignment.
- **Why?** Prompts act as *implicit query expansion* slots, storing global sequence information (modern Flash Attention broke the original [PAD] token trick).

Method / Training	Baseline	+Prompt	+Length	+Both
KD from dense supervised (with prompts)				
Distilled	52.58	+1.25 (53.83)	+0.70 (53.28)	**+1.51 (54.09)**
Supervised + KD from dense unsupervised (with prompts)				
Supervised	**51.33**	−0.25 (51.08)	−0.25 (51.08)	−0.61 (50.72)
Distilled	54.44	+0.51 (54.95)	+0.38 (54.82)	**+0.68 (55.12)**
Supervised + KD from ColBERT unsupervised (with prompts)				
Supervised	52.47	−0.31 (52.16)	**+0.23 (52.70)**	−0.66 (51.81)
Distilled	54.17	+0.42 (54.59)	+0.30 (54.47)	**+1.26 (55.43)**
Supervised + KD from ColBERT unsupervised (without prompts)				
Supervised	52.39	–		**+0.23 (52.62)**
Distilled	**54.61**	–		−0.43 (54.18)

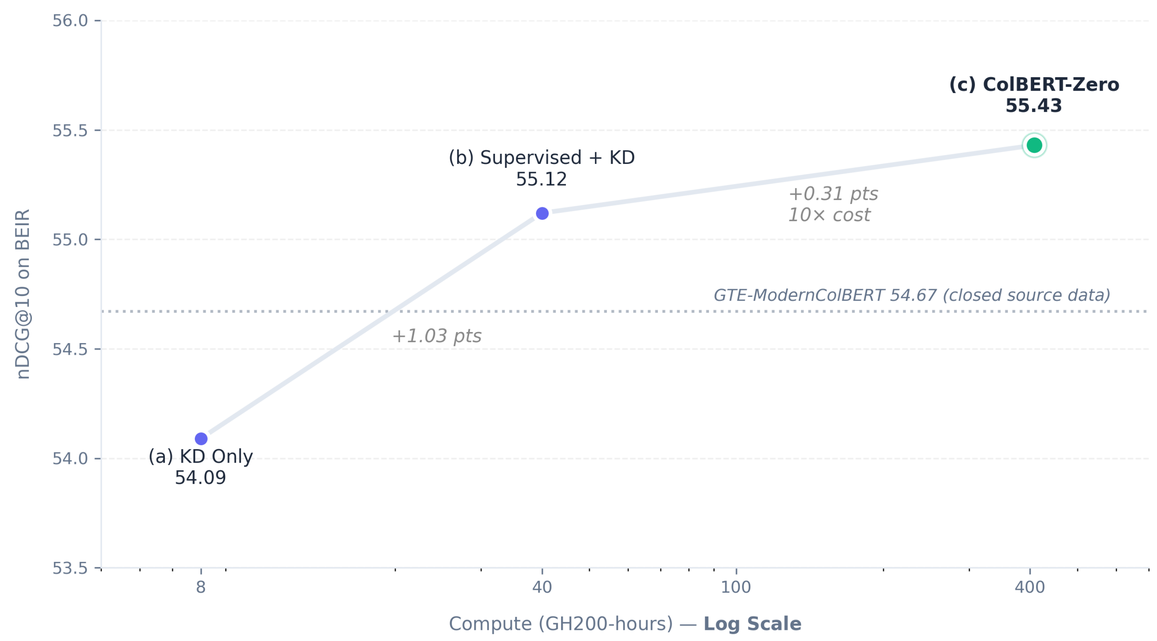
Practical takeaway: Always align prompts with the base model's pre-training. Misalignment erodes performance silently, but this effect decreases with stronger fine-tuning.

Results

- **Full multi-vector pre-training sets a new bar:** ColBERT-Zero (pipeline (c)) reaches **55.43** nDCG@10 on BEIR, outperforming GTE-ModernColBERT (54.67).
- **Overcomes a data disadvantage:** Dense ModernBERT on public Nomic data scores 52.89, while dense ModernBERT on GTE proprietary data scores 55.33 (-2.4); full ColBERT pre-training closes the gap and surpasses it.
- **Standard recipe leaves performance on the table:** treating ColBERT as a small KD-only fine-tune under-utilizes multi-vector capacity.

Supervised + KD is the efficiency sweet spot.

- **Near-SOTA at a fraction of the cost:** pipeline (b) reaches **55.12** nDCG@10, within **0.31** of full ColBERT-Zero (55.43).
- **$\approx 10\times$ cheaper:** $\sim$40 GH200-hours vs. $\sim$408 for full multi-vector pre-training.
- **Practical takeaway:** do multi-vector **supervised training + KD** to capture almost all gains, and reserve full unsupervised multi-vector pre-training for maximum accuracy.



Deliverables

- **ColBERT-Zero checkpoints:** Full multi-vector pre-training and KD-only variants on Hugging Face (including prompts ablations!)
- **Boilerplate training scripts:** PyLate-based scripts for all pipelines, with modular options for pre-training phases and prompt configurations (available in PyLate's GitHub repo).

Acknowledgements

This work was supported by the Swiss Center of Scientific Computing (CSCS) under the grant IDa120.

Models & Checkpoints

LIR'22: 1st Late Interaction Workshop @ ECIR 2026, Delft

Code