The Embedder's Dilemma: LLMs Are Better, but at What Cost?

Adnan El Assadi
Harvard University

Niklas Muennighoff
Stanford University

Jinhyuk Lee
Independent Researcher

Abstract

Should you replace your text-embedding pipeline with a large language model? We answer this with a controlled, cost-aware comparison of ten LLMs across six families and 26 embedding models (118M to 14B parameters) on 37 tasks spanning classification, semantic textual similarity (STS), clustering, pair classification, and retrieval. In aggregate the two paradigms are effectively tied: the best LLM (Gemini 3.1 Pro, 77.6) and the best embedding model (77.2) differ by 0.4 points. Their strengths differ by task: LLMs lead on reasoning-heavy retrieval, embedding models lead on classification, and the two match on clustering, STS, and pair classification. Reaching that parity is expensive. An LLM costs up to 1,431× more than an embedding model of comparable quality ($154 vs. $0.11 per benchmark pass), and the open LLMs tested process tokens 2.5 to 736× more slowly on the same GPU. Reasoning tokens account for 28 to 81% of LLM inference cost; lower reasoning budgets preserve or improve retrieval quality for most models in our ablation. The Pareto frontier contains the leading embedding models and one LLM, Gemini 3.1 Pro. These results support a division of labour: use embedding models for similarity, classification, and clustering, and reserve LLMs for reasoning-intensive retrieval. Our code, datasets, and results are publicly available at https://github.com/embeddings-benchmark/embedders-dilemma.

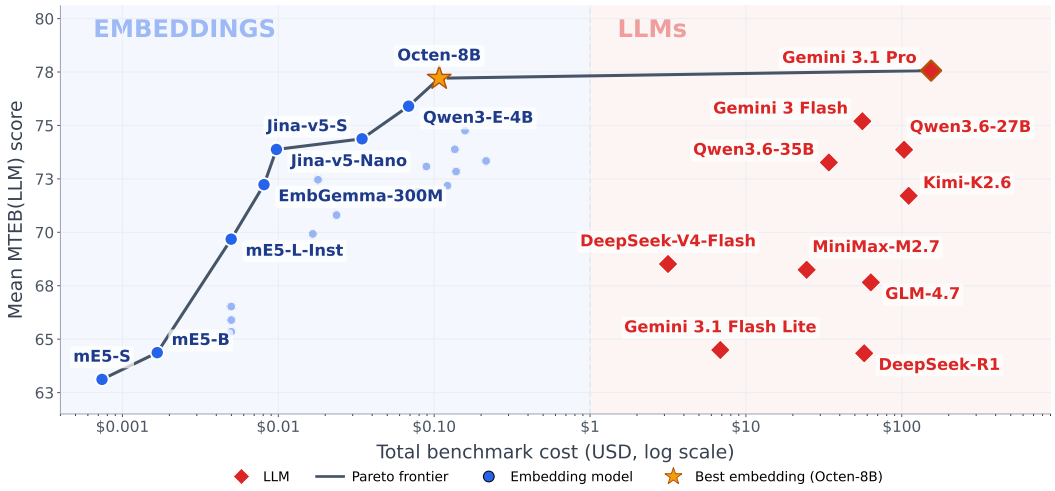

Figure 1: **Cost vs. performance across 36 models on MTEB(LLM).** The frontier contains the leading embedding models and Gemini 3.1 Pro, which extends it by 0.4 points at 1,431× the cost of a comparable embedding.

1 Introduction

We show that an LLM without task-specific training now matches the best text embedding models across a broad suite of standard embedding tasks. Embedding models reach this quality through specialised contrastive training, hard-negative mining, and multi-stage distillation; our best LLM reaches it without a dedicated embedding-training pipeline.[1] The deployment costs differ sharply: LLM inference is substantially more expensive and slower on the same hardware. We call this the **Embedder's Dilemma**: LLMs now match the best embedding models in aggregate, but at much higher cost and lower throughput.

We ask a practical question: *should you replace your embedding pipeline with an LLM?* Our results favour embedding models as the default and LLMs for reasoning-heavy retrieval. A hybrid pipeline captures the retrieval benefit while controlling cost and throughput.

We compare ten frontier LLMs from six families with 26 text embedding models (118M–14B parameters). The evaluation uses **MTEB(LLM)**, a 37-task benchmark built from the Massive Text Embedding Benchmark (MTEB; Muennighoff et al., 2023), covering classification, semantic textual similarity (STS), clustering, pair classification, and retrieval. The full model list appears in §3. Each MTEB(LLM) task is a fixed subset of the corresponding MTEB task, released on Hugging Face under `mteb/llm-eval-*`; both paradigms are scored on exactly that subset. MTEB(LLM) scores compare models within this benchmark; MTEB leaderboard scores use the full test sets and follow a different evaluation basis. We pair the quality comparison with exact cost accounting (API token tracking for LLMs; GPU throughput benchmarking for embedding models), producing a cost–performance Pareto frontier spanning both paradigms (Figure 1).

Contributions. We release MTEB(LLM), the first benchmark to compare LLMs and embedding models across all five MTEB task categories, with cost measured alongside quality for every model. We implement it with the MTEB framework, part of an ecosystem that spans multilingual text (MMTEB; Enevoldsen et al., 2025), image (MIEB; Xiao et al., 2025), audio (MAEB; El Assadi et al., 2026a), and video (MVEB; El Assadi et al., 2026b). The Pareto frontier contains the leading embeddings and Gemini 3.1 Pro, whose small score gain comes at three orders of magnitude higher cost. We identify the *thinking-token tax*: reasoning accounts for 28–81% of LLM inference cost, while lower reasoning budgets preserve retrieval quality for most models in our ablation (§4.6).

Our empirical findings are:

1. **In aggregate, the paradigms tie.** The best LLM and embedding model differ by 0.4 points, within statistical noise (§4.2). The pattern holds across the task suite.
2. **The advantage is task-specific.** LLMs lead on reasoning-heavy retrieval, embedding models lead on classification, and the paradigms are statistically even on clustering, STS, and pair classification. LLMs excel when cross-document reasoning is required; embedding pipelines excel wherever geometric similarity or labelled-reference matching suffices (§3.2–4.2).
3. **The cost gap is large.** Gemini 3.1 Pro costs 1,431× as much as a comparable embedding model, and the gap persists across hardware scenarios. Reasoning contributes substantially to this asymmetry. Lower reasoning budgets preserve or improve retrieval for most evaluated models (§4.4–4.6).
4. **Throughput lags by orders of magnitude, even on the same hardware.** Served on an identical H100, open-weight LLMs process 2.5–736× fewer tokens per second than embedding models, consistent with the architectural difference between autoregressive generation and a single encoder pass (§4.5; Figure 7).

LLMs and embedding models therefore serve complementary roles. Embedding pipelines remain the efficient default for classification, similarity, and clustering; LLMs are most useful for reasoning-intensive retrieval.

[1] While embedding models have increasingly adopted LLMs as their backbone (Tao et al., 2024; Nie et al., 2025), they output vectors and use training tailored to embedding tasks. Meanwhile, "LLMs" in this work are generative models prompted to output text, with no dedicated embedding training.

2 Related Work

2.1 Text Embedding Models, Evaluation, and Cost

Dense text representations have evolved rapidly from contextualised encoders (Devlin et al., 2019; Reimers & Gurevych, 2019) to billion-parameter, instruction-tuned models (Su et al., 2023). Leading systems such as E5 (Wang et al., 2022; 2024), NV-Embed (Lee et al., 2025a), SFR-2 (Meng et al., 2024), and Gemini Embedding (Lee et al., 2025b) achieve strong performance through contrastive training on large curated corpora. MTEB (Muennighoff et al., 2023) standardised evaluation across eight task categories and now tracks hundreds of models on a public leaderboard.[2] The framework has expanded across multilingual text (MMTEB; Enevoldsen et al., 2025), image (MIEB; Xiao et al., 2025), audio (MAEB; El Assadi et al., 2026a), and video (MVEB; El Assadi et al., 2026b). HUME (Assadi et al., 2025) measures human performance on MTEB tasks and benchmarks LLMs as annotators. Our study asks whether an LLM can replace an embedding pipeline. We extend this line of evaluation with exact cost and throughput accounting across both paradigms.

Schwartz et al. (2020) introduced "Green AI," arguing that accuracy focus neglects computational and environmental costs. Gonzalez (2026) constructs Pareto frontiers of fine-tuned BERT-scale encoders vs. GPT-4o on text classification, finding encoders match or exceed frontier LLMs at 1–2 orders of magnitude lower cost. The "overthinking" literature (Sui et al., 2025) documents verbose reasoning traces and their computational cost. Snell et al. (2024) show that adaptive test-time compute can outperform a uniform reasoning budget. We study these questions on classification, similarity, clustering, and retrieval workloads.

2.2 LLMs, Cross-Document Reasoning, and Retrieval

LLMs with controllable reasoning now span closed APIs (Gemini 3.1 Pro, GPT-5 (Singh et al., 2025), Claude Opus 4.6 (Anthropic, 2026)) and open weights (Qwen3 (Yang et al., 2025), Llama 3 (Grattafiori et al., 2024), and DeepSeek-R1 (DeepSeek-AI et al., 2025), among others). We evaluate ten models across six families (§3), including dense and mixture-of-experts models from closed and open providers. This diversity lets us distinguish recurring patterns from effects specific to one provider. Bucher & Martini (2024) find that fine-tuned encoder models significantly outperform zero-shot frontier LLMs across all tested classification benchmarks, with margins ranging from 5 to over 75 F1 points depending on task granularity, consistent with our 5.6-point embedding advantage under the kNN vs. zero-shot comparison.

Dense retrieval (Karpukhin et al., 2020) and the BEIR benchmark (Thakur et al., 2021) established strong zero-shot bi-encoder performance. Bi-encoders process queries and documents independently, without cross-attention between them. The BRIGHT benchmark (Su et al., 2024) complements BEIR with reasoning-intensive queries. Its top embedding model scores 18.3 nDCG@10, while LLM-augmented retrieval improves by up to 12.2 points, showing the limitations of independent query and document encoding on such tasks. RAG (Lewis et al., 2020; Gao et al., 2024) pairs embedding retrieval with LLM reasoning. We evaluate this reranking stage directly (Sun et al., 2023a), crossing first-stage retrievers with cross-encoder and LLM listwise rerankers on BEIR and BRIGHT (§4.3). Lu et al. (2025) show CoT reranking consistently underperforms direct-output reranking on BEIR/BRIGHT despite higher cost, the document-ranking counterpart to our finding that reduced thinking holds or improves retrieval on all six tasks. Appendix F covers LLMs as text encoders, GritLM, and the wider set of frontier models.

[2] https://huggingface.co/spaces/mteb/leaderboard

3 Experimental Setup

3.1 Models

LLMs. We evaluate ten frontier LLMs spanning six families: three Gemini 3 models via the Gemini API (Gemini 3.1 Pro, Gemini 3 Flash, Gemini 3.1 Flash-Lite) and seven open-weight models served via OpenRouter (DeepSeek-R1, DeepSeek-V4-Flash, Qwen3.6-27B, Qwen3.6-35B-A3B, GLM-4.7, Kimi-K2.6, and MiniMax-M2.7). Together they span dense and mixture-of-experts architectures, reasoning and instruct variants, and a wide capability and cost range. Gemini 3.1 Flash-Lite provides a non-reasoning baseline; the other models support extended reasoning (chain-of-thought tokens (Wei et al., 2022), billed at the output token rate).

Embedding models. We evaluate 26 text embedding models from 118M to 14B parameters. The multi-model families are multilingual-E5 (Wang et al., 2022; 2024), Qwen3-Embedding (Zhang et al., 2025), F2LLM-v2 (Zhang et al., 2026), Jina-v5 (Akram et al., 2026), and GTE-Qwen2 (Li et al., 2023). We also include models from Tencent (Zhao et al., 2025), NVIDIA (Babakhin et al., 2025), Salesforce (Meng et al., 2024), Snowflake (Yu et al., 2024), and Google (Vera et al., 2025), together with E5-Mistral (Wang et al., 2024), GritLM (Muennighoff et al., 2024), Linq-Embed-Mistral (Choi et al., 2024), BGE-M3 (Chen et al., 2025), and Octen-8B. All embedding models run locally on a single NVIDIA H100 80GB HBM3 GPU. Table 1 lists every model with its score and cost; full details appear in Appendix B.

3.2 Tasks and Evaluation Protocol

We introduce **MTEB(LLM)**, a 37-task benchmark covering classification, STS, clustering, pair classification, and retrieval, implemented with the MTEB framework (Muennighoff et al., 2023). It follows MTEB's task and result interfaces, allowing new models and datasets to use the same evaluation pipeline. Each MTEB(LLM) task is a new LLM-specific evaluation dataset derived from the corresponding original MTEB task (fixed seed = 42; Appendix B.5), ensuring all submissions are evaluated on identical data.

Why these tasks. We use subsets rather than MTEB(eng, v2) because generative evaluation is orders of magnitude more expensive: one pass over the full suite would cost hundreds of dollars per LLM, and corpus-in-context retrieval requires the corpus to fit in the prompt. Tasks are selected to span all five categories, cover multiple languages and domains, and keep retrieval corpora small enough for shorter-context models. MTEB's task-selection analysis was also designed around embedding models, so v2 is not automatically the right frame for a cross-paradigm comparison.

We compare complete deployment pipelines: embedding models use kNN for classification, cosine similarity for STS and retrieval, and k-means for clustering; LLMs receive zero-shot prompts. This comparison reflects a common deployment setting in which practitioners consider LLMs because labelled data are scarce. Our few-shot ablation (§4.7) tests whether a small number of in-context examples narrows the resulting gap.

Classification (8 tasks). Sentiment (IMDB, ToxicConversations, TweetSentiment), intent detection (Banking77: 77 classes; MassiveIntent: 60 intents), and multilingual classification (AmazonCounterfactual, MTOPDomain, MassiveScenario). *LLM:* Zero-shot structured output for each test split sample. *Embedding:* kNN trained on embeddings on the labelled training split, then used for test split predictions. *Metric:* Accuracy.

Semantic Textual Similarity (10 tasks). English benchmarks (STSBenchmark, SICK-R, STS12–16), a biomedical benchmark (BIOSSES), and multilingual benchmarks (STS17, STS22v2). *LLM:* Rates similarity on each task's native scale, returning a float. *Embedding:* Cosine similarity between embeddings. *Metric:* Spearman correlation.

Clustering (9 tasks). Scientific papers (ArXiv, BioRxiv, MedRxiv) and online discussions (Reddit, StackExchange, TwentyNewsgroups). *LLM:* All documents in a single prompt

with sequential identifiers; the model receives the ground-truth cluster count k and returns cluster assignments as a JSON array. *Embedding:* k-means on document embeddings. *Metric:* V-measure.

Pair Classification (4 tasks). Duplicate detection (SprintDuplicateQuestions), paraphrase identification (TwitterURLCorpus), legal matching (LegalBenchPC), and entailment (RTE3, multilingual). *LLM:* Binary output per pair with reasoning. *Embedding:* Cosine similarity with threshold sweep (MTEB convention; best threshold chosen post-hoc on the test set). *Metric:* Average precision and accuracy.

Retrieval (6 tasks). A domain-diverse suite: legal (AILAStatutes; LegalBench consumer-contracts QA (Guha et al., 2023)), finance (HC3-Finance (Guo et al., 2023)), public-health QA, French QA (FQuAD (d'Hoffschmidt et al., 2020)), and Danish social media (TwitterHjerne (Vejlgaard Holm et al., 2025)). *LLM:* Corpus-in-context (Lee et al., 2024): the full corpus is placed in the prompt with sequential identifiers; prompt caching (Gim et al., 2024) amortises the corpus prefix across queries. Corpora are deliberately small (82–415 documents) so that models with shorter context windows can be evaluated on identical data; this keeps the whole corpus readable in one prompt, which production-scale retrieval does not permit (see §5). *Embedding:* Cosine similarity ranking. *Metric:* Recall@1 (nDCG@k results in released data files).

3.3 Cost Methodology

Embedding costs. We measure maximum-throughput token processing on an H100 80GB at sequence length 512 (largest fitting batch) and compute cost as r_{GPU}/T, where $r_{\text{GPU}} = \$2.49/\text{hr}$ (Lambda Labs H100 spot, March 2026) and T is tokens per hour. Costs range from \$0.001 (mE5-small) to \$0.22 (F2LLM-v2-14B); full per-model throughput and cost data appear in Table 18 (Appendix H.2).

LLM costs. Token usage is extracted from each provider's `usage_stats`. We bill every non-input (generated) token at the output rate:

$$\text{Cost} = (\text{input} - \text{cached}) \cdot r_{\text{in}} + \text{cached} \cdot r_{\text{cache}} + (\text{total} - \text{input}) \cdot r_{\text{out}} \tag{1}$$

where $r_{\text{cache}} = r_{\text{in}}/10$. Reasoning ("thinking") tokens are part of the generated total and are billed at the output rate, matching provider billing; providers report them differently, and we attribute each model's reasoning share from its usage statistics (Appendix H.4). Gemini models are priced at Gemini API rates and open-weight models at OpenRouter public rates (March/June 2026). Costs range from \$3.16 (DeepSeek-V4-Flash) to \$154.14 (Gemini 3.1 Pro); the complete token-level breakdown is in Table 17 (Appendix H.1).

3.4 Throughput Methodology

We serve both paradigms on the same GPU, giving the throughput comparison common hardware and removing API rate limits. This differs from our cost accounting (§3.3), which uses API rates for LLMs and GPU-rental rates for embeddings. We serve the two open-weight LLMs that fit on a single H100 (Qwen3.6-27B dense, Qwen3.6-35B-A3B MoE) under vLLM, and run embedding models batched on the same GPU, measuring both in tokens per second (configuration in Appendix H.4). Gemini is API-only, and larger open models such as DeepSeek-R1 need multiple GPUs. This end-to-end comparison uses each paradigm's standard inference procedure: encoder batching for embeddings and autoregressive decoding for LLMs (Appendix G). Per-model embedding throughput is reported in Table 18 (Appendix H.2); the comparison appears in Figure 7 (Appendix D.1).

	Model	Params	Cls (8)	Clust (9)	STS (10)	PairCls (4)	Retr (6)	Overall (37)	Cost
LLM	Gemini 3.1 Pro	–	85.2	66.6	88.5	83.2	**64.5**	**77.6**	$154
	Gemini 3 Flash	–	84.1	65.7	87.6	86.3	52.3	75.2	$56
	Qwen3.6-27B	–	84.8	53.6	84.9	83.6	62.4	73.9	$103
	Qwen3.6-35B-A3B	–	83.8	55.3	84.0	82.9	60.4	73.3	$34
	Kimi-K2.6	–	84.5	55.4	83.9	78.0	56.7	71.7	$111
	DeepSeek-V4-Flash	–	81.9	43.7	82.2	81.6	53.3	68.5	$3
	MiniMax-M2.7	–	80.9	48.6	81.0	82.8	48.0	68.2	$25
	GLM-4.7	–	84.2	36.5	83.5	79.7	54.4	67.7	$63
	Gemini 3.1 Flash Lite	–	82.9	21.7	85.1	83.7	49.0	64.5	$7
	DeepSeek-R1	–	82.5	31.3	83.7	79.5	44.7	64.3	$57
Embedding	Octen-8B	7.6B	90.1	65.1	88.7	86.1	56.0	77.2	$0.11
	Qwen3-E-8B	7.6B	90.1	65.9	88.5	86.5	54.2	77.0	$0.11
	Qwen3-E-4B	4.0B	89.3	64.6	**88.8**	86.5	50.3	75.9	$0.07
	Nemotron-8B	7.5B	84.8	64.1	86.2	86.7	54.7	75.3	$0.11
	KaLM-12B	11.8B	88.8	63.4	84.6	**87.1**	49.9	74.8	$0.16
	Jina-v5-S	596M	90.4	61.5	86.9	85.1	48.0	74.4	$0.03
	SFR-2	7.1B	**90.8**	**66.7**	77.9	85.3	48.8	73.9	$0.14
	Jina-v5-Nano	212M	89.6	60.4	87.0	85.0	47.4	73.9	$0.01
	F2LLM-14B	14.0B	77.7	66.5	84.1	86.3	52.1	73.3	$0.22
	GTE-Qwen2-7B	7.1B	86.7	65.2	81.6	86.0	45.8	73.1	$0.09

Table 1: **Model overview and per-category results.** All ten LLMs and the ten highest-scoring embedding models. Scores are category means on a 0–100 scale; Overall is their mean. Cost is one MTEB(LLM) pass using API rates for LLMs and H100 throughput at $2.49/hr for embeddings (§3.3). Bold = best shown; full results are in Appendix B.2.

4 Results

4.1 Overall Performance

Across all 36 models, the best LLM and the best embedding model are effectively tied. Gemini 3.1 Pro leads with a mean score of 77.6 on MTEB(LLM), followed closely by Octen-8B (77.2) and Qwen3-E-8B (77.0; full ranking in Appendix B.2). Gemini 3 Flash scores 75.2, and Flash-Lite scores 64.5. The reasoning-capable Gemini models occupy the top LLM ranks.

4.2 Task-Category Analysis

The aggregate tie hides large differences between task categories. Table 1 reports per-category scores for the top models, and full per-category rankings for all 36 models appear in Appendix D. Figure 6 (Appendix D.1) shows this at the task level: some embedding model matches the best LLM on 7 of 8 classification and 7 of 10 STS tasks, but on only 1 of 6 retrieval tasks. Retrieval is where the paradigms separate; elsewhere the choice is about cost, not quality.

Retrieval (+8.5): LLMs lead. Pro leads the best embedding on retrieval (64.5 vs. 56.0) and wins five of the six retrieval tasks, losing only legal statute retrieval (Appendix C.2).

Classification (−5.6): embeddings lead. SFR-2 (Meng et al., 2024) (90.8) outscores Pro (85.2) by a wide margin. The gap widens on fine-grained tasks (Banking77: 77 classes; MassiveIntent: 60 intents), a consequence of the kNN-vs-zero-shot setup we adopt for deployment realism (§3.2); see §5 for the structural argument.

Clustering, STS, and pair classification: statistical ties. On clustering the best embedding (SFR-2, 66.7) edges Pro (66.6); on STS Qwen3-E-4B (Zhang et al., 2025) (88.8) edges Pro (88.5); on pair classification KaLM-12B (Zhao et al., 2025) (87.1) leads Pro (83.2). All three rest on geometric proximity or labelled-reference matching, which cosine similarity over learned embeddings captures directly.

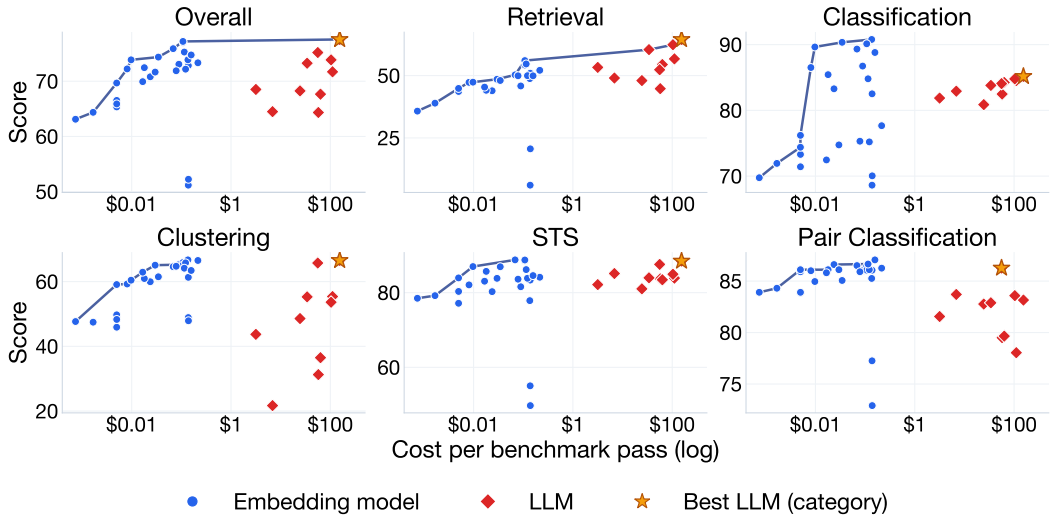

Figure 2: **Cost vs. performance by task category.** Each panel plots score (0–100) against cost per benchmark pass (log scale) for all 36 models; the line is the Pareto frontier over all models and the star marks the best LLM in that category. Pro extends the retrieval frontier; embedding models define the frontiers for classification, clustering, STS, and pair classification.

Statistical significance. A paired bootstrap test (10,000 resamples) places the overall difference between Pro and Octen-8B within statistical noise ($\Delta = +0.3$, $p = 0.85$, 95% CI $= [-2.4, +3.1]$). Category-level tests favour LLMs on retrieval and embeddings on classification; clustering, STS, and pair classification are statistically even (Appendix A, Table 2).

The category-level cost–performance frontiers show the same division (Figure 2): Pro extends the retrieval frontier, while embeddings define the frontiers for classification, clustering, STS, and pair classification.

4.3 Retrieve-then-Rerank

The corpus-in-context comparison above pits single-vector embeddings against LLMs directly; production systems often insert a middle stage: a cross-encoder or LLM reranker over a first-stage shortlist. We evaluate this pipeline on the standard IR benchmarks BEIR (semantic) and BRIGHT (reasoning-heavy), crossing four first-stage retrievers with cross-encoder and LLM listwise rerankers (Figure 3; the full matrix is in Appendix D.3, Table 16). The result follows the same task-dependent pattern: on reasoning-heavy BRIGHT, an LLM listwise reranker improves a strong embedding first stage from 22.3 to 35.1 nDCG@10. On semantic BEIR, the strong embedding alone scores 63.1, ahead of the best reranked configuration at 60.3. Cost scales with shortlist size: reranking a top-100 shortlist costs $10–30 per benchmark, against $154 for reading the full corpus in context. The MoE Qwen3.6-35B-A3B is cheaper still, reaching 33.6 nDCG@10 on BRIGHT for $10 where Qwen3.6-27B reaches 35.1 for $30.

4.4 Cost and Throughput

Embeddings dominate the Pareto frontier. The Pareto frontier contains the embedding models and Gemini 3.1 Pro, which costs 1,431× as much as the best embedding of comparable quality (Figure 1). Pro's marginal 0.4-point aggregate edge over Octen-8B comes with a benchmark cost of $154.14 versus $0.11.

The thinking-token tax. The structural driver of LLM cost is internal reasoning (Equation 1; Figure 4). Reasoning tokens account for 28–81% of inference cost across the reasoning models. Flash-Lite provides a non-reasoning reference point and ranks below most em-

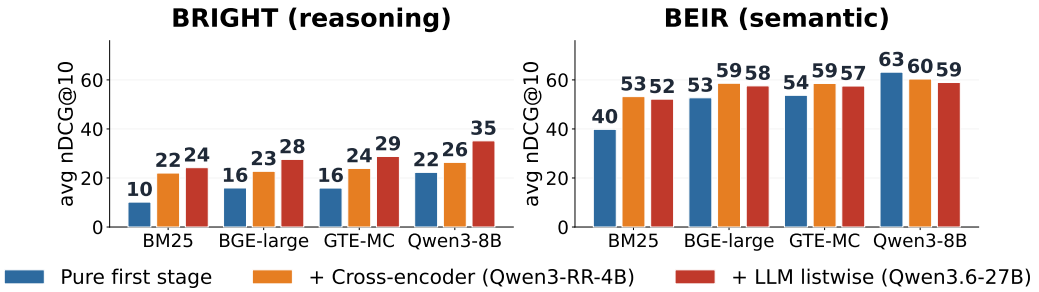

Figure 3: **Retrieve-then-rerank.** LLM listwise reranking improves every first stage on BRIGHT. On BEIR, a strong embedding first stage outperforms all reranked configurations.

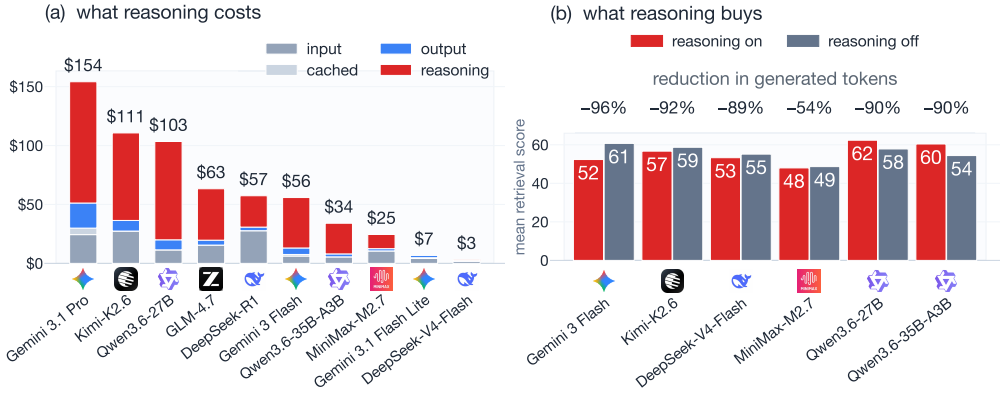

Figure 4: **The thinking-token tax: what reasoning costs, and what it buys.** *(a)* API cost per benchmark pass by token type for all ten LLMs. *(b)* Mean retrieval score with default vs. disabled reasoning for six models from five families; labels show the reduction in generated tokens. Four models preserve or improve retrieval with 54–96% fewer generated tokens; the two Qwen models lose ground.

beddings. This spending buys little on most tasks: lower reasoning budgets preserve or improve retrieval for four of six models (§4.6).

Cost sensitivity. The 1,431× ratio assumes H100 spot pricing ($2.49/hr). Under alternative hardware and pricing the ratio ranges from 338× (commercial embedding API at $0.10/MTok) to 2,424× (L4 GPU at $0.49/hr). The order-of-magnitude gap persists across all evaluated cost assumptions (Appendix H.3).

4.5 Throughput Constraints

Served on the same H100, embedding models process tokens far faster than LLMs (Figure 7, Appendix D.1). The two open-weight LLMs that fit on a single H100 (Qwen3.6-27B, Qwen3.6-35B-A3B) sustain 5,400–5,900 tokens/second, whereas embedding models range from 14,700 tok/s (the largest, F2LLM-14B) to 4.3M tok/s (the smallest, mE5-small), a 2.5× to 736× advantage on identical hardware. With hardware and serving stack fixed, the measured gap reflects the different inference procedures: autoregressive decoding and a single encoder pass. At production scale (millions of documents and continuous ingestion), this throughput gap limits the practicality of LLM-based pipelines.

4.6 Ablation: Reduced Thinking

We re-evaluate Gemini 3 Flash with `reasoning_effort=low` and, for the open models, with reasoning disabled at the serving layer, cutting reasoning tokens by 54–96% on the retrieval tasks (Table 14; cross-family results in Figure 4b; details in Appendix C). Two findings stand

out. First, reduced reasoning preserves or improves all six Flash retrieval scores and four of six cross-family averages. Second, classification changes by less than one point on all tested tasks ($|\Delta| < 1.0$). The retrieval advantage largely persists under reduced reasoning budgets, indicating a limited role for additional test-time reasoning.

4.7 Ablation: Few-Shot Classification

To test whether labelled examples can close the classification gap, we evaluate Flash with 5 in-context examples per task (Table 15; details in Appendix C). Five-shot results are similar to or worse than the zero-shot scores on the small-label tasks. Note that the embedding kNN classifier uses the full labelled train set. The LLM prompt only gets five examples here, yet given these results, it seems unlikely more would help.

5 Discussion

In aggregate, the two paradigms tie on maximum performance. The LLM advantage is narrow and specific: reasoning-heavy retrieval, where bi-encoders score systematically low (Su et al., 2024). Pro leads on 5/6 retrieval datasets, showing a consistent advantage across the suite. Capability profiles show this split: LLMs lead on retrieval while embeddings provide balanced, often superior, coverage across the other four categories (Appendix D.1, Figure 8).

Viewed through the lens of information retrieval, our results reproduce the classic bi-encoder versus cross-encoder tradeoff at LLM scale (Reimers & Gurevych, 2019; Muennighoff, 2022). One design choice separates the four architectures we evaluate (Figure 5): how many documents the model is allowed to read jointly with the query. Embedding pipelines encode queries and documents independently: each document is processed once, offline, and each query requires only vector comparisons. A cross-encoder reranker reads one document at a time, an LLM listwise reranker reads the top-k shortlist at once, and an LLM in corpus-in-context mode is the extreme case, reading the entire corpus in a single forward pass. Cost follows that ordering because query-specific computation repeats for each query: reranking scales with the shortlist k, and corpus-in-context processing scales with the corpus N. Quality follows the same ordering, which is why reranking is a more economical way to add reasoning than placing the full corpus in context (§4.3).

Our retrieve-then-rerank experiment (§4.3) shows the same pattern in a production-style setting: a reranker helps most on reasoning-heavy retrieval and little on semantic retrieval. The same-hardware results attribute much of the quality–cost gap to the underlying inference procedures.

The classification pipelines receive different supervision. Embedding classifiers use kNN over the full labelled training distribution; the LLM receives label names, task descriptions, and up to five examples. Fine-grained tasks make this difference especially visible (§4.7). Leading embedding models are also contrastively trained on data overlapping these domains and label spaces, so part of the gap reflects in-domain fit rather than a general representational advantage. The comparison is therefore between deployment pipelines as practitioners meet them, not intrinsic ceilings: a classification-post-trained LLM would likely narrow the gap, and none was available among the frontier models we evaluate.

The thinking-token tax reveals a mismatch between default reasoning budgets and these tasks. Reasoning improves the overall Flash score, while lower reasoning budgets preserve retrieval quality for most evaluated models. Where direct reading comprehension suffices, the model reconsiders straightforward relevance judgements, making the default reasoning budget wasteful for most of these benchmark tasks (Lu et al., 2025). For practitioners, reducing reasoning effort preserves retrieval quality for most evaluated models while removing the largest component of LLM cost.

Deployment implications. These findings motivate a hybrid architecture: embedding models provide high-throughput candidate retrieval, followed by LLM reasoning over a shortlist (Figure 7). This division of labour supports retrieve-then-reason pipelines (Lewis

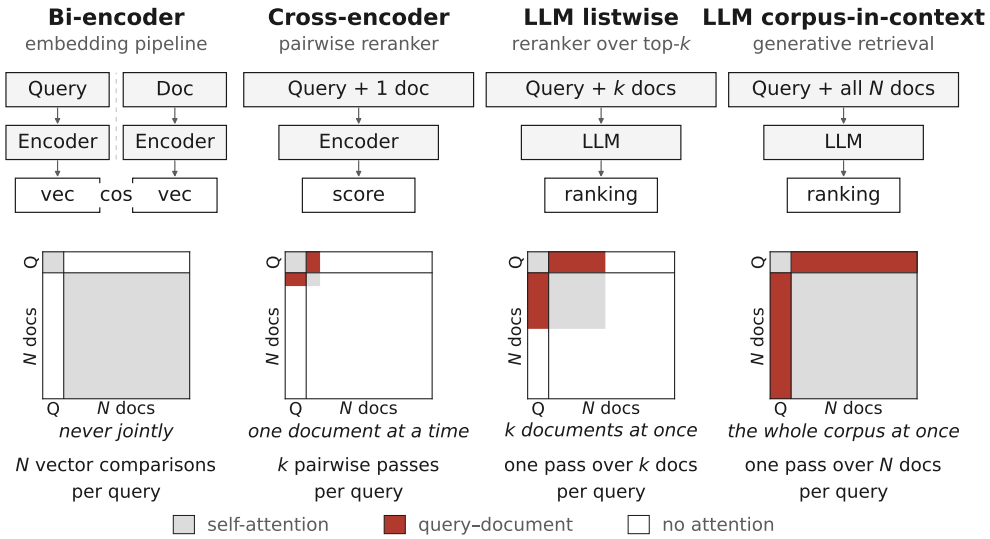

Figure 5: **How many documents each architecture reads jointly with the query.** *Top:* the pipeline. *Bottom:* its attention mask, drawn over the same N documents at the same scale in every panel so the four are directly comparable. Each row is a token and each column a token it may read. The red region is what one forward pass reads jointly with the query; it cannot be computed before the query arrives, so cost grows with it.

et al., 2020; Gao et al., 2024); our reranker results (§4.3) show where they help. Across the four non-retrieval categories, small-to-medium embedding models closely match the best LLM at a fraction of the cost. Hardware improvements will reduce the absolute cost of both paradigms. Their different inference procedures preserve an embedding throughput advantage for similarity, classification, and clustering workloads (§3.4). Accuracy-only leaderboards obscure large cost differences between similarly scored systems. We recommend reporting Pareto frontiers and significance tests alongside accuracy (Card et al., 2020; Dehghani et al., 2021).

Limitations. Our ten LLMs from six families are a snapshot of a fast-moving frontier; because MTEB(LLM) uses the MTEB framework, future models can be evaluated with the same pipeline. The corpus-in-context protocol places corpora of 82–415 documents entirely in the prompt, which is only feasible at small scale: production corpora require an indexed first stage, so the cost gap we report is a lower bound. We complement this setting with a bi-encoder, cross-encoder reranker, and LLM listwise reranker comparison on BEIR and BRIGHT (§4.3). Embeddings also benefit from post-hoc threshold optimisation in pair classification (MTEB convention) with no LLM analogue. Appendix G gives the full set.

6 Conclusion

We present MTEB(LLM), the first cost-aware comparison of LLM and embedding deployment pipelines across five MTEB task categories, evaluating ten LLMs from six families and 26 embedding models on 37 tasks. The two paradigms match in aggregate but differ by task: LLMs lead on reasoning-heavy retrieval, embeddings lead on classification, and the rest are tied. Comparable LLM quality comes with substantially higher cost and lower throughput.

Embedding pipelines therefore remain the cost-efficient default, with reasoning-intensive retrieval the clearest case for an LLM. Our retrieve-then-rerank results support a hybrid strategy: embeddings for candidate retrieval, LLMs for reasoning over the shortlist.

Pareto frontiers add deployment cost and throughput to accuracy-based rankings, and we encourage their wider use. MTEB(LLM) is released with MTEB-compatible code and datasets, so the evaluation can expand as new models appear.

Acknowledgments

We are extremely thankful to Laude Institute for supporting this work.

Reproducibility Statement

Code, raw results, and analysis scripts are released at https://github.com/embeddings-benchmark/embedders-dilemma in an implementation built on the MTEB framework. The LLM-specific datasets are hosted on Hugging Face under `mteb/llm-eval-*`, and each task pins its dataset to an exact revision. LLM prompt templates, schema validation logic, and token-usage extraction scripts are included in the release, and every figure and table in the paper regenerates from the released result files. The provided scripts reproduce the embedding throughput benchmarks on an NVIDIA H100 GPU. Detailed experimental configurations are described in §3 and Appendix I.

Ethics Statement

This work compares publicly available models on standard benchmarks and does not involve human subjects or private data. All evaluated datasets are publicly hosted and have been previously released for research purposes. Cost and throughput reporting may influence deployment decisions; we encourage practitioners to consider the environmental impact of LLM-based pipelines, including the carbon footprint of chain-of-thought inference, alongside the economic factors discussed in this paper. Reducing reasoning budgets can preserve performance on some tasks while lowering compute use. LLM assistance was used during the preparation of this paper for writing polish, phrasing refinement, and code formatting (e.g., matplotlib styling); no LLM was used to originate research ideas, generate evaluation data, produce plots, or evaluate model outputs. All scientific content, experimental design, and conclusions are the authors' own.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret (eds.), **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1051/.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In Steven Bethard, Marine Carpuat, Daniel Cer, David Jurgens, Preslav Nakov, and Torsten Zesch (eds.), *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 497–511, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1081. URL https://aclanthology.org/S16-1081/.

Mohammad Kalim Akram, Saba Sturua, Nastia Havriushenko, Quentin Herreros, Michael Günther, Maximilian Werk, and Han Xiao. jina-embeddings-v5-text: Task-targeted embedding distillation, 2026. URL https://arxiv.org/abs/2602.15547.

Anthropic. Claude Opus 4.6. Model card, Anthropic, 2026. URL https://www.anthropic.com/claude/opus.

Adnan El Assadi, Isaac Chung, Roman Solomatin, Niklas Muennighoff, and Kenneth Enevoldsen. Hume: Measuring the human-model performance gap in text embedding tasks, 2025. URL https://arxiv.org/abs/2510.10062.

Yauhen Babakhin, Radek Osmulski, Ronay Ak, Gabriel Moreira, Mengyao Xu, Benedikt Schifferer, Bo Liu, and Even Oldridge. Llama-embed-nemotron-8b: A universal text embedding model for multilingual and cross-lingual tasks, 2025. URL https://arxiv.org/abs/2511.07025.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large language models are secretly powerful text encoders, 2024. URL https://arxiv.org/abs/2404.05961.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification, 2019. URL https://arxiv.org/abs/1903.04561.

Martin Juan José Bucher and Marco Martini. Fine-tuned 'small' LLMs (still) significantly outperform zero-shot generative AI models in text classification, 2024. URL https://arxiv.org/abs/2406.08660.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 9263–9274, 2020. URL https://arxiv.org/abs/2010.06595.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 2020. URL https://arxiv.org/abs/2003.04807.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017. URL https://arxiv.org/abs/1708.00055.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2025. URL https://arxiv.org/abs/2402.03216.

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott A. Hale, David Jurgens, and Mattia Samory. SemEval-2022 task 8: Multilingual news article similarity. In Guy Emerson, Natalie Schluter, Gabriel Stanovsky, Ritesh Kumar, Alexis Palmer, Nathan Schneider, Siddharth Singh, and Shyam Ratan (eds.), *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pp. 1094–1106, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.semeval-1.155. URL https://aclanthology.org/2022.semeval-1.155/.

Chanyeol Choi, Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, and Jy yong Sohn. Linq-embed-mistral technical report, 2024. URL https://arxiv.org/abs/2412.03223.

DeepSeek-AI, Daya Guo, Dejian Yang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Mostafa Dehghani, Yi Tay, Alexey A. Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. The benchmark lottery, 2021. URL https://arxiv.org/abs/2107.07002.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, 2019. URL https://arxiv.org/abs/1810.04805.

Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. FQuAD: French question answering dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1193–1208, 2020. URL https://arxiv.org/abs/2002.06071.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/2958830.

Adnan El Assadi, Isaac Chung, Chenghao Xiao, Roman Solomatin, Animesh Jha, Rahul Chand, Silky Singh, Kaitlyn Wang, Ali Sartaz Khan, Marc Moussa Nasser, Sufen Fong, Pengfei He, Alan Xiao, Ayush Sunil Munot, Aditya Shrivastava, Artem Gazizov, Niklas Muennighoff, and Kenneth Enevoldsen. Maeb: Massive audio embedding benchmark, 2026a. URL https://arxiv.org/abs/2602.16008.

Adnan El Assadi, Roman Solomatin, Isaac Chung, Chenghao Xiao, Deep Shah, Manan Dey, Shriya Sudhakar, Zacharie Bugaud, Wissam Siblini, Ayush Sunil Munot, Yashwanth Devavarapu, Rakshitha Ireddi, Michelle Yang, Márton Kardos, Niklas Muennighoff, and Kenneth Enevoldsen. Mveb: Massive video embedding benchmark, 2026b. URL https://arxiv.org/abs/2606.14958.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzeminski, 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 Poświąta, 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. Mmteb: Massive multilingual text embedding benchmark, 2025. URL https://arxiv.org/abs/2502.13595.

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. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages, 2022. URL https://arxiv.org/abs/2204.08582.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Gemma Team et al. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In Satoshi Sekine, Kentaro Inui, Ido Dagan, Bill Dolan, Danilo Giampiccolo, and Bernardo Magnini (eds.), *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 1–9, Prague, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/W07-1401/.

In Gim, Guojun Chen, Seung seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. In *Proceedings of Machine Learning and Systems*, volume 6, 2024. URL https://arxiv.org/abs/2311.04934.

Alberto Andres Valdes Gonzalez. Cost-aware model selection for text classification: Multi-objective trade-offs between fine-tuned encoders and LLM prompting in production, 2026. URL https://arxiv.org/abs/2602.06370.

Aaron Grattafiori et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Gavin Correction, Gregory Bassett, Haggai Porat, Jason Livermore, Jesse Noss, Jonathan Noss, Kevin Ashley, Kevin D. Ashley, Li-Cheng Lan, Marshall Tyler, Neel Guha, Nikhil Patel, Rosamund Thalken, Thomas F. Gordon, Tomas Skorepa, Wendy Xu, and Zhaoyi Zhou. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2308.11462.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is ChatGPT to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*, 2023. URL https://arxiv.org/abs/2301.07597.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models, 2023. URL https://arxiv.org/abs/2307.16645.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020. URL https://arxiv.org/abs/2004.04906.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48, 2020. URL https://arxiv.org/abs/2004.12832.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1224–1234, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1126. URL https://aclanthology.org/D17-1126/.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2405.17428.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien M. R. Arnold, Vincent Perot, Siddharth Dalmia, Hexiang Hu, Xudong Lin, Panupong Pasupat, Aida Amini, Jeremy R. Cole, Sebastian Riedel, Iftekhar Naim, Ming-Wei Chang, and Kelvin Guu. Can long-context language models subsume retrieval, RAG, SQL, and more?, 2024. URL https://arxiv.org/abs/2406.13121.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, Xiaoqi Ren, Shanfeng Zhang, Daniel Salz, Michael Boratko, Jay Han, Blair Chen, Shuo Huang, Vikram Rao, Paul Suganthan, Feng Han, Andreas Doumanoglou, Nithi Gupta, Fedor Moiseev, Cathy Yip, Aashi Jain, Simon Baumgartner, Shahrokh Shahi, Frank Palma Gomez, Sandeep Mariserla, Min Choi, Parashar Shah, Sonam Goenka, Ke Chen, Ye Xia, Koert Chen, Sai Meher Karthik Duddu, Yichang Chen, Trevor Walker, Wenlei Zhou, Rakesh Ghiya, Zach Gleicher, Karan Gill, Zhe Dong, Mojtaba Seyedhosseini, Yunhsuan Sung, Raphael Hoffmann, and Tom Duerig. Gemini embedding: Generalizable embeddings from Gemini, 2025b. URL https://arxiv.org/abs/2503.07891.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020. URL https://arxiv.org/abs/2005.11401.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark, 2021. URL https://arxiv.org/abs/2008.09335.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning, 2023. URL https://arxiv.org/abs/2308.03281.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach, 2024. URL https://arxiv.org/abs/2407.16833.

Xuan Lu et al. Rethinking reasoning in document ranking: Why chain-of-thought falls short, 2025. URL https://arxiv.org/abs/2510.08985.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015/.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 216–223, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL https://aclanthology.org/L14-1314/.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Sfr-embedding-2: Advanced text embedding with multi-stage training, 2024. URL https://huggingface.co/Salesforce/SFR-Embedding-2_R.

Niklas Muennighoff. SGPT: GPT sentence embeddings for semantic search, 2022. URL https://arxiv.org/abs/2202.08904.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, 2023. URL https://arxiv.org/abs/2210.07316.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning, 2024. URL https://arxiv.org/abs/2402.09906.

Zhijie Nie, Zhangchi Feng, Mingxin Li, Cunwang Zhang, Yanzhao Zhang, Dingkun Long, and Richong Zhang. When text embedding meets large language model: A comprehensive survey, 2025. URL https://arxiv.org/abs/2412.09165.

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

OpenAI. GPT-4 technical report, 2023. URL https://arxiv.org/abs/2303.08774.

OpenAI. OpenAI o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 3982–3992, 2019. URL https://arxiv.org/abs/1908.10084.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020. URL https://arxiv.org/abs/1907.10597.

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. Adversarial domain adaptation for duplicate question detection. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1056–1063, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1131. URL https://aclanthology.org/D18-1131/.

Aaditya Singh et al. OpenAI GPT-5 system card, 2025. URL https://arxiv.org/abs/2601.03267.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. BIOSSES: A semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58, 2017.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2402.15449.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023. URL https://arxiv.org/abs/2212.09741.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval, 2024. URL https://arxiv.org/abs/2407.12883.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models, 2025. URL https://arxiv.org/abs/2503.16419.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents, 2023a. URL https://arxiv.org/abs/2304.09542.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. Text classification via large language models, 2023b. URL https://arxiv.org/abs/2305.08377.

Chongyang Tao, Tao Shen, Shen Gao, Junshuo Zhang, Zhen Li, Kai Hua, Wenpen Hu, Zhangwei Tao, and Shuai Ma. Llms are also effective embedding models: An in-depth overview. *ACM Transactions on Information Systems*, 2024.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2104.08663.

Søren Vejlgaard Holm, Lars Kai Hansen, and Martin Carsten Nielsen. Danoliteracy of generative large language models, March 2025. URL https://aclanthology.org/2025.nodalida-1.78/.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, et al. Embeddinggemma: Powerful and lightweight text representations, 2025. URL https://arxiv.org/abs/2509.20354.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2022. URL https://arxiv.org/abs/2212.03533.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models, 2024. URL https://arxiv.org/abs/2401.00368.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. URL https://arxiv.org/abs/2201.11903.

Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. Rank1: Test-time compute for reranking in information retrieval, 2025. URL https://arxiv.org/abs/2502.18418.

Chenghao Xiao, Isaac Chung, Imene Kerboua, Jamie Stirling, Xin Zhang, Márton Kardos, Roman Solomatin, Noura Al Moubayed, Kenneth Enevoldsen, and Niklas Muennighoff. Mieb: Massive image embedding benchmark, 2025. URL https://arxiv.org/abs/2504.10471.

An Yang et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel Campos. Arctic-embed 2.0: Multilingual retrieval without compromise, 2024. URL https://arxiv.org/abs/2412.04506.

Yanzhao Zhang et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025. URL https://arxiv.org/abs/2506.05176.

Ziyin Zhang, Zihan Liao, Hang Yu, Peng Di, and Rui Wang. F2LLM-v2: Inclusive, performant, and efficient embeddings for a multilingual world, 2026. URL https://arxiv.org/abs/2603.19223.

Xinping Zhao, Xinshuo Hu, Zifei Shan, Zetian Sun, Zhenyu Liu, Dongfang Li, Shaolin Ye, Xinyuan Wei, Qian Chen, Baotian Hu, Haofen Wang, Jun Yu, and Min Zhang. KaLM-Embedding-V2: Superior training techniques and data inspire a versatile embedding model, 2025. URL https://arxiv.org/abs/2506.20923.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.05685.

A Statistical Significance Testing

Bootstrap procedure. We implement a paired bootstrap test (Efron, 1979) comparing Gemini 3.1 Pro against the best embedding model in each comparison. The resampling unit is the task level: each bootstrap replicate draws MTEB(LLM) tasks with replacement from the full MTEB(LLM) task set, and the score difference is computed as the mean-of-means difference across the resampled task set. Task-level resampling accommodates the different metrics used across categories (accuracy, Spearman ρ, V-measure, average precision, and Recall@1). Equal task weighting matches the macro-average score.

We report unadjusted p-values for five planned category-level comparisons. Under a Bonferroni threshold of $\alpha = 0.01$, classification remains significant and retrieval falls above the threshold.

Table 2 reports the full results. The overall comparison (Pro vs. Octen-8B across MTEB(LLM)) yields $\Delta = +0.3$, 95% CI = $[-2.4, +3.1]$, $p = 0.85$, a statistical tie. Category-level tests identify two significant differences:

- **Classification:** embeddings win ($\Delta = -5.6$, CI = $[-9.2, -2.4]$, $p < 0.01$)
- **Retrieval:** LLMs win ($\Delta = +8.5$, CI = $[+0.2, +16.8]$, $p < 0.05$)

STS ($p = 0.74$), clustering ($p = 0.97$), and pair classification ($p = 0.49$) show no significant paradigm difference.

Comparison	Δ	**95% CI**	p	**Sig.**
Overall (Gemini 3.1 Pro vs. Octen-8B, MTEB(LLM))				
All tasks	+0.3	[−2.4, +3.1]	0.85	No
Per category (Gemini 3.1 Pro vs. best embedding)				
Retrieval	+8.5	[+0.2, +16.8]	<0.05	Yes
Clustering	−0.2	[−5.6, +5.1]	0.96	No
STS	−0.3	[−2.2, +1.8]	0.75	No
Pair Classification	−3.9	[−13.6, +5.9]	0.50	No
Classification	−5.6	[−9.2, −2.4]	<0.01	Yes

Table 2: **Statistical significance.** Paired bootstrap test (10,000 resamples, seed 42). Δ = Gemini 3.1 Pro − best embedding; significance at $\alpha = 0.05$. Pair classification uses Pro for consistency, although Flash scores higher.

B Evaluated Models and Tasks

This section lists the evaluated models, category- and task-level results, and benchmark tasks and metrics.

B.1 Model Overview

Table 3 lists all 36 evaluated models — 10 LLMs and 26 embedding models — with parameter counts, mean MTEB(LLM) scores, total benchmark costs, and paper references. Models range from mE5-small (118M parameters, $0.001) to Gemini 3.1 Pro ($154.14). Models without a dedicated paper are marked "–".

B.2 Full Per-Category Rankings

Table 4 compares Pro with the best embedding in each category. Table 5 gives per-category scores for all 36 models. Pro ranks first overall largely because of retrieval; on classification, it ranks below ten embedding models. The reasoning-capable Gemini models rank far above the non-reasoning Flash-Lite, which places 31st of 36 overall.

B.3 Per-Task Scores (Representative Models)

Table 6 reports individual task scores for 5 representative models (2 LLMs, 3 embeddings) across all MTEB(LLM) tasks, alongside the best embedding score per task. The scores vary substantially by task. For example, Pro is strongest on FQuAD retrieval, and SFR-2 is strongest on SprintDuplicateQuestions. This complementarity supports the hybrid-pipeline recommendation in §5.

B.4 Full Per-Task Scores for All Models

Tables 7–11 provide the complete score matrix for all 36 models across every MTEB(LLM) task, presented one table per category so that each fits the page upright. The tables share a common layout: models are rows, ordered by overall MTEB(LLM) score, with the ten

Model	Params	Score	Cost	Reference
LLM Models				
Gemini 3.1 Pro	–	**77.6**	$154.14	–
Gemini 3 Flash	–	75.2	$55.87	–
Qwen3.6-27B	–	73.9	$103.43	–
Qwen3.6-35B-A3B	–	73.3	$34.06	–
Kimi-K2.6	–	71.7	$110.70	–
DeepSeek-V4-Flash	–	68.5	$3.16	–
MiniMax-M2.7	–	68.2	$24.52	–
GLM-4.7	–	67.7	$63.31	–
Gemini 3.1 Flash Lite	–	64.5	$6.85	–
DeepSeek-R1	–	64.3	$57.38	–
Embedding Models (ranked by score)				
Octen-8B	7.6B	77.2	$0.108	–
Qwen3-E-8B	7.6B	77.0	$0.108	Zhang et al. 2025
Qwen3-E-4B	4.0B	75.9	$0.069	Zhang et al. 2025
Nemotron-8B	7.5B	75.3	$0.115	Lee et al. 2025a
KaLM-12B	11.8B	74.8	$0.158	Zhao et al. 2025
Jina-v5-S	596M	74.4	$0.034	Akram et al. 2026
SFR-2	7.1B	73.9	$0.136	Meng et al. 2024
Jina-v5-Nano	212M	73.9	$0.010	Akram et al. 2026
F2LLM-14B	14.0B	73.3	$0.215	Zhang et al. 2026
GTE-Qwen2-7B	7.1B	73.1	$0.089	Li et al. 2023
Linq-Mistral	7.1B	72.8	$0.138	Choi et al. 2024
Qwen3-E-0.6B	596M	72.5	$0.018	Zhang et al. 2025
EmbGemma-300M	308M	72.2	$0.008	Lee et al. 2025b
F2LLM-8B	7.6B	72.2	$0.122	Zhang et al. 2026
F2LLM-4B	4.0B	71.9	$0.079	Zhang et al. 2026
F2LLM-1.7B	1.7B	71.7	$0.030	Zhang et al. 2026
GTE-Qwen2-1.5B	1.5B	70.8	$0.024	Li et al. 2023
F2LLM-0.6B	596M	69.9	$0.017	Zhang et al. 2026
mE5-L-Inst	560M	69.7	$0.005	Wang et al. 2024
BGE-M3	568M	66.5	$0.005	Chen et al. 2025
mE5-L	560M	65.9	$0.005	Wang et al. 2022
Arctic-L-v2	568M	65.3	$0.005	Yu et al. 2024
mE5-B	278M	64.4	$0.002	Wang et al. 2022
mE5-S	118M	63.1	$0.001	Wang et al. 2022
E5-Mistral-7B	7.1B	52.2	$0.139	Wang et al. 2024
GritLM-7B	7.2B	51.2	$0.138	Muennighoff et al. 2024

Table 3: **Complete model listing** with mean (macro) scores across 37 MTEB(LLM) tasks, total benchmark costs, and references. LLM costs reflect actual API usage; embedding costs from H100 throughput benchmarking ($2.49/hr).

Category	Gemini 3.1 Pro	Gemini 3 Flash	Best Emb.	Best Model	Δ
Retrieval (6)	**64.5**	52.3	56.0	Octen-8B	+8.5
Clustering (9)	66.6	65.7	**66.7**	SFR-2	−0.2
STS (10)	88.5	87.6	**88.8**	Qwen3-E-4B	−0.3
PairCls (4)	83.2	86.3	**87.1**	KaLM-12B	−3.9
Classification (8)	85.2	84.1	**90.8**	SFR-2	−5.6

Table 4: **Category-level performance** (best LLM vs. best embedding per category). Δ = best LLM − best embedding; bold = winner; task counts in parentheses.

LLMs listed first and the 26 embedding models below; the category's tasks are columns, with its metric given in the caption. The final column gives each model's mean over the category, matching the corresponding column of Table 1. Bold marks the best value in a column, whether a single task or the category mean. Because rows are ordered by overall score rather than by category, this column is not monotonic: it shows where a model over- or under-performs its overall rank. These tables extend Table 6 to all models.

Rank	Model	Type	Cls (8)	Clust (9)	STS (10)	PairCls (4)	Retr (6)	Overall
1	Gemini 3.1 Pro	LLM	85.2	66.6	88.5	83.2	**64.5**	**77.6**
2	Octen-8B	Emb	90.1	65.1	88.7	86.1	56.0	77.2
3	Qwen3-E-8B	Emb	90.1	65.9	88.5	86.5	54.2	77.0
4	Qwen3-E-4B	Emb	89.3	64.6	**88.8**	86.5	50.3	75.9
5	Nemotron-8B	Emb	84.8	64.1	86.2	86.7	54.7	75.3
6	Gemini 3 Flash	LLM	84.1	65.7	87.6	86.3	52.3	75.2
7	KaLM-12B	Emb	88.8	63.4	84.6	**87.1**	49.9	74.8
8	Jina-v5-S	Emb	90.4	61.5	86.9	85.1	48.0	74.4
9	SFR-2	Emb	**90.8**	**66.7**	77.9	85.3	48.8	73.9
10	Jina-v5-Nano	Emb	89.6	60.4	87.0	85.0	47.4	73.9
11	Qwen3.6-27B	LLM	84.8	53.6	84.9	83.6	62.4	73.9
12	F2LLM-14B	Emb	77.7	66.5	84.1	86.3	52.1	73.3
13	Qwen3.6-35B-A3B	LLM	83.8	55.3	84.0	82.9	60.4	73.3
14	GTE-Qwen2-7B	Emb	86.7	65.2	81.6	86.0	45.8	73.1
15	Linq-Mistral	Emb	82.5	61.3	83.4	86.1	51.0	72.8
16	Qwen3-E-0.6B	Emb	85.4	60.9	85.7	86.1	44.1	72.5
17	EmbGemma-300M	Emb	86.5	59.3	82.1	86.0	47.3	72.2
18	F2LLM-8B	Emb	75.2	65.8	83.9	86.1	50.0	72.2
19	F2LLM-4B	Emb	75.3	64.7	83.6	85.9	49.9	71.9
20	Kimi-K2.6	LLM	84.5	55.4	83.9	78.0	56.7	71.7
21	F2LLM-1.7B	Emb	74.7	65.1	83.8	86.1	48.5	71.7
22	GTE-Qwen2-1.5B	Emb	83.3	60.0	80.3	86.6	43.9	70.8
23	F2LLM-0.6B	Emb	72.5	62.9	83.1	85.8	45.4	69.9
24	mE5-L-Inst	Emb	74.4	59.1	83.9	86.1	44.8	69.7
25	DeepSeek-V4-Flash	LLM	81.9	43.7	82.2	81.6	53.3	68.5
26	MiniMax-M2.7	LLM	80.9	48.6	81.0	82.8	48.0	68.2
27	GLM-4.7	LLM	84.2	36.5	83.5	79.7	54.4	67.7
28	BGE-M3	Emb	76.2	45.9	80.4	85.9	44.3	66.5
29	mE5-L	Emb	73.3	48.3	80.3	83.9	43.6	65.9
30	Arctic-L-v2	Emb	71.4	49.7	77.2	83.9	44.6	65.3
31	Gemini 3.1 Flash Lite	LLM	82.9	21.7	85.1	83.7	49.0	64.5
32	mE5-B	Emb	71.9	47.5	79.2	84.3	38.9	64.4
33	DeepSeek-R1	LLM	82.5	31.3	83.7	79.5	44.7	64.3
34	mE5-S	Emb	69.8	47.7	78.5	83.9	35.8	63.1
35	E5-Mistral-7B	Emb	70.0	47.9	49.8	72.9	20.6	52.2
36	GritLM-7B	Emb	68.6	48.9	55.1	77.3	6.1	51.2

Table 5: **Complete per-category results** for all 36 complete models, ranked by overall (macro) score. Bold = best in column.

B.5 Benchmark Task Suite

Table 12 lists all 37 tasks with languages, sample counts, and source citations. All are held-out subsets (seed = 42) derived from MTEB and MMTEB tasks, hosted on Hugging Face under `mteb/llm-eval-*`; the exact dataset path and pinned revision for each task are listed in the released repository. Table 13 provides per-task token budgets for cost estimation, broken down by split.

C Ablation Studies

This section details the reduced-thinking and few-shot experiments and provides a per-task retrieval breakdown.

C.1 Reduced-Thinking Ablation Details

Our reduced-thinking ablation uses Gemini 3 Flash with `reasoning_effort=low`, which instructs the model to minimise internal chain-of-thought reasoning. We evaluate on all 6 retrieval tasks and 3 representative classification tasks (IMDB, Banking77, ToxicConversations).

Task	Cat.	Gemini 3.1 Pro	Gemini 3 Flash	Qwen3.6-27B	Octen-8B	Qwen3-E-8B	Best Emb.
Classification (Accuracy)							
AmazonCounterfactualClassification	Cls	84.8	81.2	90.1	**93.0**	92.9	**93.0**
Banking77Classification	Cls	85.0	83.1	80.5	87.3	87.2	**91.6**
ImdbClassification	Cls	**98.0**	97.6	97.4	97.7	97.8	97.8
MTOPDomainClassification	Cls	95.5	96.4	96.9	98.2	98.1	**99.2**
MassiveIntentClassification	Cls	84.9	85.4	84.6	85.7	85.8	**88.9**
MassiveScenarioClassification	Cls	79.4	75.9	76.4	89.3	89.4	**93.0**
ToxicConversationsClassification	Cls	89.6	90.0	89.2	91.9	92.0	**94.4**
TweetSentimentExtractionClassification	Cls	64.2	63.0	63.2	77.9	77.8	**78.9**
Clustering (V-measure)							
ArxivClusteringP2P	Clust	60.1	60.3	54.5	61.9	62.5	**64.1**
ArxivClusteringS2S	Clust	**62.5**	60.1	41.6	60.5	61.1	61.5
BiorxivClusteringP2PV2	Clust	62.5	60.8	49.9	61.6	65.3	**77.3**
MedrxivClusteringP2PV2	Clust	52.5	50.6	48.0	55.9	55.9	**60.7**
MedrxivClusteringS2SV2	Clust	54.0	50.3	42.7	53.3	55.2	**57.5**
RedditClusteringP2P	Clust	**93.9**	90.5	60.1	78.7	79.8	80.8
StackExchangeClusteringP2PV2	Clust	42.9	49.0	46.1	59.7	59.5	**60.7**
StackExchangeClusteringV2	Clust	**91.8**	90.2	73.6	85.8	86.3	86.3
TwentyNewsgroupsClusteringV2	Clust	78.7	**79.9**	66.3	68.5	67.6	73.8
STS (Spearman ρ)							
BIOSSES	STS	**89.6**	88.7	85.0	84.4	82.0	88.8
SICKR	STS	86.1	86.8	76.4	87.9	88.4	**91.1**
STS12	STS	82.6	79.5	77.1	87.3	87.3	**87.4**
STS13	STS	91.9	91.5	90.6	93.9	94.0	**94.6**
STS14	STS	90.5	89.3	88.0	90.9	91.0	**91.4**
STS15	STS	94.4	93.9	91.8	94.3	94.2	**94.5**
STS16	STS	89.0	88.5	87.1	92.5	92.5	**92.9**
STS17	STS	**94.2**	94.1	91.8	93.6	93.7	93.7
STS22v2	STS	**73.6**	71.7	72.0	69.5	68.9	69.5
STSBenchmark	STS	92.6	91.7	89.6	93.2	93.4	**94.9**
PairClassification (Avg. Precision)							
LegalBenchPC	PairCls	72.2	**77.4**	71.6	70.0	70.4	73.6
RTE3PC	PairCls	95.4	**96.7**	92.9	85.4	85.7	85.7
SprintDuplicateQuestionsPC	PairCls	82.2	83.4	84.6	99.4	99.6	**99.8**
TwitterURLCorpusPC	PairCls	82.8	87.6	85.2	89.4	**90.2**	**90.2**
Retrieval (Recall@1)							
AILAStatutes	Retr	14.5	5.7	14.2	**23.2**	21.8	**23.2**
FQuADRetrieval	Retr	92.0	88.0	**96.0**	69.0	68.0	72.0
HC3FinanceRetrieval	Retr	**71.0**	60.0	57.0	66.0	59.0	67.0
LegalBenchConsumerContractsQA	Retr	86.0	79.0	**88.0**	67.0	70.0	70.0
PublicHealthQA	Retr	**90.0**	49.0	86.0	81.0	78.0	85.0
TwitterHjerneRetrieval	Retr	**33.4**	32.4	33.0	29.9	28.6	31.5

Table 6: **Per-task scores** for representative models across all 37 MTEB(LLM) tasks. Bold = best in row (incl. best embedding). Metric per category in italics.

The reduction differs by task type: 54–94% on retrieval and 12–30% on classification. This suggests the model adaptively allocates more reasoning to retrieval (where documents must be read and compared) than classification (where the decision is more immediate).

Reduced reasoning preserves or improves all six Flash retrieval scores and four of six cross-family averages (Figure 4b). We hypothesise that excessive reasoning introduces second-guessing: the model considers multiple interpretations of a document's relevance when a more direct reading would suffice.

On classification, all three tasks show $|\Delta| < 1.0$, indicating no measurable benefit from additional reasoning in this ablation. This result is consistent with the supervision asymmetry between kNN and zero-shot classification.

C.2 Per-Task Retrieval Analysis

The six retrieval tasks in our benchmark span qualitatively different retrieval challenges; per-task scores for all 36 models are in Table 11. We provide a per-task breakdown of the aggregate retrieval advantage (+8.5) reported in §4.2. Pro leads five retrieval tasks, and Octen-8B leads AILAStatutes.

FQuAD (French reading-comprehension QA). Answering requires reading French passages in context and matching the supporting one. Pro (92.0) leads the best embedding (72.0) by a wide margin, the largest LLM advantage in the suite.

Model	AmazonCF	Banking77	IMDB	MTOPDomain	MassiveIntent	MassiveScenario	ToxicConvs	TweetSent	Mean
LLMs									
Gemini 3.1 Pro	84.8	85.0	**98.0**	95.5	84.9	79.4	89.6	64.2	85.2
Gemini 3 Flash	81.2	83.1	97.6	96.4	85.4	75.9	90.0	63.0	84.1
Qwen3.6-27B	90.1	80.5	97.4	96.9	84.6	76.4	89.2	63.2	84.8
Qwen3.6-35B-A3B	88.7	77.8	97.4	96.8	79.8	76.6	91.0	62.4	83.8
Kimi-K2.6	89.3	82.8	97.4	96.6	83.2	76.5	88.8	61.6	84.5
DeepSeek-V4-Flash	82.4	78.5	96.8	95.8	82.4	77.5	80.4	61.2	81.9
MiniMax-M2.7	83.6	78.4	97.0	95.2	75.1	67.0	87.6	63.0	80.9
GLM-4.7	88.7	80.4	97.2	96.8	83.0	77.1	87.6	63.2	84.2
Gemini 3.1 FLite	79.7	78.7	97.6	95.9	83.2	76.2	88.6	63.4	82.9
DeepSeek-R1	86.9	78.9	97.0	96.1	82.7	76.6	78.2	63.4	82.5
Embedding models									
Octen-8B	**93.0**	87.3	97.7	98.2	85.7	89.3	91.9	77.9	90.1
Qwen3-E-8B	92.9	87.2	97.8	98.1	85.8	89.4	92.0	77.8	90.1
Qwen3-E-4B	92.6	86.3	97.4	97.6	85.2	86.0	92.3	77.1	89.3
Nemotron-8B	83.8	83.6	97.1	96.5	83.5	83.5	88.3	62.1	84.8
KaLM-12B	90.4	87.7	96.3	98.5	84.6	86.7	90.2	76.1	88.8
Jina-v5-S	91.8	91.5	95.9	**99.2**	**88.9**	92.6	**94.4**	68.5	90.4
SFR-2	92.7	90.1	97.6	98.3	86.3	90.5	91.9	**78.9**	**90.8**
Jina-v5-Nano	91.3	90.2	95.5	98.0	88.4	**93.0**	94.4	66.3	89.6
F2LLM-14B	64.1	84.7	91.2	98.8	77.6	89.4	58.7	56.7	77.7
GTE-Qwen2-7B	86.8	84.9	96.7	98.0	83.7	85.7	88.2	70.0	86.7
Linq-Mistral	83.9	87.7	94.9	97.0	82.7	84.5	70.2	59.2	82.5
Qwen3-E-0.6B	90.5	80.8	96.3	95.8	80.1	83.8	81.9	74.3	85.4
EmbGemma-300M	89.6	**91.6**	91.9	98.6	85.6	91.6	83.4	59.8	86.5
F2LLM-8B	63.1	83.6	87.3	99.0	72.2	85.8	59.3	51.4	75.2
F2LLM-4B	61.3	82.2	86.4	99.0	72.3	86.5	60.7	54.2	75.3
F2LLM-1.7B	61.2	77.6	83.4	98.3	74.6	87.5	61.1	54.3	74.7
GTE-Qwen2-1.5B	81.6	79.7	95.8	95.3	77.9	79.8	84.9	71.3	83.3
F2LLM-0.6B	59.5	73.8	79.7	97.3	73.4	87.3	59.7	49.0	72.5
mE5-L-Inst	69.1	77.4	96.4	91.6	70.0	69.1	64.8	56.8	74.4
BGE-M3	75.4	82.4	87.9	93.5	72.2	76.6	64.8	56.9	76.2
mE5-L	79.1	75.3	89.2	91.4	68.7	72.8	58.1	51.8	73.3
Arctic-L-v2	63.6	82.3	73.4	93.5	72.2	76.0	61.5	48.8	71.4
mE5-B	79.6	73.7	85.5	90.6	66.1	71.3	57.1	51.6	71.9
mE5-S	71.8	70.6	79.7	89.1	65.6	69.1	59.4	52.9	69.8
E5-Mistral-7B	71.8	71.4	91.6	79.3	65.8	73.4	62.2	44.9	70.0
GritLM-7B	74.7	70.6	79.0	84.9	65.2	68.3	61.4	45.0	68.6

Table 7: **Per-task scores: Classification (Accuracy).**

Model	BIOSSES	SICK-R	STS12	STS13	STS14	STS15	STS16	STS17	STS22v2	STSBench	Mean
LLMs											
Gemini 3.1 Pro	89.6	86.1	82.6	91.9	90.5	94.4	89.0	**94.2**	**73.6**	92.6	88.5
Gemini 3 Flash	88.7	86.8	79.5	91.5	89.3	93.9	88.5	94.1	71.7	91.7	87.6
Qwen3.6-27B	85.0	76.4	77.1	90.6	88.0	91.8	87.1	91.8	72.0	89.6	84.9
Qwen3.6-35B-A3B	84.0	75.5	75.1	89.6	85.1	91.4	85.1	92.7	73.4	87.8	84.0
Kimi-K2.6	86.4	82.1	72.2	89.3	85.1	91.1	83.9	90.1	70.9	87.8	83.9
DeepSeek-V4-Flash	78.5	75.3	75.6	87.3	84.5	89.2	82.4	91.5	69.2	87.9	82.2
MiniMax-M2.7	87.7	79.7	65.4	86.5	81.9	88.4	81.1	87.1	66.7	85.7	81.0
GLM-4.7	82.3	76.0	73.1	88.5	85.7	91.9	84.9	90.4	73.2	88.5	83.5
Gemini 3.1 FLite	**89.8**	78.0	76.0	90.5	86.3	91.4	86.5	91.4	71.2	90.3	85.1
DeepSeek-R1	85.0	79.7	71.9	88.7	85.6	89.9	84.2	91.5	71.4	89.2	83.7
Embedding models											
Octen-8B	84.4	87.9	87.3	93.9	90.9	94.3	92.5	93.6	69.5	93.2	88.7
Qwen3-E-8B	82.0	88.4	87.3	94.0	91.0	94.2	92.5	93.7	68.9	93.4	88.5
Qwen3-E-4B	84.8	87.7	**87.4**	**94.6**	**91.4**	**94.5**	**92.9**	93.2	67.7	93.7	**88.8**
Nemotron-8B	86.3	85.4	83.4	91.5	87.9	92.1	88.7	91.4	64.3	90.8	86.2
KaLM-12B	87.0	81.9	81.4	89.2	85.9	90.6	86.7	89.0	65.1	89.1	84.6
Jina-v5-S	85.3	90.4	86.6	89.2	88.9	92.6	87.4	89.1	65.0	**94.9**	86.9
SFR-2	87.5	76.7	74.6	81.4	80.6	87.5	84.6	85.0	38.2	82.6	77.9
Jina-v5-Nano	87.4	**91.1**	86.2	89.9	89.2	93.1	85.4	88.4	64.9	94.2	87.0
F2LLM-14B	88.2	82.8	82.4	86.9	84.8	91.3	85.6	88.9	63.4	87.0	84.1
GTE-Qwen2-7B	83.0	79.7	78.1	88.6	83.4	89.5	85.5	87.1	54.7	86.2	81.6
Linq-Mistral	86.1	84.4	77.6	87.8	84.1	91.2	87.4	89.9	56.1	89.0	83.4
Qwen3-E-0.6B	84.6	84.6	83.6	92.1	87.0	91.7	89.7	87.9	64.6	91.3	85.7
EmbGemma-300M	83.2	81.9	79.1	85.5	84.5	89.2	84.7	86.6	57.3	88.8	82.1
F2LLM-8B	88.8	82.4	81.3	86.6	83.8	91.3	85.4	89.3	63.2	86.5	83.9
F2LLM-4B	87.5	81.9	82.3	86.1	83.8	91.5	85.2	87.9	62.8	87.1	83.6
F2LLM-1.7B	88.5	81.3	82.6	86.7	84.5	91.3	85.2	87.8	62.9	87.6	83.8
GTE-Qwen2-1.5B	81.4	80.7	72.9	84.4	81.7	88.7	84.6	86.1	56.6	85.6	80.3
F2LLM-0.6B	87.8	80.0	82.7	86.7	84.1	90.6	84.9	84.0	62.3	87.6	83.1
mE5-L-Inst	87.0	81.8	83.5	86.8	85.1	91.8	86.6	87.0	60.8	89.0	83.9
BGE-M3	83.4	79.3	79.8	78.6	80.7	88.3	84.7	82.3	60.5	86.0	80.4
mE5-L	84.6	80.5	80.9	75.6	78.7	89.7	83.6	85.4	58.5	85.8	80.3
Arctic-L-v2	87.2	72.5	71.6	79.6	76.0	83.9	83.2	79.4	59.6	78.4	77.2
mE5-B	86.2	78.9	79.1	75.9	79.2	88.7	82.0	81.6	56.0	84.5	79.2
mE5-S	84.4	78.0	78.3	77.9	77.9	87.9	82.6	77.0	56.8	84.2	78.5
E5-Mistral-7B	54.7	55.7	45.9	52.1	43.4	50.2	79.0	62.5	-21.4	75.5	49.8
GritLM-7B	65.3	60.5	51.3	61.1	56.3	69.3	66.1	36.0	31.6	52.9	55.1

Table 8: **Per-task scores: STS (Spearman ρ).**

Model	ArxivP2P	ArxivS2S	BioP2P	MedP2P	MedS2S	Reddit	SE-P2P	SE-Cl	20News	Mean
LLMs										
Gemini 3.1 Pro	60.1	**62.5**	62.5	52.5	54.0	**93.9**	42.9	**91.8**	78.7	66.6
Gemini 3 Flash	60.3	60.1	60.8	50.6	50.3	90.5	49.0	90.2	**79.9**	65.7
Qwen3.6-27B	54.5	41.6	49.9	48.0	42.7	60.1	46.1	73.6	66.3	53.6
Qwen3.6-35B-A3B	48.5	60.4	53.2	49.4	46.2	69.2	42.2	72.7	55.7	55.3
Kimi-K2.6	52.6	44.2	51.9	50.6	39.1	57.4	52.1	81.8	68.9	55.4
DeepSeek-V4-Flash	37.4	45.5	29.3	27.0	40.9	55.2	40.3	53.5	64.4	43.7
MiniMax-M2.7	36.4	40.6	38.3	39.8	42.4	67.4	39.4	75.1	57.9	48.6
GLM-4.7	23.6	28.6	33.2	37.5	37.8	40.6	37.9	49.5	39.8	36.5
Gemini 3.1 FLite	14.7	18.9	17.4	19.7	21.6	28.5	21.5	25.8	27.2	21.7
DeepSeek-R1	22.0	25.3	32.4	31.7	33.4	26.6	33.1	42.1	34.9	31.3
Embedding models										
Octen-8B	61.9	60.5	61.6	55.9	53.3	78.7	59.7	85.8	68.5	65.1
Qwen3-E-8B	62.5	61.1	65.3	55.9	55.2	79.8	59.5	86.3	67.6	65.9
Qwen3-E-4B	62.6	60.3	64.0	55.5	54.1	77.1	58.8	83.9	65.3	64.6
Nemotron-8B	59.2	59.1	65.2	52.9	52.6	79.7	58.7	84.6	65.2	64.1
KaLM-12B	61.6	57.4	64.1	54.7	53.2	76.1	58.5	83.3	62.0	63.4
Jina-v5-S	60.0	57.5	63.6	53.7	52.6	70.5	58.9	76.8	59.9	61.5
SFR-2	61.8	59.6	65.6	58.9	**57.5**	80.8	59.3	83.2	73.8	**66.7**
Jina-v5-Nano	59.1	55.4	61.6	54.6	52.1	67.4	57.3	76.8	59.7	60.4
F2LLM-14B	63.1	60.6	75.6	57.7	56.7	77.9	57.2	81.1	68.8	66.5
GTE-Qwen2-7B	**64.1**	61.5	66.7	57.0	54.1	80.5	**60.7**	84.5	58.1	65.2
Linq-Mistral	58.0	55.7	61.3	51.5	52.8	77.0	56.2	78.4	61.1	61.3
Qwen3-E-0.6B	60.2	57.6	61.8	52.9	51.1	69.9	56.0	77.3	61.7	60.9
EmbGemma-300M	57.9	53.9	62.8	54.4	51.0	75.9	51.9	73.7	51.8	59.3
F2LLM-8B	61.7	60.9	75.5	59.2	54.4	75.3	58.2	81.2	65.8	65.8
F2LLM-4B	61.3	59.2	74.2	59.8	54.7	74.4	55.2	79.9	63.9	64.7
F2LLM-1.7B	62.6	59.6	**77.3**	**60.7**	56.1	74.4	55.2	75.5	64.2	65.1
GTE-Qwen2-1.5B	58.5	57.5	62.1	53.2	52.3	71.1	52.3	73.6	59.1	60.0
F2LLM-0.6B	61.7	58.6	71.1	58.7	54.5	69.7	56.0	76.0	59.9	62.9
mE5-L-Inst	56.0	53.0	57.8	51.9	51.2	71.7	53.3	76.9	60.2	59.1
BGE-M3	41.7	32.4	49.7	43.1	40.7	61.2	42.6	56.3	45.5	45.9
mE5-L	45.8	39.7	52.7	43.2	43.9	65.6	43.7	56.4	43.9	48.3
Arctic-L-v2	48.1	38.4	48.7	46.5	44.0	65.2	46.6	61.7	48.3	49.7
mE5-B	44.9	40.2	46.2	43.1	44.3	64.8	42.6	57.6	43.4	47.5
mE5-S	42.5	39.9	48.2	45.5	44.3	65.5	41.9	58.0	43.2	47.7
E5-Mistral-7B	51.2	44.6	49.9	45.6	40.0	54.7	37.3	65.3	42.4	47.9
GritLM-7B	55.7	40.4	51.7	46.5	42.7	63.9	43.9	59.8	35.5	48.9

Table 9: **Per-task scores: Clustering (V-measure).**

Model	LegalPC	RTE3	SprintDup	TwtURL	Mean
LLMs					
Gemini 3.1 Pro	72.2	95.4	82.2	82.8	83.2
Gemini 3 Flash	**77.4**	**96.7**	83.4	87.6	86.3
Qwen3.6-27B	71.6	92.9	84.6	85.2	83.6
Qwen3.6-35B-A3B	69.4	93.6	85.2	83.4	82.9
Kimi-K2.6	69.2	88.8	78.8	75.4	78.0
DeepSeek-V4-Flash	67.2	90.4	86.6	82.0	81.6
MiniMax-M2.7	71.8	91.5	82.8	85.0	82.8
GLM-4.7	69.6	90.0	82.2	76.8	79.7
Gemini 3.1 FLite	72.4	95.0	86.6	80.8	83.7
DeepSeek-R1	68.8	90.0	79.6	79.6	79.5
Embedding models					
Octen-8B	70.0	85.4	99.4	89.4	86.1
Qwen3-E-8B	70.4	85.7	99.6	**90.2**	86.5
Qwen3-E-4B	72.0	84.8	**99.8**	89.4	86.5
Nemotron-8B	72.6	84.8	99.4	89.8	86.7
KaLM-12B	73.6	85.2	**99.8**	89.6	**87.1**
Jina-v5-S	68.0	85.2	99.6	87.4	85.1
SFR-2	67.4	84.8	**99.8**	89.0	85.3
Jina-v5-Nano	68.8	85.0	99.4	86.6	85.0
F2LLM-14B	72.0	84.8	99.6	88.6	86.3
GTE-Qwen2-7B	71.2	85.2	99.6	88.0	86.0
Linq-Mistral	71.0	84.8	**99.8**	88.6	86.1
Qwen3-E-0.6B	70.4	84.8	**99.8**	89.4	86.1
EmbGemma-300M	71.0	84.8	99.6	88.6	86.0
F2LLM-8B	72.0	84.8	99.6	88.0	86.1
F2LLM-4B	70.6	84.8	**99.8**	88.4	85.9
F2LLM-1.7B	71.0	84.8	99.6	89.0	86.1
GTE-Qwen2-1.5B	73.0	84.8	99.4	89.2	86.6
F2LLM-0.6B	70.8	85.0	99.4	88.0	85.8
mE5-L-Inst	70.8	84.8	99.6	89.2	86.1
BGE-M3	71.6	84.8	99.6	87.6	85.9
mE5-L	64.0	85.0	98.4	88.2	83.9
Arctic-L-v2	64.2	84.8	99.6	86.8	83.9
mE5-B	65.6	84.8	98.2	88.6	84.3
mE5-S	62.2	85.2	99.2	89.0	83.9
E5-Mistral-7B	66.0	84.8	73.4	67.4	72.9
GritLM-7B	56.8	84.8	92.0	75.4	77.3

Table 10: **Per-task scores: Pair Classification (AP / Accuracy).**

Model	AILA	FQuAD	HC3Fin	ConsumerQA	PubHealth	TwtHjerne	Mean
LLMs							
Gemini 3.1 Pro	14.5	92.0	**71.0**	86.0	**90.0**	**33.4**	**64.5**
Gemini 3 Flash	5.7	88.0	60.0	79.0	49.0	32.4	52.3
Qwen3.6-27B	14.2	**96.0**	57.0	**88.0**	86.0	33.0	62.4
Qwen3.6-35B-A3B	12.9	95.0	55.0	86.0	83.0	30.5	60.4
Kimi-K2.6	13.2	91.0	56.0	80.0	67.0	33.0	56.7
DeepSeek-V4-Flash	13.3	83.0	50.0	76.0	64.0	**33.4**	53.3
MiniMax-M2.7	9.7	82.0	42.0	67.0	59.0	28.2	48.0
GLM-4.7	13.0	78.0	56.0	78.0	70.0	31.6	54.4
Gemini 3.1 FLite	10.2	84.0	46.0	77.0	45.0	31.9	49.0
DeepSeek-R1	11.9	57.0	43.0	62.0	64.0	30.6	44.7
Embedding models							
Octen-8B	**23.2**	69.0	66.0	67.0	81.0	29.9	56.0
Qwen3-E-8B	21.8	68.0	59.0	70.0	78.0	28.6	54.2
Qwen3-E-4B	21.0	61.0	56.0	64.0	75.0	24.8	50.3
Nemotron-8B	9.1	72.0	67.0	70.0	80.0	29.9	54.7
KaLM-12B	10.3	70.0	55.0	63.0	71.0	30.2	49.9
Jina-v5-S	14.7	64.0	48.0	59.0	77.0	25.5	48.0
SFR-2	11.2	67.0	57.0	56.0	74.0	27.6	48.8
Jina-v5-Nano	12.6	65.0	44.0	61.0	74.0	27.6	47.4
F2LLM-14B	12.2	63.0	67.0	54.0	85.0	31.5	52.1
GTE-Qwen2-7B	9.1	64.0	51.0	55.0	77.0	19.0	45.8
Linq-Mistral	10.6	68.0	54.0	66.0	78.0	29.1	51.0
Qwen3-E-0.6B	20.2	55.0	36.0	63.0	68.0	22.4	44.1
EmbGemma-300M	5.5	72.0	42.0	62.0	78.0	24.2	47.3
F2LLM-8B	8.9	63.0	63.0	51.0	85.0	28.9	50.0
F2LLM-4B	8.1	64.0	62.0	53.0	83.0	29.4	49.9
F2LLM-1.7B	5.4	66.0	55.0	54.0	82.0	28.8	48.5
GTE-Qwen2-1.5B	5.3	58.0	44.0	59.0	76.0	21.2	43.9
F2LLM-0.6B	3.6	56.0	52.0	54.0	80.0	26.7	45.4
mE5-L-Inst	6.5	68.0	44.0	50.0	73.0	27.6	44.8
BGE-M3	7.4	62.0	40.0	58.0	74.0	24.2	44.3
mE5-L	4.9	69.0	40.0	51.0	73.0	23.9	43.6
Arctic-L-v2	3.3	61.0	42.0	59.0	76.0	26.1	44.6
mE5-B	3.2	64.0	28.0	47.0	68.0	23.5	38.9
mE5-S	3.6	57.0	26.0	43.0	64.0	20.9	35.8
E5-Mistral-7B	2.7	29.0	21.0	17.0	47.0	7.1	20.6
GritLM-7B	1.0	1.0	3.0	8.0	18.0	5.3	6.1

Table 11: **Per-task scores: Retrieval (Recall@1).**

Task	Lang.	N	Cls.	Metric	Source
Classification (8 tasks)					
ImdbCls	en	500	2	Acc.	Maas et al. (2011)
Banking77Cls	en	3k	77	Acc.	Casanueva et al. (2020)
AmazonCounterfactualCls	en, de, ja	809	2	Acc.	O'Neill et al. (2021)
MTOPDomainCls	en, de, fr	2k	11	Acc.	Li et al. (2021)
MassiveIntentCls	en, de, fr, ja	4k	60	Acc.	FitzGerald et al. (2022)
MassiveScenarioCls	en, de, fr, ja	3k	18	Acc.	FitzGerald et al. (2022)
ToxicConversationsCls	en	500	2	Acc.	Borkan et al. (2019)
TweetSentimentCls	en	500	3	Acc.	Muennighoff et al. (2023)
Semantic Textual Similarity (10 tasks)					
STSBenchmark	en	500	–	Spearman	Cer et al. (2017)
SICK-R	en	500	–	Spearman	Marelli et al. (2014)
STS12	en	500	–	Spearman	Agirre et al. (2012)
STS13	en	500	–	Spearman	Agirre et al. (2012)
STS14	en	500	–	Spearman	Agirre et al. (2012)
STS15	en	500	–	Spearman	Agirre et al. (2016)
STS16	en	500	–	Spearman	Agirre et al. (2016)
BIOSSES	en	100	–	Spearman	Soğancıoğlu et al. (2017)
STS17	en, de, es, fr	1k	–	Spearman	Cer et al. (2017)
STS22v2	en, de, es, fr, ru, zh	2k	–	Spearman	Chen et al. (2022)
Clustering (9 tasks)					
RedditClustP2P	en	1k	–	V-meas.	Muennighoff et al. (2023)
TwentyNewsgroupsV2	en	1k	–	V-meas.	Muennighoff et al. (2023)
StackExchangeClustP2PV2	en	1k	–	V-meas.	Muennighoff et al. (2023)
StackExchangeClustV2	en	1k	–	V-meas.	Muennighoff et al. (2023)
ArxivClustP2P	en	1k	–	V-meas.	Muennighoff et al. (2023)
ArxivClustS2S	en	1k	–	V-meas.	Muennighoff et al. (2023)
BiorxivClustP2PV2	en	1k	–	V-meas.	Muennighoff et al. (2023)
MedrxivClustP2PV2	en	1k	–	V-meas.	Muennighoff et al. (2023)
MedrxivClustS2SV2	en	1k	–	V-meas.	Muennighoff et al. (2023)
Pair Classification (4 tasks)					
SprintDuplicateQuestionsPC	en	500	2	Avg. Prec.	Shah et al. (2018)
TwitterURLCorpusPC	en	500	2	Avg. Prec.	Lan et al. (2017)
LegalBenchPC	en	500	2	Avg. Prec.	Guha et al. (2023)
RTE3PC	de, en, fr, it	2k	2	Avg. Prec.	Giampiccolo et al. (2007)
Retrieval (6 tasks)					
AILAStatutes	en	50 Q / 82 C	–	Recall@1	Enevoldsen et al. (2025)
FQuADRetrieval	fr	100 Q / 269 C	–	Recall@1	d'Hoffschmidt et al. (2020)
HC3FinanceRetrieval	en	100 Q / 415 C	–	Recall@1	Guo et al. (2023)
LegalBenchConsumerContractsQA	en	100 Q / 154 C	–	Recall@1	Guha et al. (2023)
PublicHealthQA	en	100 Q / 172 C	–	Recall@1	HF dataset
TwitterHjerneRetrieval	da	77 Q / 262 C	–	Recall@1	Vejlgaard Holm et al. (2025)

Table 12: **MTEB(LLM) task suite (37 tasks).** N = held-out test samples (summed over languages for multilingual tasks); Q = queries, C = corpus documents. Multilingual tasks are evaluated per language and averaged. Held-out subsets (seed 42) derived from MTEB and MMTEB tasks (Muennighoff et al., 2023; Enevoldsen et al., 2025), hosted at `mteb/llm-eval-*`. Token counts: Table 13.

Consumer-Contracts QA (legal). Yes/no questions over consumer-contract clauses. Pro (86.0) outperforms the best embedding (70.0), consistent with the need to interpret the clause.

Public-Health QA. Pro (90.0) edges the best embedding (85.0); factual health QA is partly captured by dense retrieval, narrowing the gap.

HC3-Finance QA. Pro (71.0) narrowly leads the best embedding (67.0) on financial QA retrieval.

TwitterHjerne (Danish social media). Retrieving relevant tweets given a query. Pro (33.4) slightly outperforms embeddings (best: 31.5); both handle keyword-centric retrieval similarly.

AILAStatutes (legal statute retrieval): where reasoning hurts. Matching case facts to relevant statutes. The best embedding (Octen-8B, 23.2) outperforms Pro (14.5): Pro over-retrieves

Classification		
Task	**Test**	**Train (kNN)***
ImdbCls	132k	7,281k
Banking77Cls	38k	134k
AmazonCounterfactualCls	22k	258k
MTOPDomainCls	23k	393k
MassiveIntentCls	37k	433k
MassiveScenarioCls	30k	433k
ToxicConversationsCls	33k	3,239k
TweetSentimentCls	9k	483k
STS		
STSBenchmark		12k
SICK-R		10k
STS12		14k
STS13		12k
STS14		12k
STS15		12k
STS16		14k
BIOSSES		7k
STS17		21k
STS22v2		1,866k
Clustering		
RedditClustP2P		178k
TwentyNewsgroupsV2		8k
StackExchangeClustP2PV2		276k
StackExchangeClustV2		13k
ArxivClustP2P		226k
ArxivClustS2S		16k
BiorxivClustP2PV2		313k
MedrxivClustP2PV2		401k
MedrxivClustS2SV2		23k
PairClassification		
SprintDuplicateQuestionsPC		14k
TwitterURLCorpusPC		18k
LegalBenchPC		37k
RTE3PC		114k
Task	**Queries**	**Corpus**
Retrieval		
AILAStatutes	33k	37k
FQuADRetrieval	1k	59k
HC3FinanceRetrieval	1k	89k
LegalBenchConsumerContractsQA	2k	83k
PublicHealthQA	1k	28k
TwitterHjerneRetrieval	4k	10k

Table 13: **Token budget per task** (GPT-4o tokenizer; raw text). Actual counts vary by ±15–40% across model vocabularies. *Train split is the kNN reference corpus processed only by embedding models; LLMs process only the test split. Corpus-in-context formatting adds ∼20 tokens per document for LLM retrieval.

broadly associated provisions, while cosine similarity naturally produces a narrower, more precise ranking.

D Per-Category Rankings and Figures

This section provides task- and model-level figures, followed by category-level rankings. Figure 6 is the task-level view of the paper's central claim and is referenced from §4.2.

Task	Gemini 3 Flash	Gemini 3 Flash (low)	Δ	Think ↓	Best Emb.
Retrieval					
AILAStatutes	5.7	**12.0**	+6.3	–	23.2
FQuADRetrieval	88.0	**92.0**	+4.0	54%	72.0
HC3FinanceRetrieval	60.0	**66.0**	+6.0	87%	67.0
LegalBenchConsumerContractsQA	79.0	**83.0**	+4.0	85%	70.0
PublicHealthQA	49.0	**66.0**	+17.0	94%	85.0
TwitterHjerneRetrieval	32.4	**33.4**	+1.1	–	31.5

Table 14: **Reduced-thinking ablation** (Gemini 3 Flash with `reasoning_effort=low` vs. default) on the MTEB(LLM) retrieval tasks. Think ↓ = reduction in thinking tokens vs. default. Reducing thinking by 54–94% improves all six retrieval scores in this ablation.

Task	Classes	Zero-shot	5-shot	Δ	Best Emb.
IMDB	2	**0.976**	0.974	−0.002	**0.976**
ToxicConversations	2	**0.900**	0.838	−0.062	0.810
TweetSentiment	3	0.700	0.682	−0.018	**0.710**
Banking77	77	0.831	0.165	−0.666	**0.960**

Table 15: **Few-shot classification ablation** (Flash, 5 in-context examples vs. zero-shot). **Bold** = best score per task across all methods. Five-shot prompting matches zero-shot performance on the 2–3 class tasks and lowers Banking77 performance, where the prompt contains five examples for 77 labels.

D.1 Task- and Model-Level Figures

The figures below show task-level performance, throughput, category profiles, and cross-model correlations.

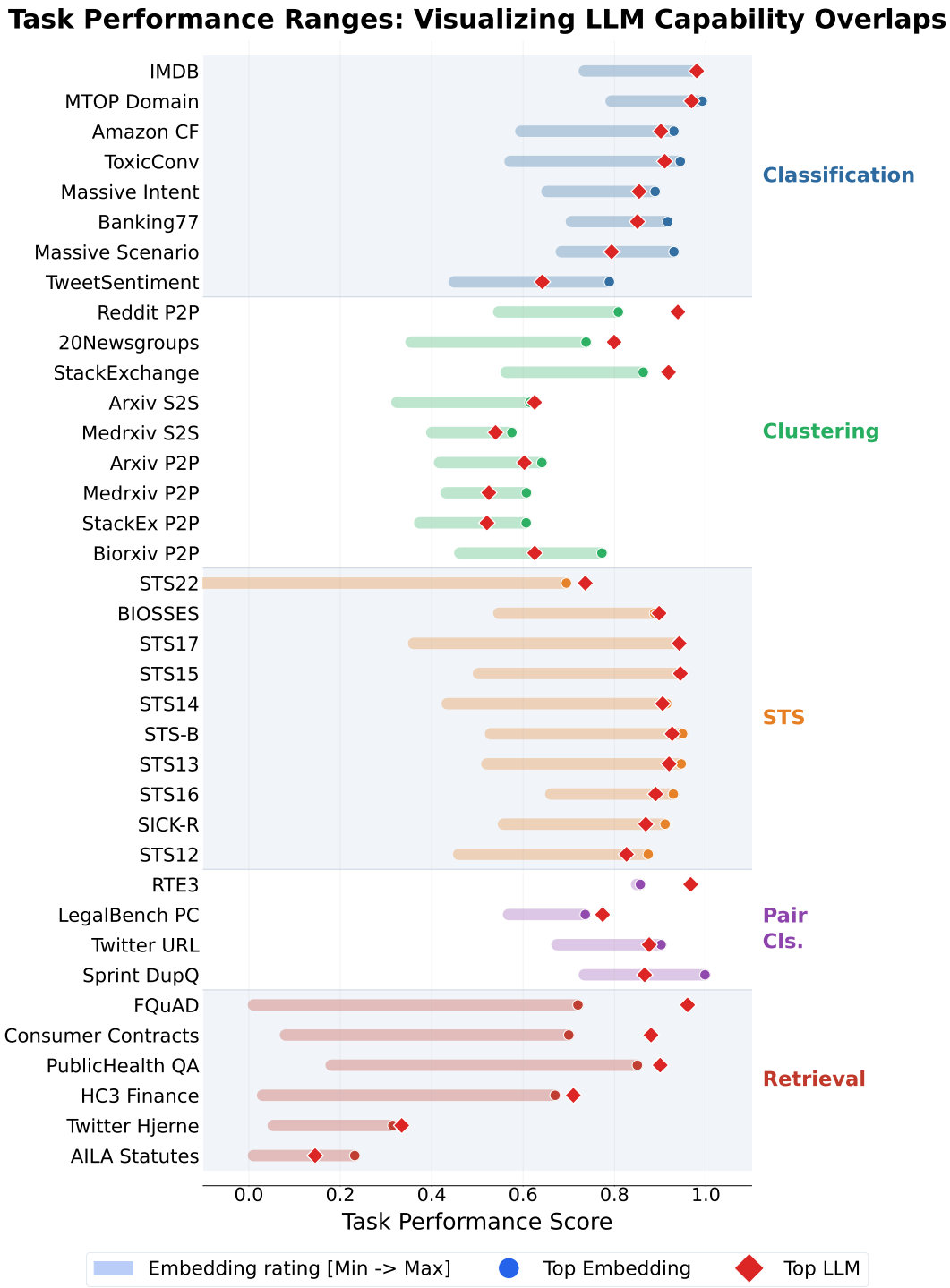

Figure 6: **Per-task performance.** Horizontal bars show the full embedding score range (min to max across 26 models) for each MTEB(LLM) task, grouped by category. Circle = best embedding; diamond = best LLM on that task, taken over all ten (six different LLMs hold it across the suite, Gemini 3.1 Pro on 21 of 37 tasks). Where the diamond falls inside the bar, some embedding model already matches the best LLM: this holds on 7 of 8 classification tasks and 7 of 10 STS tasks, but on only 1 of 6 retrieval tasks.

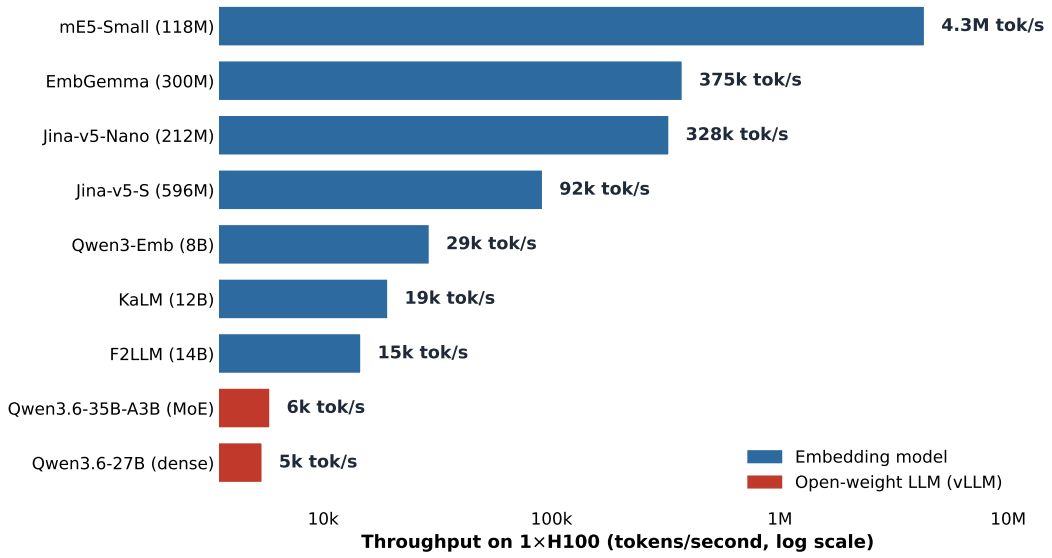

Figure 7: **Same-hardware inference throughput.** Two open-weight LLMs and seven representative embedding models (118M–14B) served on one H100 (tokens/second, log scale); the full 26-model embedding throughput is in Table 18. Even the slowest embedding runs ∼2.5× faster than the fastest LLM; the fastest runs ∼736× faster.

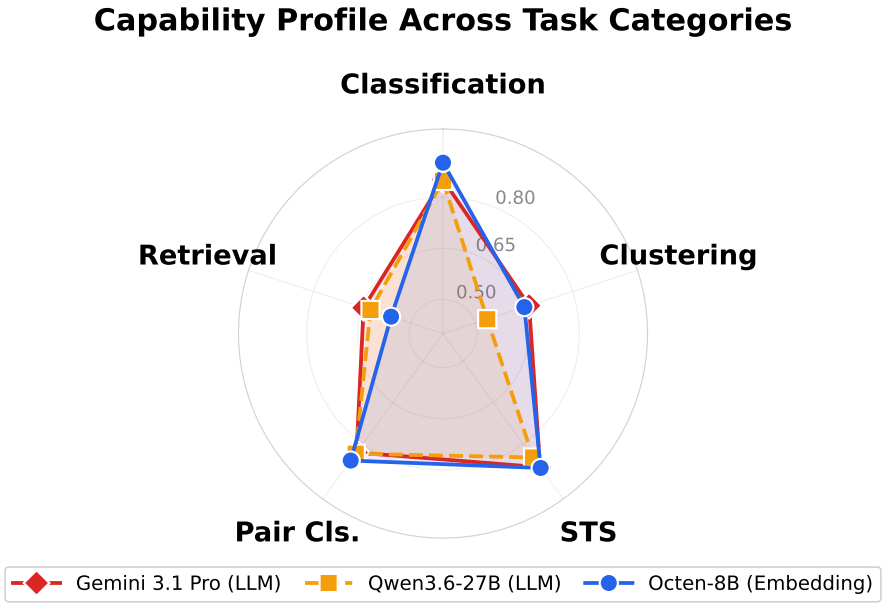

Figure 8: **Capability profiles across five task categories.** Gemini 3.1 Pro (red), Qwen3.6-27B (amber), and Octen-8B (blue). The LLMs lead on retrieval, while Octen-8B leads on classification and remains competitive across categories.

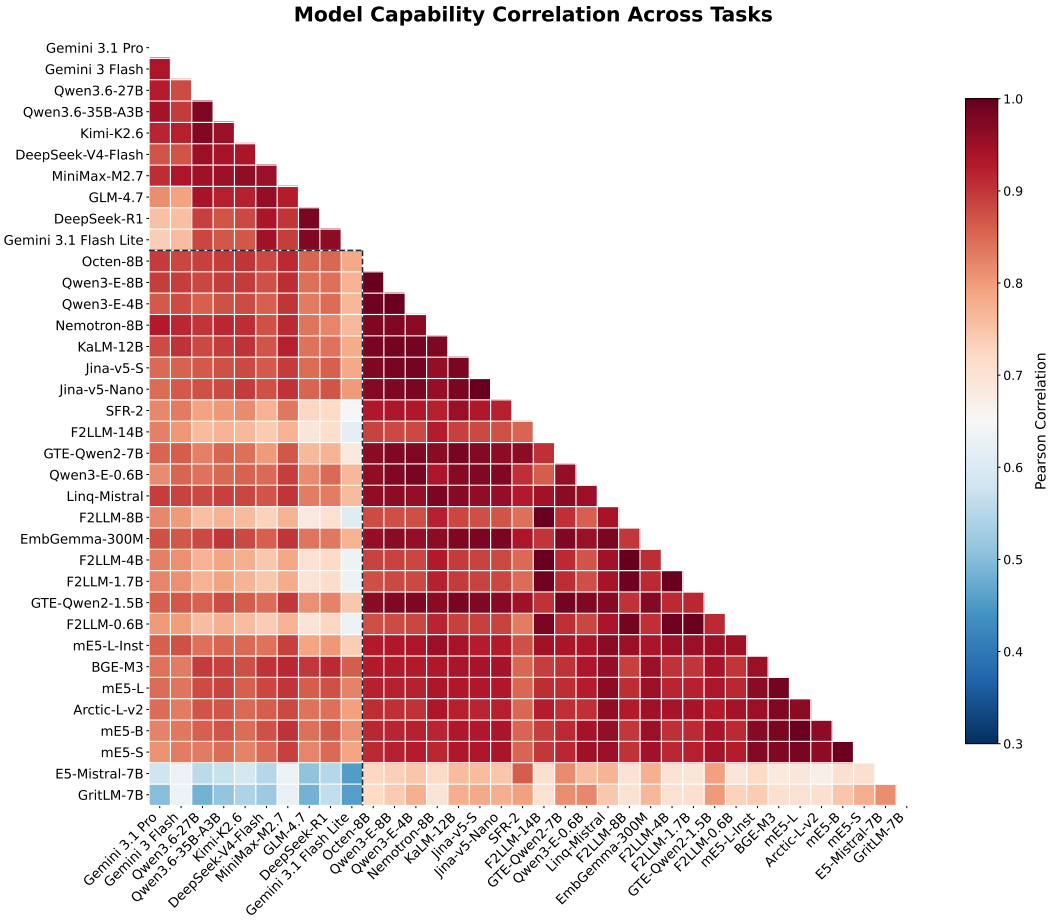

Figure 9: **Cross-model task-behaviour correlation.** Pearson correlations based on per-task MTEB(LLM) scores. LLMs and embedding models form visually distinct clusters: LLMs are strongly intercorrelated while embedding models cluster by architecture family. Lower inter-paradigm correlations are consistent with their different category profiles.

D.2 Per-Category Model Rankings

Figures 10–15 rank all 36 evaluated models overall and within each of the five MTEB task categories. They show category-dependent performance: LLMs lead on retrieval, embedding models lead on classification, and the two paradigms are statistically indistinguishable on clustering, STS, and pair classification. Flash-Lite ranks near the bottom in every category, below the reasoning-capable Gemini models.

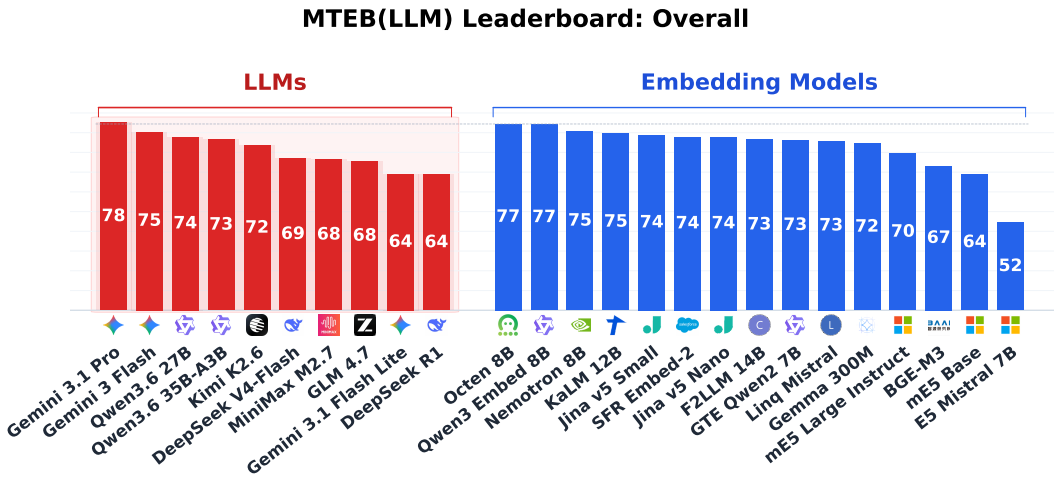

Figure 10: **Overall rankings.** Pro (77.6) narrowly leads Octen-8B (77.2); the difference lies within statistical noise ($p = 0.85$). Flash-Lite (64.5) ranks near the bottom.

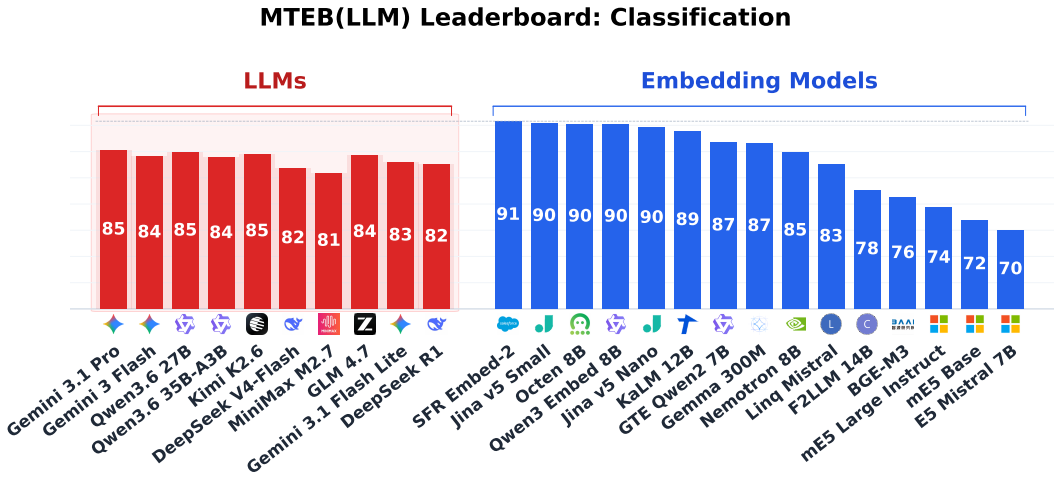

Figure 11: **Classification rankings.** SFR-2 ranks first (90.8); Pro scores 85.2 and ranks below ten embedding models.

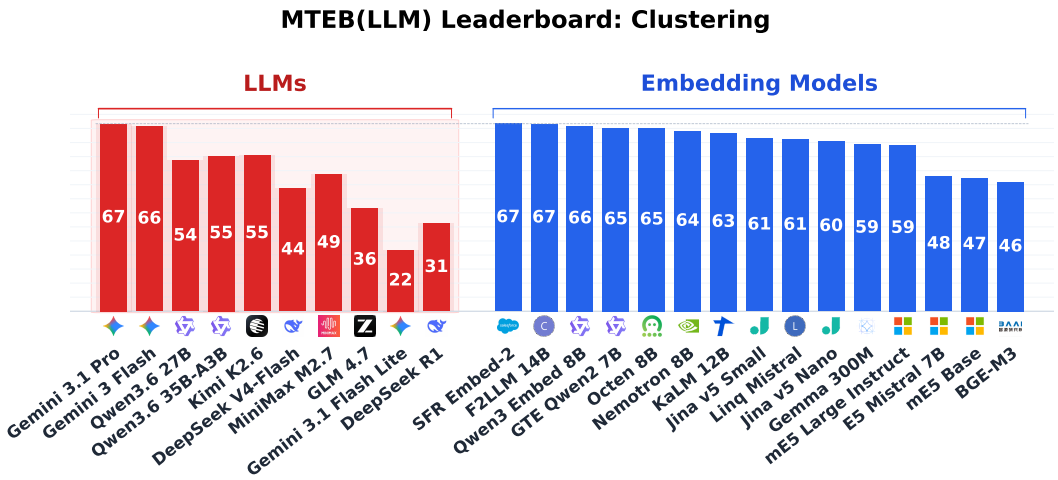

Figure 12: **Clustering rankings.** Statistical tie; SFR-2 (66.7) edges Pro (66.6). Flash-Lite scores 21.7.

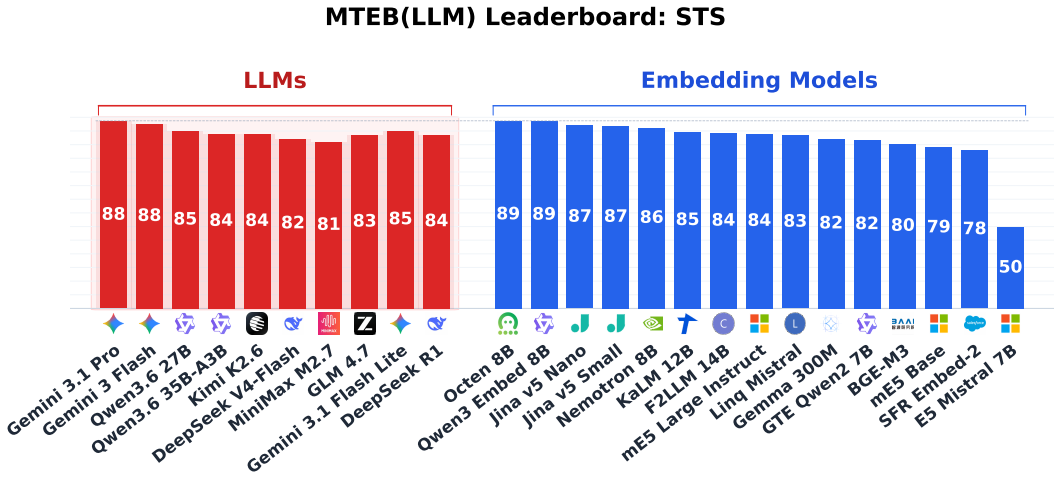

Figure 13: **STS rankings.** Qwen3-E-4B (88.8) and Pro (88.5) are essentially tied; no significant paradigm difference ($p = 0.74$).

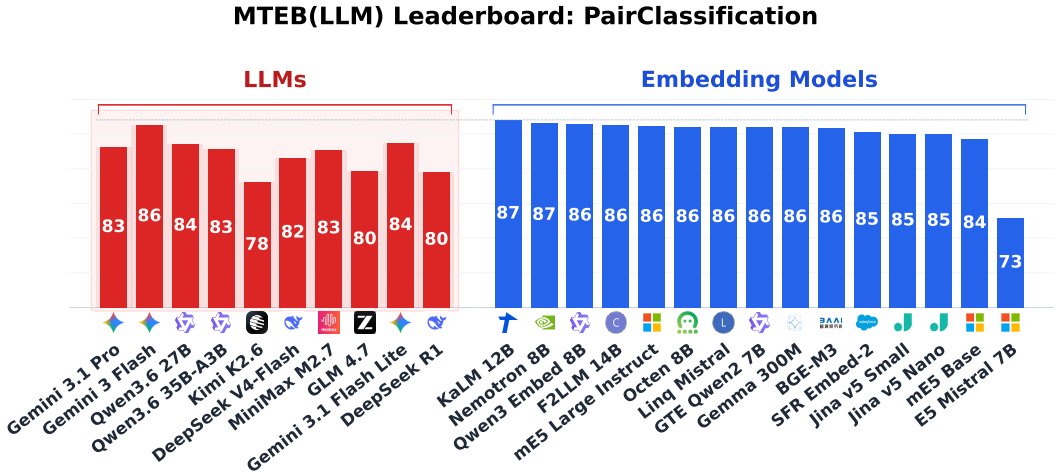

Figure 14: **Pair Classification rankings.** KaLM-12B (87.1) outperforms Pro (83.2).

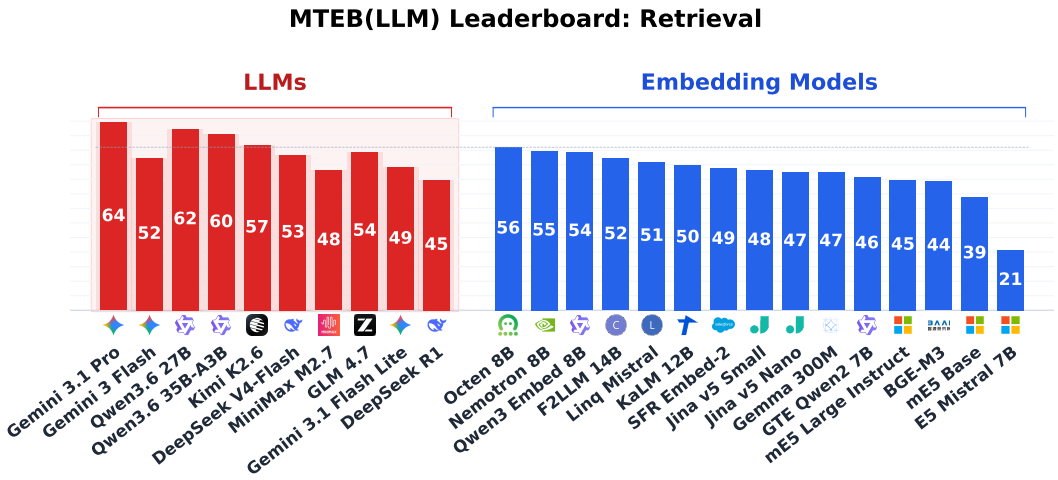

Figure 15: **Retrieval rankings.** Pro leads the best embedding (64.5 vs. 56.0) and ranks first on five of six tasks.

D.3 Full Retrieve-then-Rerank Matrix

Table 16 reports the complete matrix summarised by Figure 3 (§4.3): all four first-stage retrievers crossed with every cross-encoder and LLM listwise reranker, on BEIR and BRIGHT.

E Qualitative Error Analysis

We present model predictions from the evaluation files to illustrate differences between LLM and embedding approaches. The examples cover classification ambiguity, STS calibration, retrieval precision, and differences between Pro and Flash.

E.1 Classification: Over-Interpretation vs. Anchored Reference

Several classification errors involve ambiguous inputs for which the LLM infers an intent that differs from the dataset convention. With kNN, an embedding classifier can anchor its prediction to similar labelled examples, which helps on fine-grained tasks such as Banking77 and MassiveIntent.

Banking77 (77 classes): "locate my card."

> **Input:** "How do I locate my card?"
> **Pro reasoning:** "The user is asking how to locate their card, indicating that they have misplaced it or it is missing. This aligns with the intent for a lost or stolen card."
> **Pro prediction:** `lost_or_stolen_card` Incorrect (gold: `card_arrival`)
> **Flash prediction:** `card_arrival` Correct

Flash predicts the gold label; Pro predicts `lost_or_stolen_card`. The LLM infers "locate" implies loss; Flash's lower-depth reasoning follows the surface-level reading. In the training data, "locate my card" aligns with delivery tracking. The kNN classifier matches to examples like "Where is my card?" and correctly predicts `card_arrival`.

Banking77: "know when my card will arrive."

> **Input:** "Is there a way to know when my card will arrive?"
> **Pro reasoning:** "The user is asking for information regarding the timeframe or expected date for their card's delivery, which aligns with the `card_delivery_estimate` category."

First stage	Pure	BGE-Gemma	Qwen3-RR-4B	Qwen3-RR-8B	Qwen3.6-27B†	Qwen3.6-35B†
BRIGHT (reasoning)						
BM25	10.2	18.2	22.0	21.9	**24.2**	23.5
BGE-large	15.9	18.9	22.7	21.5	**27.6**	26.1
GTE-MC-v1	15.9	19.2	23.9	23.0	**28.8**	27.9
Qwen3-E-8B	22.3	20.6	26.4	24.6	**35.1**	33.6
BEIR (semantic)						
BM25	39.8	50.5	**53.2**	52.5	52.1	51.1
BGE-large	52.7	55.3	**58.6**	57.6	57.6	56.2
GTE-MC-v1	53.7	55.0	**58.5**	58.5	57.5	56.9
Qwen3-E-8B	**63.1**	56.8	60.3	59.5	58.9	58.9

Table 16: **Retrieve-then-rerank matrix.** Average nDCG@10 for first-stage retrievers crossed with cross-encoder and LLM listwise (†) rerankers over 7 BRIGHT and 5 BEIR tasks. Bold = best per row. An LLM reranker improves Qwen3-E-8B on BRIGHT (22.3→35.1); on BEIR, the embedding alone scores highest (63.1).

Pro prediction: `card_delivery_estimate` Incorrect (gold: `card_arrival`)
Flash prediction: `card_delivery_estimate` Incorrect (gold: `card_arrival`)

Both models make the same error here, illustrating the ambiguity between "arrival" and "delivery estimate" without labelled context. The kNN classifier uses the full 77-class reference set to resolve such near-neighbour cases.

MassiveIntent (60 classes): "remind me about my schedule."

Input (en): "remind me about my schedule for the afternoon"
Pro reasoning: "The user is asking to be reminded about their schedule for the afternoon, which involves checking or querying their calendar events."
Pro prediction: `calendar_query` Incorrect (gold: `calendar_set`)

"Remind me" is an action verb (set a reminder) but the LLM interprets it as information-seeking (query). This example illustrates the ambiguity between action and query intents.

MassiveScenario (German): "wir brauchen rosa."

Input (de): "wir brauchen rosa" (we need pink)
Pro reasoning: "The utterance 'wir brauchen rosa' (we need pink) implies adding an item ('rosa' / pink) to a shopping list or note-taking application."
Pro prediction: `lists` Incorrect (gold: `iot`)

The LLM correctly reasons about the surface form but miscategorises the scenario: `iot` refers to controlling physical smart-home devices, not arbitrary colour references.

Input (de): "wie ist die einstellung für die helligkeit an meinem bildschirm" (what is the brightness setting on my screen)
Pro reasoning: "The user is asking about the brightness setting on their screen, which involves controlling or checking the state of a device. This falls under the IoT category."
Pro prediction: `iot` Incorrect (gold: `general`)

Both errors reflect the fine-grained scenario taxonomy: `iot` refers to controlling physical smart-home devices, not arbitrary brightness settings or colour references. The LLM correctly identifies device-related surface features but miscategorises the scenario.

AmazonCounterfactual: hypothetical "would be great."

Input: "Would be great to have on my tablet instead of carrying the book."
Pro reasoning: "The review expresses a hypothetical situation ('have on my tablet') that has not actually happened, indicated by the phrase 'Would be great'. This describes an alternative condition and its potential consequence."

Pro prediction: `counterfactual` Incorrect (gold: `not-counterfactual`)

The phrase "would be" supports the LLM's hypothetical reading. The dataset convention labels forward-looking wishes as `not-counterfactual`. This may reflect a mismatch between a plausible linguistic interpretation and the dataset's annotation convention.

When LLMs succeed on classification.

Input: "I still have not received my new card, I ordered over a week ago."
Pro prediction: `card_arrival` Correct

In this less ambiguous example, the LLM matches the gold annotation.

E.2 STS: Calibration and Semantic Precision

The examples below show two kinds of disagreement between LLM scores and STS annotations: sensitivity to fine distinctions and higher scores for inferred mechanistic equivalence.

STSBenchmark: agreement on a simple paraphrase.

S1: "open again, open anew" **S2:** "open again or anew."
Pro: 5.0 **Gold:** 5.0 $|\Delta|$**:** 0.0

The LLM score matches the annotation for this near-identical paraphrase.

STSBenchmark: conservative scoring on ideological distinctions.

S1: "...a shift from a muslim identity to an islamist identity."
S2: "...to change a muslim identity for an islamic identity."
Pro: 4.5 **Gold:** 5.0 $|\Delta|$**:** 0.5

The LLM distinguishes "Islamist" (political ideology) from "Islamic" (religious adjective); the gold standard treats them as equivalent. This linguistically defensible distinction reduces agreement with the benchmark annotation.

BIOSSES: conservative scoring on biomedical paraphrases.

S1: "It has recently been shown that Craf is essential for Kras G12D-induced NSCLC."
S2: "It has recently become evident that Craf is essential for the onset of Kras-driven non-small cell lung cancer."
Pro reasoning: "Both sentences state that Craf is essential for the development of Kras-driven NSCLC. Sentence 1 specifies the particular mutation (Kras G12D), whereas Sentence 2 is slightly more general."
Pro: 3.8 **Gold:** 4.0 $|\Delta|$**:** 0.2

The LLM correctly notes the specificity difference ("Kras G12D" vs. "Kras-driven") and downweights accordingly. The gold annotation gives 4.0, treating the two as highly similar; the LLM's 3.8 is slightly more conservative.

BIOSSES: LLM over-scores on mechanistic equivalence.

S1: "More recently, IDH mutations and resultant 2-hydroxyglutarate (2HG) production in leukemia cells were found to impair TET2 function."
S2: "It has also been recently reported that mutations of the isocitrate dehydrogenase genes IDH1 and IDH2 inhibit TET catalytic activity."
Pro reasoning: "Both sentences describe the exact same biochemical pathway: mutations in IDH genes lead to the production of 2-hydroxyglutarate (2-HG), which in turn inhibits TET2 enzymatic activity."
Pro: 3.8 **Gold:** 3.2 $|\Delta|$**:** 0.6

The LLM identifies a mechanistic relationship and scores the pair 3.8, compared with the gold score of 3.2. Expert adjudication would be needed to determine whether this difference reflects model calibration or annotation criteria.

E.3 Retrieval: Reasoning Depth vs. Geometric Precision

The retrieval examples show how joint query–document processing can help on reading-comprehension tasks and how broader associations can hurt precise statute retrieval.

FQuAD: reading comprehension in context. On French reading-comprehension retrieval, the LLM reads each candidate passage in the prompt and selects the one that answers the query, reaching Recall@1 = 92.0 versus 72.0 for the best embedding. The LLM processes the query jointly with the candidate passages, which may explain part of the observed advantage.

AILAStatutes: where reasoning hurts.

> **Query:** Legal case about a bank employee terminated for misconduct.
> **Pro:** Returns 14 documents spanning broad legal concepts (contract law, employment law, due process). Gold: only 4 specific statutes. The LLM's associative reasoning retrieves plausible but non-relevant provisions.
> **Octen-8B:** Returns a narrower, more precise set via cosine similarity with legal embeddings.

In this example, Pro retrieves plausible but irrelevant provisions, while the embedding model produces a narrower ranking.

TwitterHjerne: corpus-in-context precision.

> **Query (Danish):** "Sønnen vil gerne lave #pebernødder. De par gange jeg har prøvet det, blev de kun OK. Er der nogen, der kan anbefale en opskrift? #twitterhjerne" ("My son would like to make peppernuts. The few times I've tried, they were only OK. Can anyone recommend a recipe? #twitterhjerne")
> **Pro:** Returns documents [1, 2]: exact match. **Recall@1 = 1.0 on this query.**
> **Best embedding:** Recall@1 = 31.5 over the full task.

For this query with explicit topical keywords, corpus-in-context retrieval returns the correct tweet first. The overall task-level advantage for Pro (33.4 vs. 31.5 for embeddings) is modest, reflecting that both paradigms handle keyword-centric retrieval similarly.

E.4 Model Scale Effects: Flash vs. Pro on the Same Input

Banking77: Flash succeeds where Pro over-interprets. As shown above, Flash correctly predicts `card_arrival` for "How do I locate my card?" while Pro predicts `lost_or_stolen_card`. In this example, more reasoning does not improve the prediction. Pro's extended chain-of-thought introduces an interpretive step ("locate implies misplacement") that Flash's lighter-weight reasoning bypasses. Aggregate scores provide context: Pro remains slightly stronger than Flash on classification overall.

Retrieval scale effects. On FQuAD reading-comprehension retrieval, all three Gemini models outscore the best embedding, with a clear quality ladder:

- Pro (Recall@1 = 92.0): reads and matches passages precisely.
- Flash (Recall@1 = 88.0): captures most of the advantage.
- Flash-Lite (Recall@1 = 84.0): still well above embeddings, even without reasoning.
- Best embedding (Recall@1 = 72.0): limited to single-vector matching.

Flash-Lite already outperforms the best embedding on this task; Pro adds a modest gain at much higher cost. This result is consistent with the reduced-thinking ablation (§4.6).

Summary. The examples illustrate four patterns:

1. **Ambiguous label boundaries**: LLM predictions can differ from dataset conventions; kNN anchors decisions to labelled examples.
2. **STS calibration**: model scores can differ from annotations when they attend to different semantic details.
3. **Retrieval breadth**: joint processing can help reading comprehension but can also retrieve broadly related material.
4. **Model scale**: the value of additional capacity or reasoning varies across examples and must be weighed against cost.

Estimating the frequency of these patterns requires a systematic error analysis.

F Extended Related Work

F.1 LLMs as Text Encoders and Judges

The boundary between generative LLMs and discriminative embedding models has blurred considerably. LLM-as-a-judge frameworks (Zheng et al., 2023) repurpose frozen LLMs for evaluation. Sun et al. (2023b) find that zero-shot LLM classification underperforms fine-tuned models on standard benchmarks and propose chain-of-thought prompting to narrow the gap. Fine-tuned encoder models outperform zero-shot frontier LLMs across the classification benchmarks tested by Bucher & Martini (2024), with margins of 5 to over 75 F1 points depending on task granularity. This is consistent with our 5.6-point advantage for kNN-equipped embedding pipelines. The LOFT benchmark (Lee et al., 2024) evaluates long-context LLMs on retrieval by placing entire corpora in the prompt (corpus-in-context prompting); our retrieval evaluation adopts the same protocol.

On the representational side, several lines of work extract embeddings from LLMs. SGPT (Muennighoff, 2022) uses weighted mean pooling over GPT decoder states. Jiang et al. (2023) use in-context learning to extract sentence embeddings from decoder-only LLMs, while BehnamGhader et al. (2024) adapt decoder-only LLMs into sentence encoders. Springer et al. (2025) show that repeating the input can improve these embeddings by enabling effective bidirectional attention. GritLM (Muennighoff et al., 2024) goes further, unifying generation and representation in a single model that achieved strong performance on both MTEB and generative benchmarks at the time of its release.

These methods combine elements of both paradigms. At inference time, they produce fixed-length vectors compared via cosine similarity, so our cost accounting treats them as embedding models. Several of our 26 evaluated models (e.g., E5-Mistral, GritLM) are built on LLM backbones and incorporate LLM-derived representations.

HUME (Assadi et al., 2025) also benchmarks LLMs as annotators on MTEB tasks and finds that they fall short of human judgment on reranking. This complements our finding that zero-shot LLM classifiers underperform embedding pipelines with kNN access to labelled references. Our study adds a controlled comparison on identical tasks, data, and metrics, together with exact cost accounting.

F.2 Frontier LLMs and Open-Weight Reasoning Models

LLMs have advanced rapidly since GPT-4 (OpenAI, 2023). OpenAI's o1 (OpenAI, 2024) demonstrated extended chain-of-thought reasoning at inference time using reinforcement learning. At the time of our evaluation, closed-source models with controllable reasoning included Gemini 3.1 Pro, GPT-5 (Singh et al., 2025), and Claude Opus 4.6 (Anthropic, 2026). These models support extended thinking tokens billed separately from output; newer models have since appeared (e.g. GPT-5.4). Open-weight alternatives include Llama 3 (Grattafiori et al., 2024), Qwen3 (Yang et al., 2025), and DeepSeek-R1 (DeepSeek-AI et al., 2025). DeepSeek-R1 shows that reinforcement-learning-based training can elicit strong

reasoning at lower inference cost. Gemma 3 (Gemma Team et al., 2025) is a compact multilingual model family.

We evaluate ten LLMs across six families (§3), including DeepSeek-R1 and open-weight Qwen3.6, GLM, Kimi, and MiniMax models alongside Gemini 3. The cost and task-level patterns recur across this set, although their magnitude varies by model. GPT-5 (Singh et al., 2025) and Claude Opus 4.6 (Anthropic, 2026) remain to be evaluated. The same evaluation framework can accommodate these and future models.

F.3 Cross-Encoders and Late-Interaction Retrieval

Cross-encoder rerankers (models that jointly encode query and document to produce a relevance score) and late-interaction models such as ColBERT (Khattab & Zaharia, 2020) occupy a middle ground between bi-encoder speed and LLM reasoning depth. We evaluate this middle ground directly (§4.3) by crossing four first-stage retrievers with cross-encoder and LLM listwise rerankers on BEIR and BRIGHT (Sun et al., 2023a; Weller et al., 2025). Reranking improves reasoning-heavy retrieval, while the strongest embedding alone remains best on semantic retrieval.

Li et al. (2024) compare RAG with long-context LLMs on QA tasks. Long-context LLMs achieve higher quality at greater token cost, while a Self-Route hybrid retains 93–100% of their quality at 38–61% of the token cost, depending on the model. This result supports the hybrid recommendation in §5.

G Extended Limitations

We expand on the six limitations summarised in §5.

(i) LLM coverage. The ten evaluated LLMs span six families, dense and mixture-of-experts architectures, and reasoning and instruct variants. They provide a snapshot of a fast-moving frontier. At current public rates, the Pareto frontier includes Gemini 3.1 Pro and the leading embeddings; the lowest-cost LLM evaluated scores below the best embedding. Closed APIs not yet covered (GPT-5 (Singh et al., 2025) and Claude Opus 4.6 (Anthropic, 2026)) may have different cost–capability profiles, so the numerical conclusions should be updated as new evaluations become available. The released evaluation pipeline makes this straightforward to extend as new models are released.

(ii) Supervision asymmetry. Classification uses labelled-reference kNN for embeddings and zero-shot prompting for LLMs. This setup reflects deployments with scarce labelled data and gives the two pipelines different supervision. Five examples preserve performance on the small-label tasks and reduce Banking77 performance (§4.7).

(iii) Small-corpus retrieval. The corpus-in-context protocol places entire corpora (82–415 documents) in the LLM prompt, a setting that grants the LLM three structural advantages: the corpus fits entirely in context, prompt caching amortises input cost across queries, and the model can reason globally over the full document set. Each is a property of the protocol rather than a measured effect; we did not vary corpus size, so we do not quantify how retrieval quality would change at larger scale. At BEIR or production scale, placing the full corpus in context is generally impractical; an indexed first stage is needed, so the small-corpus cost gap is a lower bound for this protocol. We complement corpus-in-context retrieval with bi-encoder, cross-encoder, and LLM-reranker comparisons on BEIR and BRIGHT (§4.3). Reranking benefits reasoning-heavy retrieval; the strongest embedding retains the lead on semantic retrieval (Sun et al., 2023a; Weller et al., 2025). nDCG@k results for the small-corpus setting are available in the released data files and preserve the paradigm-level ordering.

(iv) Reasoning-control granularity. Reasoning controls differ by provider: Gemini's `reasoning_effort=low` provides a soft hint, while the open models support fully disabled

reasoning. We use both settings. Reduced reasoning preserves or improves retrieval for four of six models across five families (Figure 4b); token accounting varies by provider.

(v) Throughput comparison. We serve two open-weight LLMs (Qwen3.6-27B and Qwen3.6-35B-A3B) and the embedding models on the same H100, removing API rate limits and hardware choice as confounds. The resulting 2.5–736× gap is consistent with the difference between autoregressive decoding and a single encoder pass.

(vi) Task weighting sensitivity. Our aggregate metric weights all MTEB(LLM) tasks equally regardless of dataset size, task difficulty, or practical importance. Alternative weighting schemes (by dataset size, normalised z-scores, or task-family weighting) may shift aggregate conclusions. Aggregate numbers should therefore be interpreted alongside the reported category- and task-level results. Alternative aggregation methods such as the Borda count used by MMTEB (Enevoldsen et al., 2025) are applicable to MTEB(LLM) and may produce different paradigm orderings.

H Cost and Throughput Details

This section reports token usage, embedding throughput, cost sensitivity, and the cost methodology underlying §4.4–4.5.

H.1 Detailed LLM Token Usage

Table 17 provides the complete token-level breakdown for each LLM across all MTEB(LLM) tasks. For each model, we report input, standard-output, cached, reasoning, and total tokens, together with the cost decomposition. Gemini reports reasoning separately, whereas OpenRouter includes it in the completion count. The reasoning-heavy LLMs generate 8–26M reasoning tokens (Qwen3.6-27B 26M, GLM-4.7 25M, Kimi-K2.6 22M, Gemini 3 Flash 14M, Pro 9M), compared with 1–3M standard-output tokens. Reasoning accounts for 28–81% of total inference cost across these models: 81% for Qwen3.6-27B and 28% for DeepSeek-V4-Flash. The non-reasoning Flash-Lite costs $6.85 and ranks below most embeddings.

	Tokens (M)				Cost (USD)			
Model	In_{nc}	Cached	Out	Think	In_{nc}	Cached	Out+Th	**Total**
Gemini 3.1 Pro	12.2	27.7	1.8	8.6	$24.39	$5.54	$124.21	**$154.14**
Gemini 3 Flash	12.4	27.5	1.8	14.3	$6.19	$1.38	$48.30	**$55.87**
Qwen3.6-27B	39.5	0.0	2.7	26.1	$11.46	$0.00	$91.97	**$103.43**
Qwen3.6-35B-A3B	39.5	0.0	2.5	26.0	$5.53	$0.00	$28.52	**$34.06**
Kimi-K2.6	40.1	0.0	2.6	21.7	$27.43	$0.00	$83.27	**$110.70**
DeepSeek-V4-Flash	16.8	22.6	2.1	4.5	$1.65	$0.22	$1.29	**$3.16**
MiniMax-M2.7	37.9	0.0	1.6	10.0	$10.58	$0.00	$13.93	**$24.52**
GLM-4.7	38.8	0.0	2.4	24.9	$15.54	$0.00	$47.77	**$63.31**
Gemini 3.1 Flash Lite	16.5	26.9	1.4	0.0	$4.13	$0.67	$2.05	**$6.85**
DeepSeek-R1	39.6	0.0	1.2	10.6	$27.73	$0.00	$29.65	**$57.38**

Table 17: **Detailed LLM token usage and cost** across all MTEB(LLM) tasks. In_{nc} = non-cached input (cached billed at 10% of input rate); Think = reasoning tokens (billed at output rate); Out+Th combines standard output and thinking cost.

H.2 Embedding Throughput and Cost

Table 18 reports the throughput (tokens/second) and derived cost per million tokens for each of the 26 embedding models, benchmarked on a single NVIDIA H100 80GB HBM3 GPU at sequence length 512 with maximum batch size. Costs range from $0.00016/MTok (mE5-small, 118M parameters, 4.3M tok/s) to $0.047/MTok (F2LLM-v2-14B, 14.7K tok/s). Total benchmark cost spans 291× across the embedding models, compared with a 1,431× gap between the best embedding and Gemini 3.1 Pro.

Each model's total benchmark cost uses its own tokenizer's token count (4.4–5.5M tokens across vocabularies), avoiding tokenizer mismatch.

Model	Params	Tok/s	$/MTok	Score
mE5-S	118M	4,314,796	0.0002	63.1
mE5-B	278M	1,910,005	0.0004	64.4
mE5-L-Inst	560M	640,842	0.0011	69.7
BGE-M3	568M	640,425	0.0011	66.5
Arctic-L-v2	568M	640,405	0.0011	65.3
mE5-L	560M	640,126	0.0011	65.9
EmbGemma-300M	308M	374,988	0.0018	72.2
Jina-v5-Nano	212M	327,790	0.0021	73.9
F2LLM-0.6B	596M	189,653	0.0036	69.9
Qwen3-E-0.6B	596M	175,692	0.0039	72.5
GTE-Qwen2-1.5B	1.5B	133,567	0.0052	70.8
F2LLM-1.7B	1.7B	106,406	0.0065	71.7
Jina-v5-S	596M	91,658	0.0075	74.4
Qwen3-E-4B	4.0B	46,080	0.0150	75.9
F2LLM-4B	4.0B	40,157	0.0172	71.9
GTE-Qwen2-7B	7.1B	35,419	0.0195	73.1
Octen-8B	7.6B	29,313	0.0236	77.2
Qwen3-E-8B	7.6B	29,262	0.0236	77.0
SFR-2	7.1B	28,196	0.0245	73.9
Nemotron-8B	7.5B	27,808	0.0249	75.3
GritLM-7B	7.2B	27,730	0.0249	51.2
Linq-Mistral	7.1B	27,677	0.0250	72.8
E5-Mistral-7B	7.1B	27,522	0.0251	52.2
F2LLM-8B	7.6B	25,957	0.0266	72.2
KaLM-12B	11.8B	19,244	0.0359	74.8
F2LLM-14B	14.0B	14,665	0.0472	73.3

Table 18: **Embedding throughput** on a single NVIDIA H100 80GB (median tokens/s over the benchmark), per-MTok cost at $2.49/hr spot, and mean (macro) MTEB(LLM) score.

H.3 Cost Sensitivity Analysis

Table 19 shows how the LLM-to-embedding cost ratio varies under alternative hardware and pricing assumptions. We consider five scenarios for embedding inference:

1. **H100 spot** ($2.49/hr, Lambda Labs, March 2026): baseline, 1,431×
2. **H100 on-demand** ($3.99/hr): higher price, 893×
3. **A100 spot** ($1.49/hr, est. 1.5× slower): 1,594×
4. **L4 spot** ($0.49/hr, est. 3× slower): much lower throughput, 2,424×
5. **Commercial API** (e.g. $0.10/MTok): fixed per-token pricing, 338×

In all five scenarios, the cost ratio exceeds 300×.

Hardware / Pricing Scenario	Emb. Cost	LLM Cost	Ratio
H100 spot $2.49/hr (our setup)	$0.108	$154.14	1,431×
H100 on-demand $3.99/hr	$0.173	$154.14	893×
A100 spot $1.49/hr (est. 1.5× slower)	$0.097	$154.14	1,594×
L4 spot $0.49/hr (est. 3× slower)	$0.064	$154.14	2,424×
Commercial API $0.10/MTok	$0.457	$154.14	338×

Table 19: **Cost sensitivity analysis.** LLM-to-embedding cost ratio under alternative hardware and pricing scenarios. Costs compare Octen-8B with Gemini 3.1 Pro at fixed API pricing; ratios range from 338–2,424×.

Per-query economics and training amortisation. The benchmark-level ratios translate directly to per-query terms: both paradigms answer the same queries, so dividing by the query count rescales both sides equally and leaves the 29–1,431× range unchanged. Two effects widen the gap further under sustained deployment. First, embedding training is a one-time cost amortised over the deployment lifetime (and, for public checkpoints, across all users), so its per-query contribution vanishes at scale, whereas LLM inference cost recurs in full on every query. Second, document embeddings are computed once and reused across queries, while corpus-in-context reading repeats per query and is only partially offset by prompt caching. The benchmark-pass comparison is therefore conservative: it charges embeddings their full encoding cost while granting the LLM cached-input rates.

H.4 Detailed Cost Methodology

Embedding cost pipeline. For each model, we tokenize all MTEB(LLM) tasks with its HuggingFace `AutoTokenizer`. We then benchmark maximum-throughput processing on an NVIDIA H100 80GB at sequence length 512, using the largest batch that fits in memory. Cost is $(\text{tokens}/10^6) \times (r_{\text{GPU}}/T)$, where $r_{\text{GPU}} = \$2.49/\text{hr}$ (Lambda Labs spot pricing, March 2026) and T is sustained throughput in tokens/hour.

Token counts range from 4.4M (Gemma-based tokenizers) to 5.5M (Mistral-based), a 26% range driven by vocabulary differences. Using a single proxy tokenizer (e.g., GPT-2 at 6.75M tokens) would overestimate embedding costs by 22–54%.

LLM cost pipeline. Token usage is extracted from each provider's `usage_stats` in every task's JSON result file. For Gemini, `input_tokens` = `prompt_token_count` (including cached tokens), `output_tokens` = `candidates_token_count` (excluding reasoning), and `cached_tokens` = `cached_content_token_count`. Total tokens equal input plus output plus reasoning. For the open models served via OpenRouter, the completion count already includes reasoning and `total` = input + output, with reasoning reported separately. We bill every generated token, total − input, at the output rate. This equals output plus reasoning for Gemini and the reasoning-inclusive completion count for open models. Each model's reasoning share comes from its usage statistics (Figure 4). For multilingual tasks (STS17, STS22v2, RTE3), per-language entries report different token counts and are summed; for classification tasks with multiple entries per split, all entries share identical `usage_stats` and are counted once.

Gemini API prices per MTok input/output are \$0.25/\$1.50 for Flash-Lite, \$0.50/\$3.00 for Flash, and \$2.00/\$12.00 for Pro; cached input is billed at 10%. Open models use OpenRouter public rates (March/June 2026).

Throughput measurement. We serve both paradigms on the same NVIDIA H100 and measure sustained tokens/second. Open-weight LLMs (Qwen3.6-27B, Qwen3.6-35B-A3B) run under vLLM (BF16, tensor-parallel = 1, 256 concurrent requests, 200/100 input/output tokens); embedding models run batch inference on the same GPU. Holding hardware fixed removes API rate limits and GPU choice as explanations for the measured gap.

I Reproducibility

All code, datasets, result files, and analysis scripts are released at https://github.com/embeddings-benchmark/embedders-dilemma and implemented with the MTEB framework.

LLM prompt templates. Prompt design follows a consistent structure across all five task categories:

- **System prompt**: one sentence describing the task type and output format.
- **User prompt**: task-specific instruction, followed by the input text(s).

- **Output schema**: a Pydantic model with fields `reasoning` (free text, optional) and `output` (constrained to valid labels or numeric ranges). JSON mode is enabled for all calls.

Example system prompts: *Classification*: "You are a text classifier. Return a JSON with your reasoning and the predicted label from the provided list." *Retrieval*: "You are a retrieval assistant. Given a query and a set of numbered documents, identify the document IDs most relevant to the query. Return a JSON list of IDs in ranked order."

Schema validation and failure handling. All LLM outputs are validated against the Pydantic schema after generation. If validation fails (e.g., invalid label or malformed JSON), the prompt is retried with an appended correction instruction, up to twice per sample. The final validation failure rate is $<0.3\%$; failed samples receive a null prediction and are counted as incorrect.

Token usage tracking. For each API call we log: `prompt_token_count`, `candidates_token_count`, `cached_content_token_count`, `total_token_count`, and `thoughts_token_count` from `GenerateContentResponse.usage_metadata`. Thinking tokens are verified as `total` − `prompt` − `candidates` to cross-check against `thoughts_token_count`; agreement was exact in all cases. Pricing sources and throughput configurations are documented in §H.4.