EmbeddingGemma 2: Unified and Efficient Multimodal Representations

Henrique Schechter Vera*, Sahil Dua*, Jovan Andonov, Sara Smoot, Sebastian Russo, Ivan Llanos, Max Matveev, Jae Hyeon Yoo, Kat Black, Shih-Cheng Huang, Josiah Aklilu, Maarten Grootendorst, Omar Sanseviero, Chandramouli Amarnath, Ian Ballantyne, Zhe Li, Daniel Salz, Ibrahim Alabdulmohsin, Ilya Philippov, Kristen Quan, Glenn Cameron, Edouard Yvinec, Luciano Martins, Doug Reid, Sindhu Raghuram Panyam, Weiyi Wang, Hengxuan Ying, Madhuri Shanbhogue, Tanmaya Dabral, Shanfeng Zhang, Juyeong Ji, Renjie Wu, Joe Zou, Xu Chen, Yunan Zhang, Yu-Hui Chen, Matthew Soulanille, Vinod Mamillapalli, Yi-Chun Kuo, Wai Hon Law, Matt Kreileder, Alice Zheng, Suril Shah, Mogan Shieh, Pooja Consul, Alberto Montes, Andreas Hess, Andres Sevilla, Babak Samari, Brendan McCloskey, Brian Potetz, Daniel Cer, Dilip Kumar Margam, Feodor Beloborodov, Georgios Kathareios, Gustavo Hernández Ábrego, Hesen Zhang, Ji Ma, Jianan Lu, Jiaxing Wu, Keegan Mosley, Paul Suganthan, Rui Meng, Shen Gao, Shengcao Cao, Wing Lowe, Yi-Ting Chen, Yichang Chen, Zhongli Ding, Kendra Byrne, Cormac Brick, Olivier Lacombe, Qin Yin, Tris Warkentin, Armand Joulin, Tom Duerig and Mojtaba Seyedhosseini
EmbeddingGemma Team, Google[1]

arXiv:2610.37364v2 [cs.LG] 9 Nov 2026

We present EmbeddingGemma 2, an open multimodal embedding model which natively represents text, images, audio, video, and combinations thereof in a unified space. Our approach introduces improvements such as simplified per-layer embeddings and a multi-stage curriculum with modality and objective progression which ensure strong quality across unimodal, cross-modal, and mixed-modal evaluations. Notably, in comprehensive embedding benchmarks across all modalities, EmbeddingGemma 2 establishes a new Pareto frontier, outperforming both specialist models and models two to twelve times larger. Furthermore, our design offers flexibility for resource-constrained environments, supporting selective modality encoder loading, adjustable vision token budgets, truncatable embedding dimensions, and quantized weights. We present ablations analyzing key decisions and release EmbeddingGemma 2 to promote further research.

1. Introduction

Embedding models map data to fixed-length vector representations, placing semantically similar data near each other in the embedding space. Text embedding models have become a popular choice for tasks like information retrieval, semantic similarity, clustering, and retrieval-augmented generation. However, to work with audio and visual data, these require additional tools like automatic speech recognition (ASR) and image captioners which both add computational cost and fail to capture signals like fine-grained visual details and non-verbal or environmental sounds.

Multimodal embedding models like CLIP [1], ALIGN [2], and SigLIP [3] add a modality tower for vision, and so achieve strong cross-modal quality, but they are unable to natively process mixed-modal input to capture interactions between input modalities. Recent efforts have adapted Multimodal Large Language Models (MLLMs) into embedding models that allow interleaved mixed-modal input and support more complex tasks such as visual question answering and visual grounding [4, 5, 6, 7]; subsequent work has built on more capable MLLMs to incorporate video and audio modalities [8, 9, 10, 11, 12]. However, as MLLMs have grown in size, so have embedding models adapted from them. These models typically contain several billion parameters, making them prohibitively expensive for applications that are latency-sensitive, high-throughput, or require on-device deployment for privacy or offline access.

[1]See Contributions and Acknowledgements section. *Co-first authors; equal contributions.

Corresponding author: hschechter@google.com
© 2026 Google DeepMind. All rights reserved

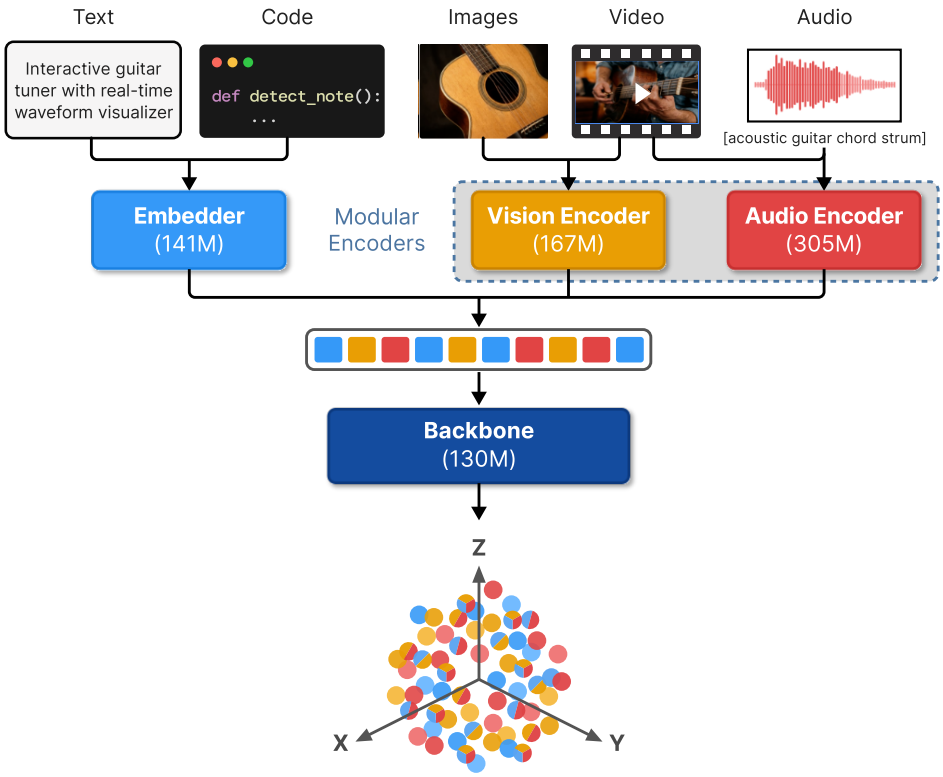


Figure 1 | EmbeddingGemma 2 natively maps text, images, video, audio, and combinations thereof into a unified, semantically-aligned vector space.

In this paper, we introduce EmbeddingGemma 2,[2] a 744M-parameter general-purpose multimodal embedding model that maps text, images, video, and audio—as well as arbitrary interleaved combinations thereof—into a single unified embedding space (Figure 1). EmbeddingGemma 2 combines a 271M-parameter bidirectional transformer backbone (embedder included) with a 167M-parameter vision encoder and a 305M-parameter audio encoder. The modality encoders independently convert media into sequences of soft token embeddings that enter the shared transformer backbone, which means these parameters are modular: EmbeddingGemma 2 can alternatively be run with 271M parameters as a text-only model, with 439M as a vision-language model, or with 577M as an audio-language model.

The model builds on the architecture and weights from the capable Gemma 4 model family [13] and learns from the powerful Gemini Embedding 2 model [8]. Our training regime consists of three stages spanning various tasks, modalities, and domains, which results in a broadly capable model, excelling at problems like code search, visual question answering, video moment retrieval, audio classification, and much more. We separate text-only and multimodal training as a form of curriculum learning, and separate distillation from contrastive and regularization objectives in order to effectively learn an embedding space tailored to the model. For extreme resource-constrained settings, the model offers high-quality truncated embeddings and separate checkpoints adapted with quantization-aware training.

We present ablation studies explaining our architectural and training choices. We find that simplifying per-layer embeddings to omit the dedicated lookup table not only reduces cost but improves performance; we present the effects of using different vision token budgets during training

[2]Our model is available at https://ai.google.dev/gemma/docs/embeddinggemma.

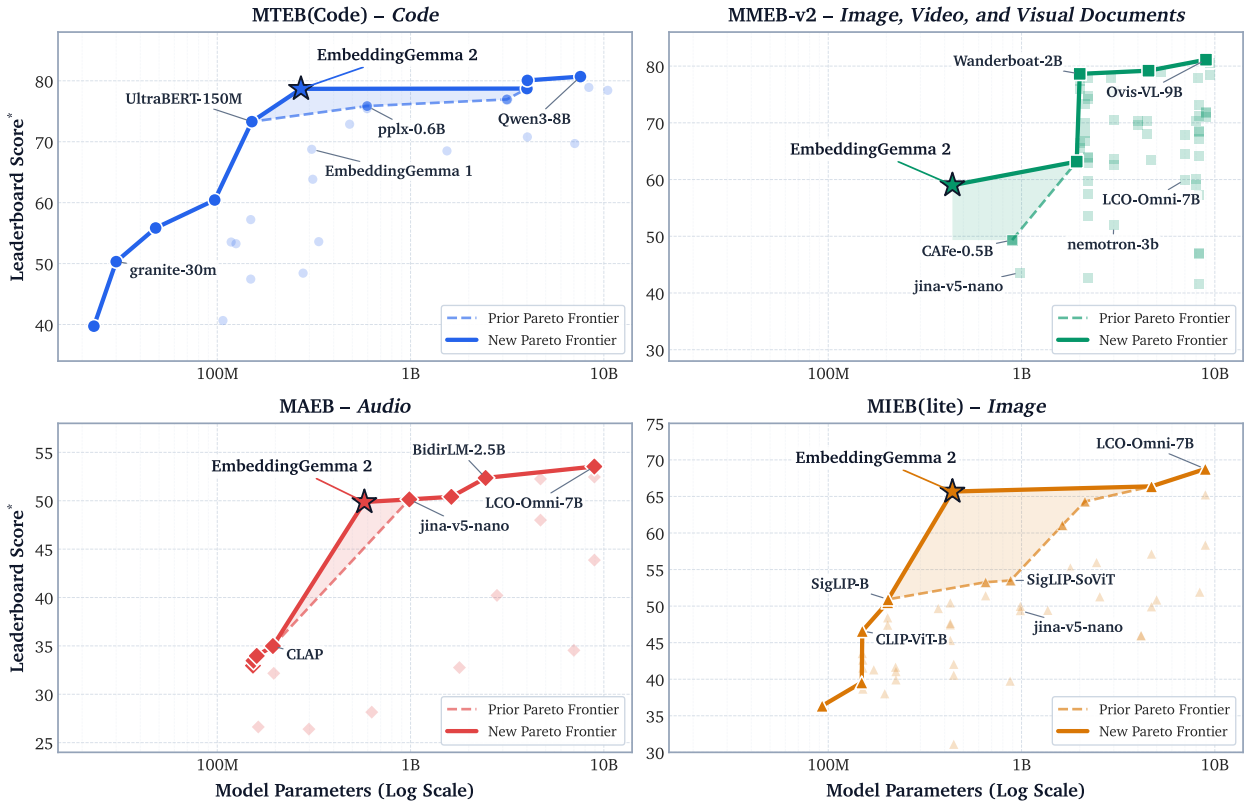


Figure 2 | EmbeddingGemma 2 establishes a new Pareto frontier across benchmarks covering code, image, video, visual documents, and audio. Methods and detailed results in Section 4.
*Mean(Task) for MTEB(Code), MAEB, and MMEB-v2; Mean(Type) for MIEB(lite).

and inference, observing it only significantly affects tasks like document retrieval which require fine-grained visual understanding; we demonstrate the importance of co-training the vision encoder; we show the impact of finetuning with contrastive learning rather than distillation; and we confirm the value of native audio understanding over ASR.

Finally, we evaluate the model on seven major, comprehensive benchmark suites spanning text, code, image, video, document, speech, and sound inputs. EmbeddingGemma 2 establishes a new Pareto frontier across benchmarks spanning all modalities (Figure 2). Despite using fewer parameters for text than its predecessor and handling more modalities, it improves on both multilingual text and code benchmarks. On multimodal evals, EmbeddingGemma 2 often matches or exceeds the quality of specialized and multi-billion-parameter models.

2. EmbeddingGemma 2

2.1. Architecture

Backbone The backbone follows a transformer architecture [14] with bidirectional attention. Given a tokenized input sequence $\mathbf{T}$ of length L, we map discrete text tokens via an embedder $\mathcal{E}$ into input token embeddings $\mathbf{T}_{\text{in}} = \mathcal{E}(\mathbf{T}) \in \mathbb{R}^{L \times d_M}$, where d_M is the model dimension used for the transformer's inner representations (note multimodal inputs are incorporated into $\mathbf{T}_{\text{in}}$ as well, described in detail in Section 2.2). We then apply $\mathcal{M}_n$, an n-layer transformer with bidirectional attention, to produce a sequence of output token embeddings $\mathbf{T}_{\text{out}} = \mathcal{M}_n(\mathbf{T}_{\text{in}}) \in \mathbb{R}^{L \times d_M}$. Next, we employ mean pooling

$\mathcal{P}$ to average the output token embeddings along the sequence axis to generate a single embedding representing all the information in the input, producing $\mathbf{P}_{\text{embed}} = \mathcal{P}(\mathbf{T}_{\text{out}}) \in \mathbb{R}^{d_M}$. Finally, we apply a randomly initialized linear projection f to scale the embedding to the target dimension d, followed by ℓ_2-normalization to produce the final unit-norm embedding, $\mathbf{E} = \frac{f(\mathbf{P}_{\text{embed}})}{\|f(\mathbf{P}_{\text{embed}})\|_2} \in \mathbb{R}^d$. In EmbeddingGemma 2, we use $n = 24, d_M = 512, d = 768$.

Notably, the transformer $\mathcal{M}_n$ harnesses per-layer embeddings (PLE), as introduced in Gemma 3n [15]. These act as residual inputs which give each layer a path to receive tailored information directly from the input token embeddings. However, instead of additionally utilizing a large, separate token lookup table specifically for PLE, EmbeddingGemma 2 simply adapts the input embeddings $\mathbf{T}_{\text{in}}$ produced by the main embedding table and modality encoders (comparison in Section 3.1). For each layer $l \in [n]$, we project the input token embeddings into a layer-specific representation $\mathbf{P}_l = \text{RMSNorm}(p_l(\mathbf{T}_{\text{in}})) \in \mathbb{R}^{L \times d_P}$ via linear projection p_l. We use the layer's output $\mathbf{H}'_l \in \mathbb{R}^{L \times d_M}$ to compute gate $\mathbf{G}_l = \sigma(g_l(\mathbf{H}'_l)) \in \mathbb{R}^{L \times d_P}$, where g_l is a linear projection and σ is a GELU activation [16]. We then modulate the per-layer input and project it back to the model dimension via linear projection u_l, producing update $\mathbf{U}_l = \text{RMSNorm}(u_l(\mathbf{G}_l \odot \mathbf{P}_l)) \in \mathbb{R}^{L \times d_M}$, to finally yield $\mathbf{H}_l = \mathbf{H}'_l + \mathbf{U}_l$ to be processed by the next layer. In EmbeddingGemma 2, we use $d_P = d_M = 512$.

Modality Encoders As in Gemma 4, the vision encoder is a vision transformer [ViT, 17] with a patch size of 16 and the audio encoder consists of two downsampling convolution layers followed by twelve conformer layers [18], based on the Universal Speech Model (USM) architecture [19]. The vision and audio encoders produce soft tokens with dimensionalities $d_V = 768$ and $d_A = 1536$, respectively. Using learned linear projections ϕ_{vis} and ϕ_{aud}, we map these to the model dimension d_M. These soft tokens are interleaved with the token embeddings at their corresponding positions to form a combined input token embedding sequence $\mathbf{T}_{\text{in}} \in \mathbb{R}^{L \times d_M}$, which is processed as described above. See Section 2.2 below for more details on how multimodal inputs are handled.

2.2. Input

In general, an input x consists of an ordered sequence of $M \geq 1$ modality segments $x = (s_1, s_2, \ldots, s_M)$, where each segment s_m belongs to one of four modalities:

1. **Text** is tokenized into N_{txt} tokens which are mapped via the embedding table to a sequence of token embeddings: $\mathbf{T}_{\text{txt}} = \mathcal{E}(s_m) \in \mathbb{R}^{N_{\text{txt}} \times d_M}$.
2. **Images** are each processed by the vision encoder $\mathcal{V}$ and spatially pooled to below a fixed budget of N_{vis} soft tokens, $\mathbf{T}'_{\text{vis}} = \mathcal{V}(s_m) \in \mathbb{R}^{N_{\text{vis}} \times d_V}$. N_{vis} defaults to 280 for images and 140 for video during training, but is configurable to 70, 140, 280, 560, or 1120 at inference time—see how this impacts quality in Section 3.4. These soft tokens are then projected by the vision adapter and wrapped with delimiter tokens: $\mathbf{T}_{\text{vis}} = \texttt{<|SOI|>} \oplus \phi_{\text{vis}}(\mathbf{T}'_{\text{vis}}) \oplus \texttt{<|EOI|>} \in \mathbb{R}^{(N_{\text{vis}}+2) \times d_M}$.
3. **Audio** is processed in 40ms chunks with Mel filterbank inputs, producing 25 soft tokens per second of audio: $\mathbf{T}'_{\text{aud}} = \mathcal{A}(s_m) \in \mathbb{R}^{N_{\text{aud}} \times d_A}$. These are projected by the audio adapter and wrapped with different delimiter tokens to produce: $\mathbf{T}_{\text{aud}} = \texttt{<|SOA|>} \oplus \phi_{\text{aud}}(\mathbf{T}'_{\text{aud}}) \oplus \texttt{<|EOA|>} \in \mathbb{R}^{(N_{\text{aud}}+2) \times d_M}$.
4. **Video** clips are typically sampled at 1 frame per second to begin. Then, for longer videos, frames are further sampled to fit a per-video frame budget N_{frm}—this is sometimes via truncation, by selecting the first N_{frm} frames, and sometimes via uniform sampling, by selecting N_{frm} evenly spaced frames over the entire video. During our training and evaluations we use $N_{\text{frm}} = 32$. Since each frame is processed by the vision encoder as an image, this results in up to N_{frm} segments, each with at most N_{vis} soft tokens and two image delimiter tokens.

Optionally, frames can be interleaved with audio tokens from the video's soundtrack, with each frame coming first and then the audio that follows it divided into 1-second segments. Usually, this corresponds to a single 1-second audio segment per frame, but if frames are truncated we might have multiple trailing audio segments, and if frames are uniformly sampled, we might have multiple audio segments between frames. In interleaved video-audio, we include text timestamps in `MM:SS` format both before each frame and before each audio segment, to indicate when the frame is taken and when the audio chunk begins. Thus, video is represented as a combination of the former modalities, yielding an embedding sequence $\mathbf{T}_{\text{vid}}$ composed of interleaved sequences of $\mathbf{T}_{\text{txt}}$, $\mathbf{T}_{\text{vis}}$, and $\mathbf{T}_{\text{aud}}$.

Then, denoting as $\mathbf{T}_m \in \{\mathbf{T}_{\text{txt}}, \mathbf{T}_{\text{vis}}, \mathbf{T}_{\text{aud}}, \mathbf{T}_{\text{vid}}\}$ the embedding sequence of segment s_m, the final concatenated sequence is $\mathcal{T}(x) = \texttt{<|BOS|>} \oplus \mathbf{T}_1 \oplus \mathbf{T}_2 \oplus \cdots \oplus \mathbf{T}_M \oplus \texttt{<|EOS|>} \in \mathbb{R}^{L \times d_M}$, with a total sequence length L of up to 8192 during training. This is consumed by the model as $\mathbf{T}_{\text{in}}$ in Section 2.1.

Each training example includes multiple inputs: a query q_i, a positive target (or *passage*) p_i^+, and (optionally) hard negative targets $\{p_i^-\}$. An example may also include prescribed task prefixes t_q and t_p for queries and targets respectively, describing the nature of the task (see Appendix A). For instance, for retrieval we use "*task: search result | query:* {text}" and "*title:* {title | 'none'} | *text:* {text}". The query and targets are embedded as vectors in $\mathbb{R}^d$:

$$\mathbf{q}_i = f(\mathcal{P}(\mathcal{M}_n(\mathcal{T}(t_q \oplus q_i)))), \quad \mathbf{p}_i^{\pm} = f(\mathcal{P}(\mathcal{M}_n(\mathcal{T}(t_p \oplus p_i^{\pm})))). \tag{1}$$

2.3. Objective

EmbeddingGemma 2 was trained using multiple losses. Here we introduce the loss functions themselves, and in Section 2.4 we explain how they are introduced during training.

Contrastive loss The first is a noise-contrastive estimation (NCE) loss with in-batch, hard, and same-tower negatives. Given a batch of size B, we define the contrastive loss as:

$$\begin{aligned}
\mathcal{L}_C &= -\frac{1}{B}\sum_{i=1}^{B} \log \frac{Z_i^{\text{P}}}{Z_i^{\text{P}} + Z_i^{\text{HN}} + Z_i^{\text{IB}} + Z_i^{\text{STN}}} \\
Z_i^{\text{P}} &= e^{\text{sim}(\mathbf{q}_i, \mathbf{p}_i^+)/\tau} \\
Z_i^{\text{HN}} &= \sum_{\mathbf{p}_i^-} e^{\text{sim}(\mathbf{q}_i, \mathbf{p}_i^-)/\tau} \\
Z_i^{\text{IB}} &= \sum_{j=1}^{B} \mathbb{1}_{\text{TN}}(i,j)\, e^{\text{sim}(\mathbf{q}_i, \mathbf{p}_j^+)/\tau} \\
Z_i^{\text{STN}} &= \sum_{j=1}^{B} \mathbb{1}_{\text{TN}}(i,j)\, \left(e^{\text{sim}(\mathbf{q}_i, \mathbf{q}_j)/\tau} + e^{\text{sim}(\mathbf{p}_i^+, \mathbf{p}_j^+)/\tau}\right)
\end{aligned} \tag{2}$$

where $\text{sim}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y} / \|\mathbf{x}\| \|\mathbf{y}\|$ is cosine similarity. Above, Z_i^{P} corresponds to the positive, Z_i^{HN} corresponds to the hard negatives, Z_i^{IB} corresponds to in-batch negatives, and Z_i^{STN} corresponds to same-tower negatives [20]. $\mathbb{1}_{\text{TN}}$ masks out false negatives from duplicates:

$$\mathbb{1}_{\text{TN}}(i,j) = \begin{cases} 0 & \text{if } \mathbf{q}_i = \mathbf{q}_j \text{ or } \mathbf{p}_i^+ = \mathbf{p}_j^+, \\ 1 & \text{otherwise,} \end{cases} \tag{3}$$

We forego the hardness weighting mechanism from our previous model as it amplifies noisy or weakly associated positive pairs, especially during early training, and compounds gradient scale imbalances between tasks of different modalities. We add same-tower negatives to prevent queries and targets from clustering into disjoint subspaces.

For listwise ranking data with P ordered positives $(\mathbf{p}^+_{i,1} \succ \cdots \succ \mathbf{p}^+_{i,P})$ per query, we train both on extracted triplets using $\mathcal{L}_C$ and on the full ranked lists using a listwise loss that extends $\mathcal{L}_C$ based on the Plackett–Luce model, as in Xia et al. [21]:

$$\begin{aligned} \mathcal{L}_C^{\text{list}} &= -\frac{1}{BP}\sum_{i=1}^{B}\sum_{r=1}^{P}\log\frac{Z^{\text{P}}_{i,r}}{\sum_{k=r}^{P} Z^{\text{P}}_{i,k} + Z^{\text{HN}}_i + Z^{\text{IB}}_i} \\ Z^{\text{P}}_{i,r} &= e^{\text{sim}(\mathbf{q}_i,\mathbf{p}^+_{i,r})/\tau}, \qquad Z^{\text{IB}}_i = \sum_{j=1}^{B}\mathbb{1}_{\text{TN}}(i,j)\Big(\sum_{k=1}^{P} e^{\text{sim}(\mathbf{q}_i,\mathbf{p}^+_{j,k})/\tau} + \sum_{\mathbf{p}^-_j} e^{\text{sim}(\mathbf{q}_i,\mathbf{p}^-_j)/\tau}\Big) \end{aligned} \tag{4}$$

Here, we contrast each rank-r positive against all lower-ranked positives ($k > r$) in addition to hard and in-batch negatives, with $\mathbb{1}_{\text{TN}}(i,j)$ adapted to mask out in-batch negatives that duplicate $\mathbf{p}^+_{i,r}$. We also add small squared-hinge penalties on positive and hard-negative similarities to calibrate the absolute score distribution for threshold-based tasks.

Spread-out regularizer The second loss is based on the global orthogonal regularizer (GOR) introduced in Zhang et al. [22]. The idea is to make the embeddings of a random pair of inputs have similar statistical properties (mean and second moment) as two points independently and uniformly sampled from the unit sphere. Following EmbeddingGemma 1 [23], we use only the "second moment" term of the regularizer's original definition, as we find this also sufficiently pushes the "mean" term towards its target value:

$$\mathcal{L}_S = \frac{1}{B\,(B-1)}\sum_{1\le i\ne j\le B}(\mathbf{q}_i^\top \mathbf{q}_j)^2 + \frac{1}{B\,(B-1)}\sum_{1\le i\ne j\le B}(\mathbf{p}_i^{+\top}\mathbf{p}_j^+)^2. \tag{5}$$

This loss encourages the model to produce embeddings that are spread out over the embedding space, to fully utilize its expressive power. Embedding spaces have been observed to form narrow cones [24, 25, 26, 27], but there are theoretical and empirical benefits to isotropy [28, 29], including performance in downstream tasks. Note that pushing random pairs toward orthogonality complements the squared-hinge penalties above to make scores more calibrated and interpretable. Enforcing isotropy also aims to ensure that a) the model is robust to weight and embedding quantization, and b) the embeddings can be retrieved efficiently in vector databases using approximate nearest neighbor (ANN) algorithms.

Distillation loss The final is an embedding matching loss based on Kim et al. [30]. In contrast to previous distillation research which has relied only on the teacher's query-document relevance scores as a signal [31, 32], this loss directly aligns EmbeddingGemma 2's embedding space with that of a teacher model, allowing it to more effectively learn from the larger, more powerful Gemini Embedding 2 model. Unlike Kim et al. [30], which applies a projection layer to distill from a model with different dimensionality, we instead make use of Gemini Embedding 2's support for lower dimensions and simply truncate the teacher embeddings. As in Schechter Vera et al. [23], we apply embedding matching not only to queries and targets but also to hard negative targets, which intuitively serves to transfer the teacher's ability to discriminate between queries and their hard negatives, and empirically

does improve performance. We weigh these individual losses uniformly, letting

$$\mathcal{L}_D = \mathcal{L}_D^{\mathbf{q}} + \mathcal{L}_D^{\mathbf{p}^+} + \mathcal{L}_D^{\mathbf{p}^-}, \quad \text{where} \quad \mathcal{L}_D^{\mathbf{u}} = \frac{1}{B}\sum_{i=1}^{B} \left\| \mathbf{u}_i - \tilde{\mathbf{u}}_i \right\|_2^2 \quad \text{for } \mathbf{u} \in \{\mathbf{q}, \mathbf{p}^+, \mathbf{p}^-\}, \tag{6}$$

and $\tilde{\mathbf{u}}_i$ denotes the truncated, ℓ_2-normalized teacher embedding for input $\mathbf{u}_i$ (with $\mathcal{L}_D^{\mathbf{p}^-}$ averaged over hard negatives when multiple $\mathbf{p}_i^-$ are present).

Matryoshka Representation Learning We adapt the contrastive, spread-out, and distillation losses with Matryoshka Representation Learning (MRL) [33], splitting each loss into k separate losses applied to k overlapping sub-dimensions of the embedding. EmbeddingGemma 2 provides 768-dimensional embeddings, additionally supporting 512-, 256-, and 128-dimensional embeddings via MRL.

2.4. Recipe

Initialization EmbeddingGemma 2 builds directly on Gemma 4's extensive world knowledge and capabilities. The backbone is adapted from a pretrained Gemma 4 decoder-only model, switching the causal attention mask to bidirectional attention to enable fully contextual token representations [34, 35, 36, 8]. We omit explicit adaptation stages for bidirectional attention [23, 37], which can be costly and have been shown to be unnecessary for learning expressive representations [23, 38]. We also remove the PLE-specific token lookup table (detailed in Section 3.1), and add mean pooling [39, 23], a linear projection, and ℓ_2-normalization after the transformer. The vision encoder is initialized with the smaller Gemma 4 vision encoder and further trained in the two multimodal stages, which we show in Section 3 to be critical for quality. The audio encoder is bit-identical to that in the Gemma 4 E2B and E4B models [13] and kept frozen during training.

We then train EmbeddingGemma 2 on a mixture of embedding tasks spanning various modalities and task types. In total, the model sees approximately 2.9T tokens, of which ~300B are seen during text pre-finetuning, ~2T during multimodal pre-finetuning, and ~600B during multimodal finetuning.

Text Pre-finetuning We begin with a pre-finetuning stage like the one in Schechter Vera et al. [23]. This entails a large-scale unsupervised data mixture of query-target pairs from question answering, sentence similarity, code retrieval, and web search tasks using various languages. As before, we opt for a larger batch size both to stabilize the gradient and to provide more (in-batch) negatives for each input example. In this stage we almost exclusively employ distillation loss, with only an additional contrastive loss using a small weight. Here, as with subsequent stages, tasks are sampled at pre-determined, empirically defined rates. We found the inclusion of a text-only stage important for the model's final text quality. This stage can be seen as a form of curriculum learning, or as an implicit warm-up for the bidirectional attention, similar to BehnamGhader et al. [37]; it has also been shown to enhance the discriminability of both text and non-text embeddings [10].

Multimodal Pre-finetuning We then continue with pre-finetuning, but with a new mixture which also includes inputs with images, audio, video, and documents across unimodal, cross-modal, and mixed-modal tasks, most with additional hard negatives. The text tasks in this stage and the next include those described in the finetuning stage of Schechter Vera et al. [23], which aim to cover task diversity, language diversity, and coding capability. In this stage, we exclusively employ distillation loss, to inherit the teacher's ability to map diverse and mixed modalities into a shared semantic space. We continue training with larger batch sizes and larger learning rates, but train for twice as many steps as in text pre-finetuning to cover the expanded multimodal training set.

		MTEB(Multilingual, v2)		MTEB(English, v2)		MTEB(Code)	MIEB(lite)‡		MSEB(Retrieval)‡		
Model	Precision	Mean(Task)	Mean(Type)	Mean(Task)	Mean(Type)	Mean(Task)	Mean(Task)	Mean(Type)	Overall	In-Lang	Cross-Lang
EG 2	bfloat16	61.36	53.82	68.46	63.86	78.68	60.17	65.67	69.54	71.63	67.68
	int4 (Q4_0)*	60.96	53.47	68.48	63.91	76.85	56.59	61.95	64.99	68.56	61.80
	int4 (Q4_C)†	60.84	53.36	68.49	63.93	76.70	56.71	62.26	63.92	67.93	60.33
EG 1	bfloat16	61.15	54.31	69.67	65.11	68.76	—	—	—	—	—
	int4 (Q4_0)*	60.62	53.61	69.31	64.65	67.99	—	—	—	—	—

Table 1 | Comparison of raw and quantized EmbeddingGemma 2 and EmbeddingGemma 1 checkpoints across benchmarks covering text, vision, and audio: MTEB, MIEB(lite), and MSEB(Retrieval). *Per-block int4. †Per-channel int4. ‡Different eval harnesses between float and quantized runs.

Multimodal Finetuning In the finetuning stage, we completely replace distillation with contrastive loss. This allows the model to learn a tailored embedding space independent of its teacher, as the teacher's embedding space is not optimized for the student's architecture, size, and MRL sub-dimensions. We also add other losses which aim to shape the final embedding space to have desirable properties: spread-out regularizer and MRL losses. As in Schechter Vera et al. [23], we transition to smaller learning rates and batch sizes, and shift towards higher-quality, task-specific datasets, such as tool retrieval, audio captioning, video ranking, medical imaging, visual product search, cropped-object retrieval, and academic question-answering.

Model Merging Finally, we combine checkpoints from our finetuning run by averaging their weights, to improve our final model's quality and robustness. Rather than running multiple finetuning jobs, either with different data mixture rates [23] or with different hyperparameters [40], we merge checkpoints along a single finetuning trajectory [41]. Specifically, we compute an unweighted average between the final checkpoint from the finetuning run and a significantly earlier checkpoint from the same run. Like Wortsman et al. [42], which similarly merges pretrained and finetuned checkpoints, we found that this helps mitigate degradation on tasks that overfit during finetuning, while also further improving performance on tasks that already benefited from finetuning.

Quantization-Aware Training We additionally provide quantized versions of EmbeddingGemma 2 in different quantization configurations. We provide checkpoints with two types of weight representations for the backbone: int4 per-block (Q4_0) and int4 per-channel (Q4_C). For the modality encoders, we follow the quantization schemes used in Gemma 4: the vision encoder uses 8-bit precision for both weights and activations (W8A8), and the audio encoder uses 8-bit activation precision and {2, 4, 8}-bit weight precision varying by layer cluster.

We apply quantization-aware training [QAT, 43] to both the backbone and vision encoder to minimize quality degradation (the audio encoder is taken directly from Gemma 4). However, instead of applying QAT directly in the finetuning stage as in Schechter Vera et al. [23], we introduce a separate post-merging tail-patch stage in which we distill (and initialize) our QAT checkpoint from the frozen float checkpoint [44]. This allows for a more resource-efficient QAT stage and narrows the gap in quality between the float and quantized checkpoints for some evaluations, notably MTEB(English, v2). Table 1 shows the quality is comparable between our quantized checkpoints and our raw checkpoint (evaluated at half-precision bfloat16 weights). With Q4_C quantization, the peak on-device memory footprint during inference (including runtime compilation overhead) on a Google Pixel 11 Pro is reduced to 191 MB for text-only, 447 MB for text and vision, and 567 MB for the full multimodal model.

PLE Strategy	**MMEB-v2 Overall**	Image (Hit@1)	Video (Hit@1)	VisDoc (NDCG@5)
Projection-only	**55.46**	**54.67**	**47.82**	**62.37**
With lookup	55.07	54.57	47.13	61.76

Table 2 | Results with and without a dedicated Per-Layer Embeddings (PLE) lookup table.

Training Stage	**MMEB-v2 Overall**	Image (Hit@1)	Video (Hit@1)	VisDoc (NDCG@5)
Multimodal Pre-finetuning	52.27	50.10	43.26	62.26
+ Extra Pre-finetuning	52.28	49.74	43.86	**62.40**
+ Finetuning Stage	**55.46**	**54.67**	**47.82**	62.37

Table 3 | Results of continued distillation versus adding a finetuning stage with contrastive learning.

3. Ablation Studies

We analyze some key design choices to better understand how they contributed to EmbeddingGemma 2's ability to produce high-quality embeddings across modalities, domains, and task types. Baseline and ablation models use an intermediate recipe and so numbers are not representative of the final model. Unless otherwise stated, the experiments run finetuning on a pre-finetuned model. We evaluate the models on MMEB-v2 as it covers various tasks involving combinations of text, images, documents, and video, excluding only audio.

3.1. Per-Layer Embeddings Table

Gemma 3n and Gemma 4 incorporate Per-Layer Embeddings (PLE) to inject token-level information into intermediate transformer layers. Specifically, they combine a learned projection of the main token embedding with an additional dedicated PLE lookup table. However, a dedicated PLE lookup table provides no direct signals for non-text modalities and consumes additional memory.

We utilize only the projection, as described in detail in Section 2.1; this way, we continue providing per-layer inputs but can omit the lookup table. We compare our approach to PLE with a lookup table in Table 2. Removing the dedicated PLE table not only reduces parameter overhead, but also strengthens performance across all three categories (notably +0.69 on Video and +0.61 on VisDoc), confirming that deriving per-layer inputs solely from the shared d_M-dimensional embedding space improves multimodal representation quality. We also found similar improvements in text-only evaluations, where one might expect the lookup table to help the most.

3.2. Finetuning Stage

We investigate whether the gains from our finetuning stage are truly driven by a difference in recipe or merely by extending training. Starting from the multimodal pre-finetuning checkpoint, we compare running the multimodal finetuning stage against continuing distillation for an approximately compute-equivalent duration (the same number of steps, but with larger batch sizes on distillation). Table 3 shows that while distillation plateaus, transitioning to the finetuning recipe with contrastive and spread-out objectives boosts performance significantly for image and video tasks while yielding similar results for visual documents. This demonstrates that the finetuning stage enables the model to move beyond imitating its teacher and learn a better-optimized representation space.

Vision Encoder	**MMEB-v2 Overall**	Image (Hit@1)	Video (Hit@1)	VisDoc (NDCG@5)
Trainable	**52.27**	**50.10**	**43.26**	**62.26**
Frozen	48.18	44.35	39.24	60.59

Table 4 | Results with trainable and frozen vision encoder.

Experiment	Vision Token Budget		**MMEB-v2 Overall**	Image (Hit@1)	Video (Hit@1)	VisDoc (NDCG@5)
	Train (Img / Vid)	Eval				
Inference-Time Scaling	280 / 140	70	50.07	50.90	46.33	51.61
	280 / 140	140	55.37	53.58	47.99	63.60
	280 / 140	280	56.15	55.15	**48.51**	63.38
	280 / 140	560	**57.02**	**55.21**	46.52	**67.62**
Train-Time Scaling	280 / 70	70	49.62	50.56	**46.60**	50.49
	280 / 140	70	**49.76**	**50.76**	45.79	**51.25**
	280 / 70	140	54.66	53.29	47.16	62.33
	280 / 140	140	**55.46**	**54.67**	**47.82**	**62.37**

Table 5 | Results with different vision token budgets (max soft tokens per image or video frame) at train and inference time. Evaluation uses a single budget for all tasks. **Top:** Inference-time scaling of a model trained with 280 tokens per image and 140 per video frame. **Bottom:** Train-time scaling with 70 vs. 140 tokens per video frame. The two experiments use different intermediate recipes.

3.3. Vision Encoder Training

MLLMs often keep modality encoders frozen throughout training [45, 46, 47, 13]. This strategy is also popular in recent multimodal embedding models [48, 9, 49, 4]. For MLLMs, ablations investigating whether freezing or unfreezing the vision encoder is better are mixed [50, 45, 51]. However, we show in Table 4 that, though the Gemma 4 vision encoder is quite strong by itself, training the vision encoder greatly improves performance of the resulting embeddings on multimodal tasks. Analogous gains are likely possible by further training the audio encoder. Note that both runs here start with a text pre-finetuning checkpoint and run multimodal pre-finetuning, to ensure the vision encoder has not already been updated prior to either experiment.

Some MLLMs manage to additionally keep the backbone frozen [52]. Hönicke et al. [49] similarly joins a backbone text embedding model with modality encoders, keeping each of them frozen. This prevents users from having to backfill embedding databases created with the text embedding model; however, it is possible to translate embeddings from one model's space to another [53], even without access to paired data or to the original models [54]. Moreover, freezing a text-only backbone empirically seems to limit its ability to understand multimodal inputs, and indeed many of the strongest multimodal embedding models finetune the backbone alongside the modality encoders [12, 10].

3.4. Vision Token Budgets

We train EmbeddingGemma 2 with a budget of 280 vision tokens for images and 140 tokens per frame for videos, but the underlying Gemma 4 vision encoder supports budgets from 70 to 1120 tokens. In Table 5 (top), we show the effect of using different vision token budgets at inference time using an intermediate checkpoint trained with 280 tokens per image and 140 per frame. Overall performance scales monotonically, with some distinct modality-specific behaviors. Visual document

tasks are highly sensitive to fine-grained details and continue to improve greatly past the training budget; image quality in aggregate begins to saturate at the 280-token budget (despite continued gains on several individual tasks); and video performance actually degrades between 280 tokens and 560, likely because the excessive sequence lengths dilute attention and mean-pooled representations.

Additionally, we compare models trained with 70 vs. 140 per-frame video tokens (Table 5, bottom) to see whether training with a smaller vision budget could improve quality when using smaller vision budgets at inference time, for resource-constrained deployment. When evaluated with 70 tokens, training with 70 video tokens improves performance on video tasks, as the model is directly exposed to 70-token frames during training. But this advantage does not hold for image or visual document tasks, which drop in quality. This is likely because a) single-image tasks, which continue to be trained with 280 tokens, are treated as a separate modality and do not benefit from seeing 70-token video frames; and b) coarser video representations discard spatial details that otherwise transfer to stronger general visual representations. The latter is further supported when evaluating with 140 tokens, where training with 70 video tokens degrades performance even more, and across modalities—this makes the 140-token video budget preferable across both evaluation regimes. More gains are likely possible by increasing train-time budgets, but with seemingly diminishing returns and at a significant computational cost.

4. Evaluation

We evaluate EmbeddingGemma 2 on a comprehensive set of benchmarks across diverse modalities, task types, domains, and languages. Specifically, we report numbers from the Massive Text Embedding Benchmark [MTEB, 55, 56] for natural language and code; the Massive Multimodal Embedding Benchmark v2 [MMEB-v2, 57] and Massive Image Embedding Benchmark [MIEB(lite), 58] for image, video, and visual documents; and the Massive Sound Embedding Benchmark and Massive Audio Embedding Benchmark for speech, music, and environmental sounds [MSEB and MAEB, 59, 60]. We provide a summary of our results in Table 6 and dive into each benchmark in more detail below.

4.1. Methods

We select as baselines the top-ranked models on each benchmark, including both modality specialists and general-purpose embedding models which support all evaluated modalities. For MTEB, MAEB, and MIEB(lite), we compare against models under 1B parameters; for MMEB-v2, we include models under 2B parameters, as small models are scarce. Because it is one of the most directly comparable models, we evaluate `jina-embeddings-v5-omni-nano` on MMEB-v2 using its official implementation and verify that, on the subset reported in Hönicke et al. [49], our reproduced score is slightly higher. All other baseline scores are sourced from public leaderboards. We restrict our analysis to single-vector models, given the disparity in storage and latency costs with multi-vector models, and omit models trained on >25% of the evaluation data, as data contamination undermines the evaluation's validity.

We evaluate EmbeddingGemma 2 using half-precision weights (bfloat16) and the seven types of task instructions detailed in Appendix A. We use different sequence lengths depending on the modality and task type. For visual inputs, we use up to $N_{\text{frm}} = 32$ frames per video at 140 vision tokens per frame and 280 vision tokens per image to reflect realistic model use, though we note that higher vision token budgets could yield further gains, particularly for visual documents (Section 3.4). Because most baseline models support limited modalities, we count only the parameters EmbeddingGemma 2 utilized for each benchmark; other models' parameter counts are as reported on respective leaderboards.

		Embedding Gemma 2	Embedding Gemma 1	Jina Embed v5 Omni Nano	Qwen3 Embed 0.6B	InterestFM UIR CAFe 0.5B	SigLIP SoViT 400M	LAION CLAP General
Total Parameters		744M	308M	986M	596M	894M	878M	194M
Embedding Dimension		768	768	768	1024	896	1152	512
MTEB(Multi, v2) [55]	Mean (Task)	61.36	61.15	**65.52**	64.34	–	–	–
	Mean (Type)	53.82	54.31	**57.66**	56.01	–	–	–
MTEB(Eng, v2) [55]	Mean (Task)	68.46	69.67	**71.11**	70.47	–	–	–
	Mean (Type)	63.86	65.11	**65.22**	64.72	–	–	–
MTEB(Code) [55]	Mean (Task)	**78.68**	68.76	–	75.42	–	–	–
MMEB-v2 [57]	Mean (Task)	**59.01**	*	43.61	*	49.33	–	*
	- Image	**57.28**	*	31.56	*	55.43	–	*
	- Video	**50.67**	*	30.73	*	35.87	–	*
	- VisDoc	67.84	*	**71.34**	*	50.26	–	*
MIEB(lite) [58]	Mean (Task)	**60.17**	*	44.24	*	–	52.28	*
	Mean (Type)	**65.67**	*	49.38	*	–	53.50	*
MAEB [60]	Mean (Task)	49.87	*	**50.14**	*	*	*	34.98
	Mean (Type)	48.65	*	**55.31**	*	*	*	41.08
MSEB(Retrieval) [59]	Mean (Task)	**69.54**	50.95†	–	*	*	*	–

Table 6 | Comparison on benchmarks against a slightly larger multimodal peer and modality specialists. * Modality not supported. – Score unavailable. †Computed by transcribing audio with ASR.

4.2. Text

The MTEB suite consists of a massive collection of individual, quality-controlled evaluation tasks. Together, the 162 tasks cover 250+ natural languages and 14+ programming languages, 20 domains (e.g. legal, medical, news, programming, web), and 10 task types: bitext mining, classification, clustering, instruction retrieval, multilabel classification, pair classification, reranking, retrieval, semantic text similarity (STS), and summarization.

Despite having fewer text parameters and sharing its representation space with vision and audio, EmbeddingGemma 2 improves over its predecessor on multilingual text (driven by gains in bitext mining and classification). On English, the model sees a modest regression. Only two smaller models surpass it on multilingual text, and one on English, but they are all trained as specialist text models (Appendix B). The model presents competitive text scores which outperform much larger models.

On code (Table 7), EmbeddingGemma 2's performance is stellar, establishing a new Pareto frontier. It outperforms every sub-4B model on the leaderboard and even comes within 2 points of the 28× larger frontrunner. In all three benchmarks, the model's truncated representations degrade minimally, especially down to 256 dimensions.

Model Name	Params	Dim.	**Mean (Task)**	AppsR.	COIR	CESR	CFMT	CSMT	CSNCCR	CSNR	CTOC	CTODL	CQA	SOQA	ST2SQL
EmbeddingGemma 2	271M	768	**78.7**	**89.2**	**86.9**	**80.5**	85.6	82.2	**93.8**	90.9	84.6	35.6	**53.3**	89.1	72.5
		512	77.2	**89.2**	86.6	79.9	85.3	82.0	93.7	90.7	82.9	35.7	52.7	87.0	61.1
		256	76.2	88.3	85.5	77.9	84.1	80.5	92.9	90.2	82.8	35.8	50.5	84.9	60.9
		128	71.4	82.5	80.8	70.0	78.5	74.8	89.5	88.1	77.4	34.6	44.1	77.0	59.7
pplx-embed-v1-0.6b	596M	1024	75.8	78.7	82.4	76.4	83.0	84.7	88.1	89.9	85.0	35.2	43.3	90.7	73.0
UltraBERT-Embed-150M	151M	768	73.3	68.5	82.2	62.2	84.4	86.4	93.7	90.0	**88.1**	37.2	36.3	**92.4**	58.0
Giga-Embeddings-instruct	484M	1024	72.9	81.6	71.8	61.8	**91.2**	**89.6**	73.9	87.3	86.5	**37.6**	37.9	92.3	62.9
Qwen3-Embedding-0.6B	596M	1024	75.4	75.3	84.7	64.4	90.8	86.4	91.7	**91.0**	86.0	31.4	36.5	90.0	**76.7**
embeddinggemma-300m	308M	768	68.8	84.4	75.5	62.1	51.4	80.3	73.7	90.1	85.5	33.5	43.6	86.5	58.4
granite-embed-311m-multi	312M	768	63.8	60.6	77.6	56.2	57.0	77.3	55.5	86.9	83.2	34.4	35.4	86.3	55.7
granite-embed-english-r2	149M	768	57.2	14.0	64.7	59.3	52.5	77.2	47.7	80.8	77.1	35.0	37.0	91.8	49.6
granite-embed-97m-multi	97M	384	60.4	50.2	75.4	53.3	55.2	75.2	51.9	85.3	76.5	34.0	34.0	82.0	52.3
granite-embed-small-eng	48M	384	55.8	13.5	60.5	57.2	52.2	76.9	48.4	78.3	77.6	33.6	35.6	90.0	46.3
granite-embed-125m-eng	125M	768	53.3	11.8	55.1	58.8	42.1	75.3	47.6	76.9	66.7	29.6	36.6	89.8	48.7
b1ade-embed	335M	1024	53.6	6.7	79.8	55.8	37.8	72.5	51.8	86.7	57.0	22.3	35.3	83.5	54.0

Table 7 | Performance of top-ranking models under 1B parameters on MTEB(Code).

				Image					**Video**					**VisDoc**				
Model Name	Params	Dim.	**Mean (Task)**	CLS	QA	RET	GD	**All**	CLS	QA	RET	MRET	**All**	VDRv1	VDRv2	VR	OOD	**All**
EmbeddingGemma 2	439M	768	59.0	54.4	41.9	**65.8**	77.4	57.3	**56.2**	48.6	**47.2**	**50.6**	**50.7**	70.6	**58.3**	72.7	63.3	67.8
		512	58.4	54.2	41.4	65.4	77.1	56.9	56.0	48.5	46.5	50.3	50.3	69.2	56.3	72.0	62.2	66.6
		256	56.2	53.2	39.5	63.6	76.5	55.5	54.6	48.4	44.9	49.2	49.3	64.9	53.2	67.2	59.5	62.6
		128	45.6	39.9	29.6	54.7	72.8	45.6	40.8	44.4	34.9	43.6	40.6	51.9	39.2	52.6	48.8	49.5
Eddy-vl_embed_1.9B_v1	1.93B	2048	**63.2**	54.9	**61.7**	64.8	**84.7**	**63.4**	51.5	**57.9**	41.9	48.0	50.0	**78.8**	55.7	**79.1**	**64.9**	**72.7**
interestFM-UIR-CAFe	894M	896	49.3	**56.4**	45.3	57.5	72.0	55.4	33.9	41.7	29.7	39.7	35.9	56.9	32.6	68.6	23.8	50.3
jina-embed-v5-omni-nano	986M	768	43.6	34.6	16.0	37.4	45.5	31.6	28.1	41.5	18.0	38.3	30.7	77.6	53.9	78.6	62.2	71.3

Table 8 | Performance of top-ranking models under 2B parameters on MMEB-v2.

4.3. Vision

MMEB-v2 spans 78 tasks across image, video, and visual document modalities, the latter of which includes PDFs, slides, and charts. These include retrieval, classification, question answering, visual grounding, and moment retrieval. MIEB(lite) complements this with linear-probe classification, clustering, multiple choice, compositionality, and visual semantic text similarity. They both implicitly evaluate countless visual skills like object detection, counting, optical character recognition, and temporal reasoning. They also cover various permutations of cross-modal and mixed-modal situations.

EmbeddingGemma 2 is the smallest model on the MMEB-v2 leaderboard (Table 8). It excels especially at classification and retrieval, but provides strong scores across the board—its overall score is exceeded only by models around 2B parameters. It also outperforms various recent models including vision-language specialists with more than 8B parameters. On MIEB(lite), only two models beat EmbeddingGemma 2, and they have well over 4B parameters. Compared to top models under 1B parameters (Table 9), EmbeddingGemma 2 sets the state of the art on both aggregate metrics by a wide margin (+7.9 Task, +12.2 Type). It also offers remarkable improvements in nearly all task types, notably multilingual visual STS (+22.8) and multilingual retrieval (+12.5). Even at 128 dimensions, the model outperforms every other sub-1B model at full dimensionality.

4.4. Audio

MAEB includes 30 tasks encompassing five acoustic domains: speech, music, environmental, bioacoustics, and emotion. These are spread across unimodal and cross-modal tasks within retrieval, reranking, clustering, and multiple types of classification. MSEB(Retrieval) contributes a focus on multilingual speech understanding. It evaluates the model's ability to retrieve a text passage from a spoken query, both in the same language and across languages, with a total of 26 languages and regional locales.

Model Name	Params	Dim.	**Mean (Task)**	**Mean (Type)**	Multi. Retr.	A2A Retr.	Comp.	Doc. Und.	Image Class.	Image Clus.	Vision QA	VSTS (eng)	VSTS (multi)	0-shot Class.
EmbeddingGemma 2	439M	768	**60.2**	**65.7**	**65.4**	**38.3**	**45.7**	67.3	74.3	**81.3**	**59.1**	81.5	**76.8**	67.0
		512	59.9	65.3	64.8	38.1	**45.7**	66.5	74.2	80.0	58.8	81.6	**76.8**	66.9
		256	58.6	64.2	62.9	37.1	44.5	64.0	73.7	78.9	56.4	**81.8**	**76.8**	65.5
		128	54.2	60.0	50.7	33.0	40.1	56.6	71.5	79.3	56.7	80.8	74.8	56.4
siglip-so400m-patch14-384	878M	1152	52.3	53.5	35.4	32.4	32.8	46.9	**78.8**	75.9	48.0	69.6	41.4	**73.8**
siglip-large-patch16-384	652M	1024	51.4	53.3	43.5	31.9	32.1	44.9	77.7	75.2	46.8	69.6	39.8	71.3
siglip-base-patch16-512	204M	768	49.6	50.9	34.8	30.8	29.7	42.6	74.6	69.7	55.5	67.1	38.1	66.3
CLIP-ViT-L-14-DataComp.XL	428M	768	49.0	50.4	22.6	31.0	31.7	30.8	75.3	80.4	54.9	72.5	35.8	69.4
CLIP-ViT-H-14-laion2B	986M	1024	49.3	50.0	23.9	32.8	34.8	33.7	76.8	79.3	46.8	68.3	33.9	69.4
siglip-base-patch16-256-multi	371M	768	46.6	49.7	52.9	28.2	30.7	22.9	73.4	68.2	53.3	63.7	40.3	63.2
jina-embed-v5-omni-nano	986M	768	44.2	49.4	38.7	23.6	29.6	**74.2**	63.7	60.5	50.0	77.9	54.0	21.8
CLIP-ViT-L-14-laion2B	428M	768	47.4	47.6	21.4	31.1	33.6	28.7	75.9	76.4	46.9	68.7	26.0	67.8
clip-vit-large-patch14	428M	768	46.4	47.4	19.8	26.7	39.4	29.4	74.5	71.3	44.8	69.4	35.1	63.8
siglip-base-patch16-224	203M	768	46.1	47.4	33.6	29.3	32.5	20.9	73.5	68.4	53.0	64.2	33.5	65.0
CLIP-ViT-B-16-DataComp.XL	150M	512	45.5	46.6	19.9	28.3	31.4	22.7	73.2	73.6	56.9	69.6	28.5	61.9

Table 9 | Performance of top-ranking models under 1B parameters on MIEB(lite).

Model Name	Params	Dim.	**Mean (Task)**	**Mean (Type)**	A2A Retr.	Audio Class.	Audio Clus.	Multi. Class.	Pair Class.	Audio Rerank.	0-shot Class.
EmbeddingGemma 2	577M	768	49.9	48.7	**53.6**	**52.9**	4.1	38.7	**67.4**	68.5	55.4
		512	49.8	48.6	53.4	52.7	4.3	38.8	**67.4**	68.8	55.0
		256	49.1	48.2	52.1	52.2	4.2	38.4	67.2	68.7	54.8
		128	47.3	46.3	49.0	50.4	4.5	30.8	66.9	67.1	55.1
larger_clap_general	194M	512	35.0	41.1	18.9	47.9	6.6	80.6	51.9	66.8	14.9
jina-embed-v5-omni-nano	986M	768	**50.1**	**55.3**	50.0	51.3	6.0	70.8	65.6	83.2	**60.3**
larger_clap_music_and_speech	194M	512	34.7	40.7	18.6	47.4	7.7	80.6	52.1	65.6	13.2
clap-htsat-unfused	153M	512	33.0	40.5	18.1	41.9	12.5	80.8	52.6	66.5	11.3
msclap-2023	160M	1024	34.0	42.5	18.7	42.2	15.2	79.7	53.6	75.4	12.6
clap-htsat-fused	154M	512	33.5	41.3	17.6	41.2	**22.7**	**81.1**	52.0	61.3	13.2
MuQ-MuLan-large	630M	512	28.2	34.0	11.7	38.8	4.3	33.1	51.9	**85.4**	12.6
wav2clip	163M	512	26.6	31.1	9.5	37.4	6.0	31.8	53.6	68.9	10.8
msclap-2022	196M	1024	32.2	39.3	21.7	36.1	19.9	70.9	51.7	62.9	12.1
speecht5_multimodal	298M	768	26.4	30.0	7.9	38.5	1.1	32.5	57.9	56.5	15.9
larger_clap_music	194M	512	20.7	22.8	9.1	26.7	5.3	8.1	50.3	48.6	11.5

Table 10 | Performance of top-ranking models under 1B parameters on MAEB.

On MAEB, EmbeddingGemma 2 again sets a new frontier for efficiency. Notably, among sub-1B models, it achieves state-of-the-art scores on any-to-any retrieval and audio classification (Table 10), as well as on about a third of all individual tasks. It is also state-of-the-art among models of all sizes for classification of audio pairs and a sixth of all individual tasks. Its mean task score is only about 0.2 points away from the best model under 1B parameters, and about 3.5 points away from the highest-scoring model overall. Even its 128-dimensional embeddings beat every other sub-1B model.

On average, the model surpasses various audio-text specialists, but there is an interesting specialization divide between them. While EmbeddingGemma 2 does beat specialists like CLAP at non-speech tasks like clustering vehicle sounds (`VehicleSoundClustering`), these models still outperform in tasks like audio captioning (`ClothoT2ARetrieval`). In speech tasks, EmbeddingGemma 2 sets the state of the art even in tasks with multiple languages (`FleursT2ARetrieval`) and singing (`JamAltLyricA2TRetrieval`). But it sometimes struggles at zero-shot tasks with a paralinguistic focus, despite understanding the acoustics deep down. For example, it struggles tremendously at clustering speakers by gender (`VoxPopuliGenderClustering`: 0.40), but recovers quickly when doing the same task with a linear probe (`IEMOCAPGender`: 66.40). This indicates the model's speech representations are dominated by semantics, desirably avoiding biases such as speaker demographic while retaining paralinguistic and acoustic attributes for extraction when needed.

Model Name	Audio Mode	Parameters	Dim.	**Overall**	In-Lang	Cross-Lang
EmbeddingGemma 2	Native Audio	577M	768	**69.54**	**71.63**	**67.68**
			512	69.18	71.20	67.38
			256	66.76	68.86	64.89
			128	56.71	60.30	53.49
EmbeddingGemma 2	ASR	271M	768	57.59	62.14	53.52
EmbeddingGemma 1	ASR	308M	768	50.95	61.10	41.87
Gemini Embedding 2	Native Audio	–	3072	**73.99**	**75.58**	**72.56**
Gemini Embedding 2	ASR	–	3072	70.40	73.58	67.55

Table 11 | Results with ASR and native audio on MSEB(Retrieval) (MRR@10).

The model's excellence on spoken language is further underscored on MSEB. Even when using ASR and text embeddings, EmbeddingGemma 2 greatly outperforms EmbeddingGemma 1, indicating the strength persists throughout modalities (Table 11). Encoding audio natively yields another even larger improvement. Instead of suffering from transcription errors from accents, homophones, or artifacts, native audio inputs preserve these signals and let the model take them into account. As a result, even the 256-dimensional embeddings greatly outperform the full-dimension ASR embeddings.

5. Future Work

Our work points to several concrete directions for future research. EmbeddingGemma 2 supports long sequences and variable vision token budgets, but further extending context length, for example via content-aware methods, could enable direct indexing of full-length media, improve performance on long-horizon tasks, and support novel agentic memory systems. Innovations such as encoder-free architectures could enable both quality improvements and further reductions in memory footprint and latency. Changes in training paradigms like the introduction of end-to-end RAG training present another opportunity to further optimize embedding models for downstream use. As interleaved all-to-all multimodal search becomes more prevalent, evaluation frameworks that represent novel use cases will become more important.

At the same time, EmbeddingGemma 2's utility extends far beyond single-vector retrieval, serving as a general-purpose multimodal encoder. Operating on its final embeddings or unpooled token representations, the model can be adapted into (or may even excel zero-shot as) a general decision model [61, 62, 63, 64], model router [65], reward model [66], reranker [67], speech-to-text model [68, 69], late interaction retriever [70, 71, 72], and much more.

6. Conclusion

EmbeddingGemma 2 builds on its predecessor with a smaller, stronger backbone and native multimodal capabilities. Our approach introduces improvements like simplified per-layer embeddings and a multi-stage curriculum with both modality and objective progression. This enables EmbeddingGemma 2 to set a new Pareto frontier across text, vision, and audio benchmarks, often surpassing specialist and general-purpose models several times larger. With native support for weight quantization, embedding truncation, and modular encoder loading, EmbeddingGemma 2 makes high-quality multimodal embeddings practical in resource-constrained and on-device settings. We release EmbeddingGemma 2 checkpoints to the community to accelerate research and development, for both off-the-shelf use and as a foundational multimodal encoder.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. doi: 10.48550/ARXIV.2103.00020. URL https://doi.org/10.48550/arXiv.2103.00020.

[2] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *CoRR*, abs/2102.05918, 2021. doi: 10.48550/ARXIV.2102.05918. URL https://doi.org/10.48550/arXiv.2102.05918.

[3] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. URL https://arxiv.org/abs/2502.14786.

[4] Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-V: Universal embeddings with multimodal large language models, 2024. URL https://arxiv.org/abs/2407.12580.

[5] Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks, 2025. URL https://arxiv.org/abs/2410.05160.

[6] Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. MM-Embed: universal multimodal retrieval with multimodal LLMs, 2025. URL https://arxiv.org/abs/2411.02571.

[7] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. GME: Improving universal multimodal retrieval by multimodal LLMs, 2025. URL https://arxiv.org/abs/2412.16855.

[8] Madhuri Shanbhogue, Zhe Li, Shanfeng Zhang, Gustavo Hernández Ábrego, Shih-Cheng Huang, Aashi Jain, Daniel Salz, Sonam Goenka, Chaitra Hegde, Ji Ma, Feiyang Chen, Jiaxing Wu, Tanmaya Dabral, Babak Samari, Kevin Poulet, Daniel Cer, Kaifeng Chen, Paul Suganthan, Hui Hui, Jovan Andonov, Philippe Schlattner, Jay Han, Iftekhar Naim, Wing Lowe, Vladimir Pchelin, Albert Yang, Yi-Ting Chen, Zhongli Ding, Grace Zhang, Georg Heigold, Yichang Chen, Antoine Reveillon, Brendan McCloskey, Wenlei Zhou, Dahun Kim, Rui Meng, Emma Wang, Jack Zheng, Halley Fede, Zhen Yang, Keegan Mosley, Brian Potetz, Sahil Dua, Henrique Schechter Vera, Shen Gao, Hesen Zhang, Andreas Hess, Hengxuan Ying, Alberto Montes, Karan Gill, Min Choi, Sebastian Russo, Anja Hauth, Jinhyuk Lee, Michael Boratko, Megan Barnes, Vikram Rao, Claudiu Musat, Cyril Allauzen, Ehsan Variani, Shankar Kumar, Tom Bagby, Junyi Jiao, Yang Gu, Tengxin Li, Ayush Agrawal, Roberto Santana, Dev Nath, Stephen Karukas, Shuoxuan Han, Lucia Loher, Alice Twu, Nidhi Vyas, Siddharth Bhai, Frank Palma Gomez, Wangyuan Zhang, Chaoren Liu, Jizheng Yang, Steve Qiu, Shijie Zhang, Sujay Kulkarni, Sascha Rothe, Sean Nakamoto, Raphael Hoffmann, Zach Gleicher, Yunhsuan Sung, Qin Yin, Tom Duerig, and Mojtaba Seyedhosseini. Gemini Embedding 2: A native multimodal embedding model from Gemini, 2026. URL https://arxiv.org/abs/2605.27295.

[9] Mengyao Xu, Wenfei Zhou, Yauhen Babakhin, Gabriel Moreira, Ronay Ak, Radek Osmulski, Bo Liu, Even Oldridge, and Benedikt Schifferer. Omni-Embed-Nemotron: A unified multimodal retrieval model for text, image, audio, and video, 2025. URL https://arxiv.org/abs/2510.03458.

[10] Chenghao Xiao, Hou Pong Chan, Hao Zhang, Weiwen Xu, Mahani Aljunied, and Yu Rong. Scaling language-centric omnimodal representation learning, 2025. URL https://arxiv.org/abs/2510.11693.

[11] Haonan Chen, Sicheng Gao, Radu Timofte, Tetsuya Sakai, and Zhicheng Dou. e5-omni: Explicit cross-modal alignment for omni-modal embeddings, 2026. URL https://arxiv.org/abs/2601.03666.

[12] Embedding Team. Ovis-Embedding: Pushing the frontiers of universal omni-modal embeddings, 2026. URL https://arxiv.org/abs/2609.25165.

[13] Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. doi: 10.48550/ARXIV.1706.03762. URL https://doi.org/10.48550/arXiv.1706.03762.

[15] Gemma Team. Gemma 3n. https://deepmind.google/models/gemma/gemma-3n/, 2025.

[16] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with Gaussian Error Linear Units. *CoRR*, abs/1606.08415, 2016. doi: 10.48550/ARXIV.1606.08415. URL https://doi.org/10.48550/arXiv.1606.08415.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

[18] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition, 2020. URL https://arxiv.org/abs/2005.08100.

[19] Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara Sainath, Pedro Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. Google USM: Scaling automatic speech recognition beyond 100 languages, 2023. URL https://arxiv.org/abs/2303.01037.

[20] Fedor Moiseev, Gustavo Hernandez Abrego, Peter Dornbach, Imed Zitouni, Enrique Alfonseca, and Zhe Dong. SamToNe: Improving contrastive loss for dual encoder retrieval models with same tower negatives, 2023. URL https://arxiv.org/abs/2306.02516.

[21] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *International Conference on Machine Learning*, 2008. doi: 10.1145/1390156.1390306. URL https://api.semanticscholar.org/CorpusID:207168334.

[22] Xu Zhang, Felix X. Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. *CoRR*, abs/1708.06320, 2017. URL http://arxiv.org/abs/1708.06320.

[23] Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, Weiyi Wang, Zhe Li, Gus Martins, Jinhyuk Lee, Mark Sherwood, Juyeong Ji, Renjie Wu, Jingxiao Zheng, Jyotinder Singh, Abheesht Sharma, Divyashree Sreepathihalli, Aashi Jain, Adham Elarabawy, AJ Co, Andreas Doumanoglou, Babak Samari, Ben Hora, Brian Potetz, Dahun Kim, Enrique Alfonseca, Fedor Moiseev, Feng Han, Frank Palma Gomez, Gustavo Hernández Ábrego, Hesen Zhang, Hui Hui, Jay Han, Karan Gill, Ke Chen, Koert Chen, Madhuri Shanbhogue, Michael Boratko, Paul Suganthan, Sai Meher Karthik Duddu, Sandeep Mariserla, Setareh Ariafar, Shanfeng Zhang, Shijie Zhang, Simon Baumgartner, Sonam Goenka, Steve Qiu, Tanmaya Dabral, Trevor Walker, Vikram Rao, Waleed Khawaja, Wenlei Zhou, Xiaoqi Ren, Ye Xia, Yichang Chen, Yi-Ting Chen, Zhe Dong, Zhongli Ding, Francesco Visin, Gaël Liu, Jiageng Zhang, Kathleen Kenealy, Michelle Casbon, Ravin Kumar, Thomas Mesnard, Zach Gleicher, Cormac Brick, Olivier Lacombe, Adam Roberts, Qin Yin, Yunhsuan Sung, Raphael Hoffmann, Tris Warkentin, Armand Joulin, Tom Duerig, and Mojtaba Seyedhosseini. EmbeddingGemma: Powerful and Lightweight Text Representations, 2025. URL https://arxiv.org/abs/2509.20354.

[24] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning, 2022. URL https://arxiv.org/abs/2203.02053.

[25] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models, 2020. URL https://arxiv.org/abs/2011.05864.

[26] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models, 2019. URL https://arxiv.org/abs/1907.12009.

[27] Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *CoRR*, abs/1909.00512, 2019. doi: 10.48550/ARXIV.1909.00512. URL https://doi.org/10.48550/arXiv.1909.00512.

[28] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *CoRR*, abs/2005.10242, 2020. doi: 10.48550/ARXIV.2005.10242. URL https://doi.org/10.48550/arXiv.2005.10242.

[29] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations, 2018. URL https://arxiv.org/abs/1702.01417.

[30] Seungyeon Kim, Ankit Singh Rawat, Manzil Zaheer, Sadeep Jayasumana, Veeranjaneyulu Sadhanala, Wittawat Jitkrittum, Aditya Krishna Menon, Rob Fergus, and Sanjiv Kumar. EmbedDistill: A geometric knowledge distillation for information retrieval, 2023. URL https://arxiv.org/abs/2301.12005.

[31] Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. NV-Retriever: Improving text embedding models with effective hard-negative mining, 2025. URL https://arxiv.org/abs/2407.15831.

[32] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: effective and efficient retrieval via lightweight late interaction, 2022. URL https://arxiv.org/abs/2112.01488.

[33] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30233–30249. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c32319f4868da7613d78af9993100e42-Paper-Conference.pdf.

[34] 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, 2024. URL https://arxiv.org/abs/2212.03533.

[35] 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.

[36] 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, 2025. URL https://arxiv.org/abs/2503.07891.

[37] 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.

[38] 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, 2025. URL https://arxiv.org/abs/2405.17428.

[39] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning, 2025. URL https://arxiv.org/abs/2402.09906.

[40] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022. URL https://arxiv.org/abs/2203.05482.

[41] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *CoRR*, abs/1803.05407, 2018. URL http://arxiv.org/abs/1803.05407.

[42] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models, 2022. URL https://arxiv.org/abs/2109.01903.

[43] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. doi: 10.1109/CVPR.2018.00286.

[44] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy, 2017. URL https://arxiv.org/abs/1711.05852.

[45] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: Investigating the design space of visually-conditioned language models, 2024. URL https://arxiv.org/abs/2402.07865.

[46] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL https://arxiv.org/abs/2304.08485.

[47] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model, 2023. URL https://arxiv.org/abs/2303.03378.

[48] Tingyu Song, Mingxin Li, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Zhijie Nie, Yilun Zhao, and Shu Wu. UEmbed: Unified sparse and dense multimodal embeddings, 2026. URL https://arxiv.org/abs/2608.02583.

[49] Florian Hönicke, Michael Günther, Andreas Koukounas, Mohammad Kalim Akram, Saba Sturua, and Han Xiao. jina-embeddings-v5-omni: Geometry-preserving embeddings via locked aligned towers, 2026. URL https://arxiv.org/abs/2605.08384.

[50] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs, 2024. URL https://arxiv.org/abs/2406.16860.

[51] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Ankur Jain, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Guoli Yin, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. MM1: Methods, analysis & insights from multimodal LLM pre-training, 2024. URL https://arxiv.org/abs/2403.09611.

[52] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL https://arxiv.org/abs/2301.12597.

[53] Jaeseok Byun, Gukyeong Kwon, Han-Kai Hsu, Meher Gitika Karumuri, Zhikang Zhang, Hao Yang, and Davide Modolo. Multi-modal knowledge preserving adapter for embedding backward compatibility, 2026. URL https://arxiv.org/abs/2609.16875.

[54] Rishi Jha, Collin Zhang, Vitaly Shmatikov, and John X. Morris. Harnessing the universal geometry of embeddings, 2026. URL https://arxiv.org/abs/2505.12540.

[55] Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, et al. MMTEB: massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*, 2025. doi: 10.48550/arXiv.2502.13595. URL https://arxiv.org/abs/2502.13595.

[56] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022. doi: 10.48550/ARXIV.2210.07316. URL https://arxiv.org/abs/2210.07316.

[57] Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, Yingbo Zhou, Wenhu Chen, and Semih Yavuz. VLM2Vec-V2: Advancing multimodal embedding for videos, images, and visual documents, 2025. URL https://arxiv.org/abs/2507.04590.

[58] 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.

[59] Georg Heigold, Ehsan Variani, Tom Bagby, Cyril Allauzen, Ji Ma, Shankar Kumar, and Michael Riley. Massive sound embedding benchmark (MSEB), 2026. URL https://arxiv.org/abs/2602.07143.

[60] 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, 2026. URL https://arxiv.org/abs/2602.16008.

[61] Ying Fang, Zhifu Gao, Qing Gu, Jiajun He, Kangqi Jing, Xiangang Li, Yu Pu, and Wupeng Wang. Qwen-Audio-Jev: A unified interface for audio decisions, 2026. URL https://arxiv.org/abs/2610.21614.

[62] Yinheng Li and Justin Wagle. LLM-as-Jev: LLMs are already Jev-style decision models – when and how to fine-tune them, 2026. URL https://arxiv.org/abs/2610.02076.

[63] Jonathan Geuter, Andrew Cheng, and David Alvarez-Melis. Releasing the inner Jev: A simple recipe for open decision models, 2026. URL https://arxiv.org/abs/2610.17205.

[64] Diogo Almeida. Introducing system one models and jev. https://typesafe.ai/blog/introducing-system-one-models-and-jev, 2026.

[65] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs with preference data, 2025. URL https://arxiv.org/abs/2406.18665.

[66] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. RewardBench: Evaluating reward models for language modeling, 2024. URL https://arxiv.org/abs/2403.13787.

[67] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval, 2024. URL https://arxiv.org/abs/2407.19669.

[68] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020. URL https://arxiv.org/abs/2006.11477.

[69] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. W2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training, 2021. URL https://arxiv.org/abs/2108.06209.

[70] Jianrui Zhang, Hyun Jung Lee, Sukanta Ganguly, Tae-Eui Kam, Donghyun Kim, and Yong Jae Lee. Your embedding model is SMARTer than you think, 2026. URL https://arxiv.org/abs/2605.24938.

[71] Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT, 2020. URL https://arxiv.org/abs/2004.12832.

[72] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. ColPali: Efficient document retrieval with vision language models, 2025. URL https://arxiv.org/abs/2407.01449.

A. Task Instructions

Use Case	Query Task Instruction	Target Task Instruction
Web / Document Search	`task: search result | query: {text}`	`title: {title} | text: {text}`
Question Answering	`task: question answering | query: {text}`	`title: {title} | text: {text}`
Fact Checking	`task: fact checking | query: {text}`	`title: {title} | text: {text}`
Code Search	`task: code retrieval | query: {text}`	`title: {title} | text: {text}`
Classification	`task: classification | query: {text}`	*Same as query*
Clustering	`task: clustering | query: {text}`	*Same as query*
Semantic Similarity	`task: sentence similarity | query: {text}`	*Same as query*

Table 12 | Task instruction prefixes used for EmbeddingGemma 2. Asymmetric tasks (top) format queries and targets differently, using `title: none` when no document title is available. Symmetric tasks (bottom) apply the same prefix to all inputs.

B. Multilingual and English Leaderboards

Model Name	Parameters	Dim.	**Mean (Task)**	**Mean (Type)**	Bitext Mining	Class.	Clus.	Inst. Retrieval	Multi. Class.	Pair Class.	Rerank.	Retrieval	STS
EmbeddingGemma 2	271M	768	61.4	53.8	65.7	63.0	50.4	2.9	25.2	78.8	62.9	60.8	74.7
		512	61.2	53.7	65.4	62.7	51.0	2.5	25.1	78.8	62.8	60.2	74.4
		256	60.4	53.0	63.6	61.9	51.1	2.0	24.7	78.7	62.0	58.7	74.1
		128	57.9	50.7	59.4	59.7	49.0	0.8	23.3	78.5	59.9	52.3	73.6
harrier-oss-v1-0.6b	596M	1024	**69.0**	**59.0**	**82.8**	**73.9**	54.0	0.8	26.4	**82.1**	63.2	**70.8**	77.1
harrier-oss-v1-270m	268M	640	66.5	56.9	81.5	70.8	52.5	-0.5	24.0	80.1	61.9	66.4	75.4
Qwen3-Embedding-0.6B	596M	1024	64.3	56.0	72.2	66.8	52.3	5.1	24.6	80.8	61.4	64.6	76.2
jina-embed-v5-text-nano	212M	768	65.5	57.7	67.7	69.2	52.7	0.0	**41.3**	81.9	64.6	63.3	**78.2**
multilingual-e5-large-instruct	560M	1024	63.2	55.1	80.1	64.9	50.7	-0.4	23.0	80.8	62.6	57.1	76.8
F2LLM-v2-0.6B	596M	1024	62.7	55.0	70.3	64.1	**56.6**	1.4	25.2	79.0	**65.1**	59.3	74.2
embeddinggemma-300m	308M	768	61.1	54.3	64.4	60.9	51.2	**5.6**	24.7	81.4	63.2	62.5	74.7
BidirLM-0.6B-Embedding	596M	1024	60.4	52.9	67.6	62.5	49.5	1.2	24.1	78.4	62.7	59.1	70.7
F2LLM-v2-330M	334M	896	60.8	53.3	69.8	61.7	54.7	-1.7	24.6	77.8	63.3	56.5	72.7
jina-embeddings-v3	572M	1024	58.4	50.7	65.3	58.8	45.6	-1.3	18.4	79.3	57.1	55.8	77.1
nomic-embed-text-v2-moe	475M	768	57.6	49.6	60.7	60.4	44.5	-3.9	19.6	76.3	60.3	57.2	71.0

Table 13 | Performance of top-ranking models under 1B parameters on MTEB(Multilingual, v2).

Model Name	Parameters	Dim.	**Mean (Task)**	**Mean (Type)**	Class.	Clus.	Pair Class.	Rerank.	Retrieval	STS	Summ.
EmbeddingGemma 2	271M	768	68.5	63.9	88.8	55.4	86.4	46.3	51.7	83.2	35.1
		512	68.4	63.9	88.7	55.3	86.4	46.3	51.6	83.3	35.6
		256	67.8	63.3	88.5	55.5	86.1	45.9	49.4	83.1	34.8
		128	65.7	61.5	87.6	54.3	85.3	45.7	43.7	82.5	31.4
jina-embed-v5-text-nano	212M	768	71.1	**65.2**	**90.1**	53.5	84.7	49.2	58.8	**88.3**	31.9
Qwen3-Embedding-0.6B	596M	1024	70.5	64.7	84.6	54.1	84.4	48.2	61.8	86.6	33.4
Yuan-embedding-2.0-en	596M	1024	**71.7**	**65.2**	84.4	51.7	82.0	**53.3**	**70.7**	84.9	29.7
embeddinggemma-300m	308M	768	69.7	65.1	87.6	**56.6**	87.3	47.4	55.7	83.6	**37.6**
Nano-Em1-0.6B-v2.1	596M	1024	67.9	64.1	87.6	55.4	**89.9**	49.5	46.6	85.4	34.2
Tarka-Embedding-350M-V1	354M	1024	69.3	63.3	88.4	55.7	84.0	47.8	55.1	84.6	27.4
Giga-Embeddings-instruct	484M	1024	69.5	64.4	90.0	55.2	84.8	47.9	56.3	82.3	34.1
GIST-large-Embedding-v0	335M	1024	66.3	62.0	78.9	48.8	86.7	48.8	54.5	84.4	31.5
UltraBERT-Embed-150M	151M	768	67.2	62.8	81.7	49.7	85.3	47.5	57.2	83.0	34.9
mxbai-embed-large-v1	335M	1024	66.3	62.0	79.1	47.5	87.2	48.1	55.4	84.4	32.6
UAE-Large-V1	335M	1024	66.4	61.8	79.1	47.9	87.2	48.4	55.9	84.4	30.1

Table 14 | Performance of top-ranking models under 1B parameters on MTEB(English, v2).

C. Full Results

Task Name	Performance
AILAStatutes	32.44
AfriSentiClassification	45.67
AlloProfClusteringS2S.v2	52.31
Alloprof Reranking	76.92
AmazonCounterfactualClassification	85.83
ArXivHierarchicalClusteringP2P	63.42
ArXivHierarchicalClusteringS2S	60.59
ArguAna	59.88
ArmenianParaphrasePC	93.36
BUCC.v2	98.50
BelebeleRetrieval	71.91
BibleNLPBitextMining	12.27
BigPatentClustering.v2	33.31
BiorxivClusteringP2P.v2	48.83
BornholmBitextMining	49.41
BrazilianToxicTweetsClassification	28.35
BulgarianStoreReviewSentimentClassfication	68.68
CEDRClassification	50.38
CLSClusteringP2P.v2	39.91
CSFDSKMovieReviewSentimentClassification	35.08
CTKFactsNLI	78.55
CataloniaTweetClassification	51.66
Core17InstructionRetrieval	4.34
CovidRetrieval	77.22
CyrillicTurkicLangClassification	68.49
CzechProductReviewSentimentClassification	58.35
DBpediaClassification	91.83
DalajClassification	50.30
DiaBlaBitextMining	81.71
EstonianValenceClassification	38.59
FaroeseSTS	71.58
FilipinoShopeeReviewsClassification	37.17
FinParaSTS	24.57
FinancialPhrasebankClassification	83.89
FloresBitextMining	54.89
GermanSTSBenchmark	83.14
GreekLegalCodeClassification	35.42
GujaratiNewsClassification	87.44
HALClusteringS2S.v2	28.35
HagridRetrieval	99.10
IN22GenBitextMining	73.59
IndicCrosslingualSTS	42.92
IndicGenBenchFloresBitextMining	86.74
IndicLangClassification	85.49
IndonesianIdClickbaitClassification	64.53
IsiZuluNewsClassification	24.76
ItaCaseholdClassification	69.09
JSICK	84.23
KorHateSpeechMLClassification	11.58
KorSarcasmClassification	59.80
KurdishSentimentClassification	64.04
LEMBPasskeyRetrieval	54.25
LegalBenchCorporateLobbying	93.02
MIRACLRetrievalHardNegatives	66.11
MLQARetrieval	76.82
MacedonianTweetSentimentClassification	57.97
MalteseNewsClassification	30.94
MasakhaNEWSClassification	75.70
MasakhaNEWSClusteringS2S	46.02
MassiveIntentClassification	61.52
MedrxivClusteringP2P.v2	43.13
MultiEURLEXMultilabelClassification	4.78
MultiHateClassification	57.64
NTREXBitextMining	73.97
NepaliNewsClassification	95.46
News21InstructionRetrieval	5.10

Task Name	Performance
NollySentiBitextMining	37.21
NordicLangClassification	65.88
NorwegianCourtsBitextMining	92.98
NusaParagraphEmotionClassification	45.05
NusaTranslationBitextMining	71.37
NusaX-senti	68.86
NusaXBitextMining	67.19
OdiaNewsClassification	72.81
OpusparcusPC	92.38
PAC	67.16
PawsXPairClassification	56.38
PlscClusteringP2P.v2	72.46
PoemSentimentClassification	57.04
PolEmo2.0-OUT	58.10
PpcPC	88.46
PunjabiNewsClassification	82.87
RTE3	86.63
Robust04InstructionRetrieval	-0.75
RomaniBibleClustering	40.46
RuBQReranking	71.05
SCIDOCS	18.14
SIB200ClusteringS2S	26.40
SICK-R	82.19
STS12	77.82
STS13	86.48
STS14	82.75
STS15	88.45
STS17	85.79
STS22.v2	69.91
STSB	81.34
STSBenchmark	87.48
STSES	81.24
ScalaClassification	50.12
SemRel24STS	65.43
SentimentAnalysisHindi	68.23
SinhalaNewsClassification	69.54
SiswatiNewsClassification	55.88
SlovakMovieReviewSentimentClassification	78.44
SpartQA	14.56
SprintDuplicateQuestions	96.50
StackExchangeClustering.v2	91.55
StackOverflowQA	89.08
StatcanDialogueDatasetRetrieval	45.01
SwahiliNewsClassification	61.29
SwednClusteringP2P	42.72
SwissJudgementClassification	57.68
T2Reranking	66.64
TERRa	57.02
TRECCOVID	79.33
Tatoeba	54.18
TempReasonL1	1.32
ToxicConversationsClassification	91.03
TswanaNewsClassification	32.61
TweetTopicSingleClassification	73.01
TwitterHjerneRetrieval	68.69
TwitterURLCorpus	86.83
VoyageMMarcoReranking	61.78
WebLINXCandidatesReranking	12.79
WikiCitiesClustering	89.59
WikiClusteringP2P.v2	27.05
WikipediaRerankingMultilingual	88.38
WikipediaRetrievalMultilingual	89.47
WinoGrande	57.93
XNLI	75.95
indonli	54.59

Table 15 | Results of EmbeddingGemma 2 on MTEB(Multilingual, v2).

Task Name	Performance
AmazonCounterfactualClassification	91.10
ArXivHierarchicalClusteringP2P	63.66
ArXivHierarchicalClusteringS2S	60.59
ArguAna	59.95
AskUbuntuDupQuestions	60.61
BIOSSES	85.33
Banking77Classification	92.27
BiorxivClusteringP2P.v2	48.85
CQADupstackGamingRetrieval	56.90
CQADupstackUnixRetrieval	42.16
ClimateFEVERHardNegatives	23.08
FEVERHardNegatives	81.65
FiQA2018	32.47
HotpotQAHardNegatives	78.25
ImdbClassification	91.03
MTOPDomainClassification	98.74
MassiveIntentClassification	88.41
MassiveScenarioClassification	91.66
MedrxivClusteringP2P.v2	42.86
MedrxivClusteringS2S.v2	39.54
MindSmallReranking	32.05
SCIDOCS	18.16
SICK-R	82.20
STS12	77.82
STS13	86.48
STS14	82.75
STS15	88.46
STS17	91.08
STS22.v2	67.42
STSBenchmark	87.48
SprintDuplicateQuestions	96.50
StackExchangeClustering.v2	91.79
StackExchangeClusteringP2P.v2	48.23
SummEvalSummarization.v2	35.11
TRECCOVID	79.33
Touche2020Retrieval.v3	44.70
ToxicConversationsClassification	91.03
TweetSentimentExtractionClassification	66.44
TwentyNewsgroupsClustering.v2	48.05
TwitterSemEval2015	75.83
TwitterURLCorpus	86.83

Task Name	Performance
AppsRetrieval	89.19
COIRCodeSearchNetRetrieval	86.89
CodeEditSearchRetrieval	80.52
CodeFeedbackMT	85.60
CodeFeedbackST	82.16
CodeSearchNetCCRetrieval	93.84
CodeSearchNetRetrieval	90.85
CodeTransOceanContest	84.55
CodeTransOceanDL	35.60
CosQA	53.34
StackOverflowQA	89.08
SyntheticText2SQL	72.47

Table 16 | Results of EmbeddingGemma 2 on MTEB(English, v2) (left) and MTEB(Code) (right).

Task Name	Performance
BeijingOpera	81.37
BirdCLEF	16.00
CREMADPairClassification	52.50
CREMA_D	19.67
CREMA_DClustering	0.57
ClothoT2ARetrieval	10.71
CommonLanguageAgeDetection	17.59
CommonVoiceMini21T2ARetrieval	93.26
FSD2019Kaggle	38.72
FleursT2ARetrieval	97.32
GTZANAudioReranking	68.48
GTZANGenre	58.00
GigaSpeechT2ARetrieval	82.03
IEMOCAPGender	66.40
JamAltArtistA2ARetrieval	55.55
JamAltLyricA2TRetrieval	64.01
MACST2ARetrieval	6.36
MInDS14	90.30
MridinghamTonic	32.89
NMSQAPairClassification	98.12
RavdessZeroshot	13.96
SIBFLEURS	54.53
SpeechCommandsZeroshotv0.02	96.86
SpokenSQuADT2ARetrieval	72.33
UrbanSound8KT2ARetrieval	0.47
VehicleSoundClustering	11.31
VoxCelebSA	45.17
VoxPopuliAccentPairClassification	51.71
VoxPopuliGenderClustering	0.40
VoxPopuliLanguageID	99.60

		Performance (MRR@10)	
Language		In-Lang	Cross-Lang
Arabic (Egypt)	(ar-EG)	75.51	75.45
Arabic (Gulf)	(ar-x-gulf)	77.02	77.61
Arabic (Levant)	(ar-x-levant)	78.60	78.47
Arabic (Maghreb)	(ar-x-maghrebi)	73.31	72.54
Bengali (Bangladesh)	(bn-BD)	79.19	79.06
Bengali (India)	(bn-IN)	80.01	80.17
English (Australia)	(en-AU)	74.26	-
English (UK)	(en-GB)	68.29	-
English (India)	(en-IN)	72.78	-
English (Philippines)	(en-PH)	68.94	-
English (US)	(en-US)	72.30	-
Finnish (Finland)	(fi-FI)	48.55	48.40
Gujarati (India)	(gu-IN)	-	62.58
Hindi (India)	(hi-IN)	-	65.27
Indonesian (Indonesia)	(id-ID)	75.24	-
Japanese (Japan)	(ja-JP)	-	67.09
Kannada (India)	(kn-IN)	-	61.26
Korean (South Korea)	(ko-KR)	62.45	73.09
Malayalam (India)	(ml-IN)	-	61.63
Marathi (India)	(mr-IN)	-	62.42
Russian (Russia)	(ru-RU)	65.24	80.79
Swahili	(sw)	65.35	-
Tamil (India)	(ta-IN)	-	54.11
Telugu (India)	(te-IN)	80.74	69.85
Urdu (India)	(ur-IN)	-	59.24
Urdu (Pakistan)	(ur-PK)	-	56.81

Table 17 | Results of EmbeddingGemma 2 on MAEB (left) and MSEB(Retrieval) (right).

Task Name	Performance
AROCocoOrder	57.00
AROFlickrOrder	56.12
AROVisualAttribution	77.97
AROVisualRelation	56.93
BLINKIT2IMultiChoice	75.87
CIFAR100ZeroShot	75.61
CIRRIT2IRetrieval	22.12
CUB200I2IRetrieval	78.84
CVBenchCount	59.77
CVBenchDepth	52.67
CVBenchDistance	53.67
CVBenchRelation	53.39
Country211	20.83
Country211ZeroShot	22.73
DTD	77.50
EuroSAT	85.90
FER2013ZeroShot	61.80
FGVCAircraftZeroShot	45.21
Fashion200kI2TRetrieval	15.32
Food101ZeroShot	81.58
GTSRB	82.86
HatefulMemesI2TRetrieval	78.21
ImageCoDe	14.16
ImageNetDog15Clustering	91.39
InfoSeekIT2TRetrieval	13.87
NIGHTSI2IRetrieval	26.16

Task Name	Performance
OVENIT2TRetrieval	16.00
OxfordPets	93.32
OxfordPetsZeroShot	90.81
PatchCamelyon	69.34
RESISC45	89.76
RP2kI2IRetrieval	73.74
STS13VisualSTS	77.31
STS15VisualSTS	85.67
SUN397	75.13
StanfordCarsZeroShot	91.17
TinyImageNetClustering	71.26
VQA2IT2TRetrieval	16.60
VidoreDocVQARetrieval	39.81
VidoreInfoVQARetrieval	77.34
VidoreShiftProjectRetrieval	65.31
VidoreSyntheticDocQAAIRetrieval	87.76
VidoreTabfquadRetrieval	91.33
VidoreTatdqaRetrieval	42.26
VisualNewsI2TRetrieval	16.94
VisualSTS-b-Multilingual	75.59
VisualSTS17Multilingual	77.95
WITT2IRetrieval	65.31
WebQAT2ITRetrieval	64.00
Winoground	12.00
XM3600T2IRetrieval	65.45

Table 18 | Results of EmbeddingGemma 2 on MIEB(lite).

Task Name	Performance
A-OKVQA	35.20
ActivityNetQA	52.10
Breakfast	35.57
CIRR	40.10
Charades-STA	25.58
ChartQA	18.90
Country211	18.50
DiDeMo	52.49
DocVQA	51.50
EDIS	80.70
EgoSchema	56.60
FashionIQ	27.90
GQA	61.40
HMDB51	55.30
HatefulMemes	51.40
ImageNet-1K	65.00
ImageNet-A	39.80
ImageNet-R	87.40
InfographicsVQA	22.70
K700	54.70
MMLongBench-doc	45.08
MMLongBench-page-fixed	46.34
MSCOCO	63.20
MSCOCO_i2t	72.80
MSCOCO_t2i	77.70
MSR-VTT	50.90
MSVD	67.46
MVBench	43.80
MomentSeeker	47.88
N24News	38.10
NExTQA	52.74
NIGHTS	66.70
OK-VQA	43.40
OVEN	68.00
ObjectNet	65.50
Place365	35.30
QVHighlight	78.21
RefCOCO	82.90
RefCOCO-Matching	72.40

Task Name	Performance
SUN397	67.50
ScienceQA	26.90
SmthSmthV2	59.80
TextVQA	74.10
UCF101	75.90
VATEX	48.06
VOC2007	75.20
ViDoRe_arxivqa	70.56
ViDoRe_biomedical_lectures_v2_multilingual	46.99
ViDoRe_docvqa	36.60
ViDoRe_economics_reports_v2_multilingual	53.81
ViDoRe_esg_reports_human_labeled_v2	78.22
ViDoRe_esg_reports_v2_multilingual	54.25
ViDoRe_infovqa	78.79
ViDoRe_shiftproject	63.54
ViDoRe_syntheticDocQA_artificial_intelligence	81.78
ViDoRe_syntheticDocQA_energy	78.67
ViDoRe_syntheticDocQA_government_reports	81.87
ViDoRe_syntheticDocQA_healthcare_industry	87.60
ViDoRe_tabfquad	91.62
ViDoRe_tatdqa	34.75
ViDoSeek-doc	79.50
ViDoSeek-page-fixed	82.29
Video-MME	37.89
VisDial	76.50
VisRAG_ArxivQA	68.14
VisRAG_ChartQA	72.33
VisRAG_InfoVQA	83.26
VisRAG_MP-DocVQA	67.67
VisRAG_PlotQA	55.52
VisRAG_SlideVQA	89.08
Visual7W	38.50
Visual7W-Pointing	91.00
VisualNews_i2t	60.00
VisualNews_t2i	58.50
VizWiz	46.40
WebQA	89.70
Wiki-SS-NQ	71.40
YouCook2	17.08

Table 19 | Results of EmbeddingGemma 2 on MMEB-v2.

Contributions and Acknowledgements

Core Contributors (*: equal contributions)
Henrique Schechter Vera*
Sahil Dua*
Jovan Andonov
Sara Smoot
Sebastian Russo
Ivan Llanos
Max Matveev
Jae Hyeon Yoo
Kat Black

Contributors
Shih-Cheng Huang
Josiah Aklilu
Maarten Grootendorst
Omar Sanseviero
Chandramouli Amarnath
Ian Ballantyne
Zhe Li
Daniel Salz
Ibrahim Alabdulmohsin
Ilya Philippov
Kristen Quan
Glenn Cameron
Edouard Yvinec
Luciano Martins
Doug Reid
Sindhu Raghuram Panyam
Weiyi Wang
Hengxuan Ying
Madhuri Shanbhogue
Tanmaya Dabral
Shanfeng Zhang
Juyeong Ji
Renjie Wu
Joe Zou
Xu Chen
Yunan Zhang
Yu-Hui Chen
Matthew Soulanille
Vinod Mamillapalli
Yi-Chun Kuo
Wai Hon Law
Matt Kreileder
Alice Zheng
Suril Shah
Mogan Shieh
Pooja Consul

Gemini Embedding *(alphabetical order)*
Alberto Montes
Andreas Hess
Andres Sevilla
Babak Samari
Brendan McCloskey
Brian Potetz
Daniel Cer
Dilip Kumar Margam
Feodor Beloborodov
Georgios Kathareios
Gustavo Hernández Ábrego
Hesen Zhang
Ji Ma
Jianan Lu
Jiaxing Wu
Keegan Mosley
Paul Suganthan
Rui Meng
Shen Gao
Shengcao Cao
Wing Lowe
Yi-Ting Chen
Yichang Chen
Zhongli Ding

Leadership
Kendra Byrne
Cormac Brick
Olivier Lacombe
Qin Yin
Tris Warkentin
Armand Joulin
Mojtaba Seyedhosseini
Tom Duerig

Acknowledgements
Anand Rao, Raushan Turganbay, Anton Vlasjuk, Tom Aarsen, Srinivasan (Cheenu) Venkatachary, Tomáš Ižo, Raia Hadsell, Zoubin Ghahramani