jina-embeddings-v4: Universal Embeddings for Multimodal Multilingual Retrieval

Michael Günther*, **Saba Sturua***, **Mohammad Kalim Akram***, **Isabelle Mohr***, **Andrei Ungureanu***, **Bo Wang***, **Sedigheh Eslami**, **Scott Martens**, **Maximilian Werk**, **Nan Wang** and **Han Xiao**
Jina AI GmbH, Prinzessinnenstraße 19, 10969, Berlin, Germany
research@jina.ai

Abstract

We introduce jina-embeddings-v4, a 3.8 billion parameter embedding model that unifies text and image representations, with a novel architecture supporting both single-vector and multi-vector embeddings. It achieves high performance on both single-modal and cross-modal retrieval tasks, and is particularly strong in processing visually rich content such as tables, charts, diagrams, and mixed-media formats that incorporate both image and textual information. We also introduce JVDR, a novel benchmark for visually rich document retrieval that includes more diverse materials and query types than previous efforts. We use JVDR to show that jina-embeddings-v4 greatly improves on state-of-the-art performance for these kinds of tasks.

1 Introduction

We present jina-embeddings-v4, a multimodal embedding model capable of processing text and image data to produce single- and multi-vector embeddings, with modular LoRA adapters (Hu et al., 2022) for information retrieval and semantic text similarity. An adapter is also provided for programming language embeddings, technical question-answering, and natural language code retrieval.

This model supports dual-mode output, producing both single-vector outputs suitable for conventional embeddings-based applications and multi-vector embeddings for "late interaction" applications along the lines of ColBERT (Khattab and Zaharia, 2020) and ColPali (Faysse et al., 2025). This single-model approach entails significant savings in practical use cases when compared to deploying multiple AI models for different tasks and modalities.

A major contribution of this model is introducing new functionality for processing "visually rich" documents: mixed textual and visual media like tables, charts, diagrams, screenshots, web page captures, and similar images. (Ding et al., 2024) We have devised a new diversified benchmark, JVDR, for measuring performance on visually rich materials and show that jina-embeddings-v4 far outpaces comparable models on this type of media.

2 Related Work

Late interaction models generally have higher precision than traditional embedding models. (Khattab and Zaharia, 2020; Faysse et al., 2025) These models produce multi-vector outputs that consist of sequences of context-sensitive token embeddings. Similarity is calculated using a form of chamfer distance adapted to the task: Given two sequences of token embeddings, a query and a document, sum the maximum cosine similarity values of each query token embedding to any of the document token embeddings.

Faysse et al. (2025) train a late-interaction embedding model to search document screenshots using text queries, performing significantly better than traditional approaches involving OCR and CLIP-style models trained on image captions. To show this, they introduce the *ViDoRe* (Vision Document Retrieval) benchmark. However, this benchmark is limited to question-answering tasks in English and French involving only charts, tables, and pages from PDF documents. Xiao et al. (2025) extend this benchmark to create MIEB (Massive Image Embedding Benchmark) by rendering the texts from existing semantic textual similarity tasks as images.

The principal purpose of multimodal embedding models is to project objects from multiple modalities into the same semantic embedding space. Bimodal image-text models derived from OpenAI's CLIP architecture (Radford et al., 2021) consist of one model for each modality, typically trained with bimodal contrastive pairs to produce embeddings in a common semantic space. The *Vision-Language*

*These authors contributed equally to this work

Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025), pages 531–550
November 8-9, 2025 ©2025 Association for Computational Linguistics

Model (VLM) is an alternate architecture with a single processing path for both images and texts, significantly improving performance on bimodal text-image tasks. (Chen et al.; Bai et al., 2025)

Previous work has shed light on the so-called *modality gap* in this kind of model. (Liang et al., 2022; Schrodi et al., 2025; Eslami and de Melo, 2025) Good semantic matches across modalities tend to lie considerably further apart in the embedding space than comparable or even worse matches of the same modality, i.e., texts in CLIP-style models are more similar to semantically unrelated texts than to semantically similar images. Bai et al. (2025) demonstrate that VLMs have less of a modality gap than CLIP-style dual encoder architectures.

3 Model Architecture

The architecture of `jina-embeddings-v4`, schematized in Figure 1, is a VLM built on a `Qwen2.5-VL-3B-Instruct`[1] backbone. Text and image inputs are processed through a shared pathway: Images are first converted to token sequences via a vision encoder, then both modalities are jointly processed by the language model decoder with contextual attention layers.

As shown in Figure 1, this architecture supports single- and multi-vector output. Additionally, three task-specific LoRA adapters, each with 60M parameters, provide specialized task optimization without modifying the frozen backbone weights.

The core specifications of `jina-embeddings-v4` are summarized in Table 1.

`jina-embeddings-v4` differs from CLIP-style dual-encoder models in offering a single processing path for both text and image input. For text input, it behaves like other Transformer-based embedding models: The text is tokenized, each token is replaced with a vector representation from a lookup table, and then these vectors are stacked and become the input vector to a Transformer-based language model.

For images, a Transformer-based image model acts as a preprocessor to the language model: The image is divided into patches and the image model processes it as if each patch were a token given to a language model. The output is a multi-vector embedding which becomes the input to the language model, as if it were a stacked set of tokenized text vectors.

Users can choose between traditional single (dense) vector embeddings and ColBERT-style multi-vector embeddings. Single-vector embeddings are the result of mean-pooling the final layer of the base model to 2048 dimensions. `jina-embeddings-v4` has been trained with Matryoshka Representation Learning (Kusupati et al., 2022), so its single-vector embeddings can be truncated to as few as 128 dimensions with minimal loss of precision. An additional layer projects the output of the base model into multi-vector embeddings comparable to ColBERT (Khattab and Zaharia, 2020) and ColPali (Faysse et al., 2025) outputs. Single-vector embeddings offer fast, memory-efficient retrieval ideal for large-scale or first-stage search, while multi-vector late-interaction approaches are more costly but achieve higher accuracy by capturing fine-grained interactions, as shown in the evaluation results in Table 2. Multi-vector embeddings are best used to re-rank first-stage retrieval results on a smaller set of candidates or for technically challenging matching scenarios where single-vector approaches perform poorly, such as scanned technical documents.

Feature	Value
Parameters	3.8 billion (3.8×10^9) plus 60M per LoRA
Text input	Up to 32,768 tokens
Image input	All images resized to 20 megapixels
Single-vector embedding	2048 dimensions, truncatable down to 128
Multi-vector embedding	128 dimensions per token

Table 1: Basic specifications of `jina-embeddings-v4`

We have implemented three task-specific LoRA adapters for different information retrieval use cases described in Section 4.2. Each LoRA adapter has only 60M parameters, so keeping all three in memory adds less than 2% to the memory footprint of `jina-embeddings-v4`. See Section 6 for performance information about these adapters. We employ PEFT (Mangrulkar et al., 2022) to support LoRA and dynamically switch between adapters based on the intended task for each batch, without significant runtime overhead. We used a standard LoRA configuration with rank 32 and a scaling factor of 1, parameterizing all linear layers in the backbone LLM.

[1] https://huggingface.co/Qwen/Qwen2.5-VL-3B-Instruct

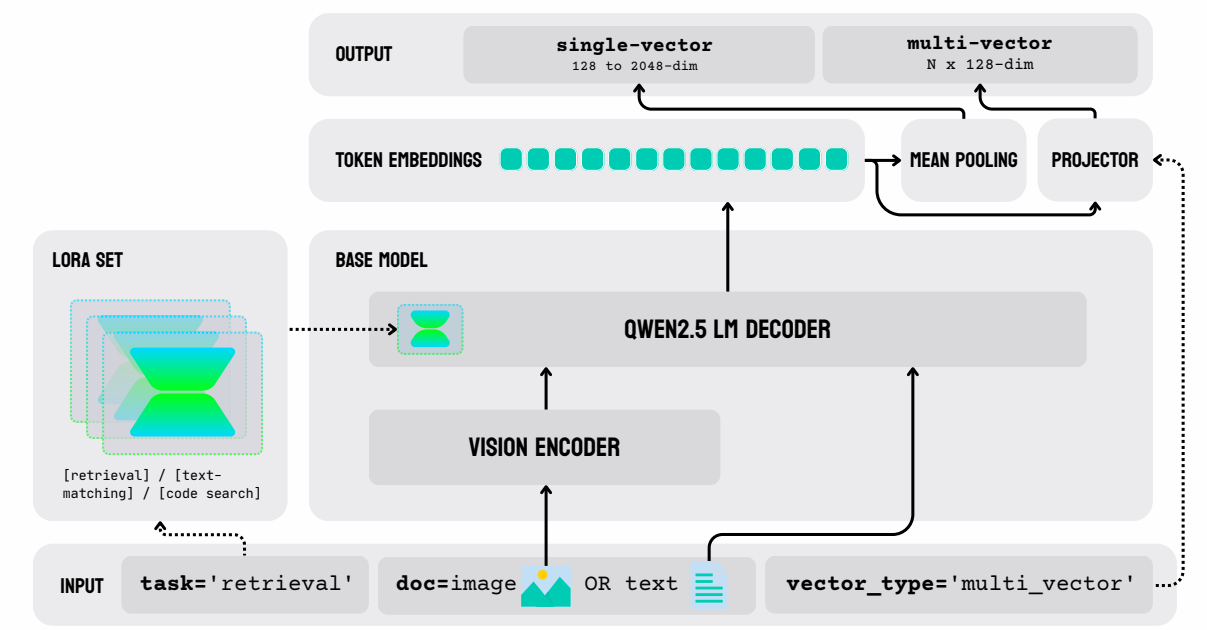

Figure 1: Architecture of `jina-embeddings-v4`.

4 Training Method

Before training, model weights are initialized to match `Qwen/Qwen2.5-VL-3B-Instruct`. The multi-vector projection layer and LoRA adapters are randomly initialized. Only the LoRA adapters are trained, the base model and projection layer remain as initialized.

In all phases of training, we apply Matryoshka loss (Kusupati et al., 2022) to our base loss function so that single-vector embeddings from `jina-embeddings-v4` are truncatable.

4.1 LoRA pre-training

We pre-train a single LoRA adapter using pair data and the contrastive InfoNCE (van den Oord et al., 2018) loss function. There is no task-specific training in the pre-training phase.

The training data consists of text-text and text-image pairs from more than 300 sources. Text-text pairs are selected and filtered as described in Sturua et al. (2024). Text-image pairs have been curated from a variety of sources following a more eclectic strategy than previous work on training text-image embedding models. We have also created images from website screenshots, rendered Markdown files, charts, tables, and other kinds of materials "found in the wild." Queries primarily consist of questions, keywords, key phrases, long descriptions, and statements of fact.

In each training step, we sample two different batches of training data:

- A batch $\mathcal{B}_{text}$ of text pairs.
- A batch $\mathcal{B}_{multi}$ of multimodal pairs containing a text and a related image.

We generate normalized single-vector and multi-vector embeddings for all texts and images in the selected pairs. We then construct a matrix of similarity values $\mathbf{S}_{\text{dense}}(\mathcal{B})$ by calculating the cosine similarity of all combinations of single-vector embeddings $\boldsymbol{q}_i$ and $\boldsymbol{p}_j$ in $\mathcal{B}$. We construct an analogous matrix $\mathbf{S}_{\text{chamf}}$ for each $\mathcal{B}$ for the multi-vector embeddings using a normalized version of the chamfer distance metric described by Khattab and Zaharia (2020) for the ColBERT late interaction model. Our choice of loss function requires a normalized score, so we divide the chamfer distance by the number of tokens in the query.

The result is four matrices of normalized similarity scores for each batch:

- Cosine similarity of single-vector embeddings for text-text pairs.
- Chamfer similarity of multi-vector embeddings for text-text pairs.
- Cosine similarity of single-vector embeddings for text-image pairs.
- Chamfer similarity of multi-vector embeddings for text-image pairs.

Then, we apply the contrastive InfoNCE loss function $\mathcal{L}_{\text{NCE}}$ (van den Oord et al., 2018) to each of the four matrices to calculate the training loss.

Following Hinton et al. (2014), we compensate for differences in error distributions between the single-vector and multi-vector similarity scores by adding the Kullback–Leibler divergence (D_{KL}) of the two sets of softmax-normalized similarity scores. This enables us to train for the single-vector and multi-vector outputs simultaneously, even though the multi-vector/late interaction scores have much less error.

Given $\mathbf{S}_{\text{dense}}(\mathcal{B})$ as the softmax of a matrix of single-vector cosine similarity scores for batch $\mathcal{B}$, and $\mathbf{S}_{\text{chamf}}(\mathcal{B})$ as the softmax of a matrix of multi-vector chamfer similarity scores for batch $\mathcal{B}$, define the added term $\mathcal{L}_D(B)$

$$\mathcal{L}_D(B) := D_{\text{KL}}(\mathbf{S}_{\text{dense}}(\mathcal{B}) \| \mathbf{S}_{\text{chamf}}(\mathcal{B}))$$

The resulting joint loss function, which we use in training, is defined as:

$$\begin{aligned}
&\mathcal{L}_{\text{joint}}(\mathcal{B}_{\text{text}}, \mathcal{B}_{\text{multi}}) := \\
&\quad w_1 \mathcal{L}_{\text{NCE}}(\mathbf{S}_{\text{dense}}(\mathcal{B}_{\text{text}}),) \\
&\quad + w_2 \mathcal{L}_{\text{NCE}}(\mathbf{S}_{\text{chamf}}(\mathcal{B}_{\text{text}})) + w_3 \mathcal{L}_D(\mathcal{B}_{\text{text}}) \\
&\quad + w_4 \mathcal{L}_{\text{NCE}}(\mathbf{S}_{\text{dense}}(\mathcal{B}_{\text{multi}})) \\
&\quad + w_5 \mathcal{L}_{\text{NCE}}(\mathbf{S}_{\text{chamf}}(\mathcal{B}_{\text{multi}})) + w_6 \mathcal{L}_D(\mathcal{B}_{\text{multi}})
\end{aligned}$$

The weights $w_1, \ldots, w_6$ are training hyperparameters.

4.2 Task-Specific Training

We instantiate three copies of the pre-trained LoRA adapter and give each task-specific training.

4.2.1 Asymmetric Retrieval Adapter

We used the prefix method described by Wang et al. (2022) to generate different query and document embeddings in `jina-embeddings-v4`.

Our training data consists of *hard negatives*. (Wang et al., 2022; Li et al., 2023) For every pair $(q_i, p_i) \in \mathcal{B}$ in a batch, p_i is intended to be a good match for q_i, and we presume that for all $(q_j, p_j) \in \mathcal{B}$ where $j \neq i$, p_j is a hard negative for q_i. We incorporate those negatives into the training process via an extended version of the $\mathcal{L}_{\text{NCE}}$ loss described in Günther et al. (2023).

We used existing datasets to create multimodal pairs for training, including Wiki-SS (Ma et al., 2024) and VDR multilingual,[2] but we also mined hard negatives from curated multimodal datasets.

4.2.2 Text Matching Adapter

We find that for symmetric semantic similarity tasks like text matching, training data with ground truth similarity values works best. As discussed in Sturua et al. (2024), we use the CoSENT[3] loss function $\mathcal{L}_{\text{co}}$ from Li and Li (2024), which operates on two pairs of text values with known ground truth similarity.

We used data from semantic textual similarity training datasets such as STS12 (Agirre et al., 2012) and SICK (Marelli et al., 2014), where ground truth similarity values are available. However, the amount of data in this format is very limited, so we enhanced our training data with pairs that do not have known similarity scores. For these pairs, we use the standard InfoNCE loss in place of the CoSENT loss.

4.2.3 Code Adapter

Code embeddings in `jina-embeddings-v4` are designed for natural language-to-code retrieval, code-to-code similarity search, and technical question answering. Because code embeddings do not involve image processing, the vision portion of `jina-embeddings-v4` is not affected by training the code retrieval LoRA adapter. `Qwen2.5-VL-3B-Instruct` was pre-trained on data including the StackExchangeQA[4] and the CodeSearchNet (Husain et al., 2020) datasets, giving it some capacity to support code embeddings before further adaptation.

Our LoRA training used the same method described in Section 4.2.1. Training triplets are derived from a variety of sources, including CodeSearchNet, CodeFeedback (Zheng et al., 2024), APPS (Hendrycks et al., 2021), and the CornStack dataset (Suresh et al., 2025).

5 JVDR: Visually Rich Document Retrieval Benchmark

To evaluate the performance of `jina-embeddings-v4` across a broad range of visually rich document retrieval tasks, we have produced a new benchmark collection and released it to the public.[5]

[2] https://huggingface.co/datasets/llamaindex/vdr-multilingual-train

[3] https://github.com/bojone/CoSENT

[4] https://github.com/laituan245/StackExchangeQA

[5] https://huggingface.co/collections/jinaai/jinavdr-visual-document-retrieval-

This new collection extends the ViDoRe benchmark by adding more than 30 additional tests designed to be compatible with ViDoRe. They span a broad range of domains (e.g. legal texts, historic documents, marketing materials), cover a variety of material types (e.g. charts, tables, manuals, printed text, maps) and query types (e.g. questions, facts, descriptions), and use up to 20 languages. These tests include re-purposed existing datasets, new manually-annotated data, and generated synthetic data. We employed LLM-based filtering to ensure all queries are relevant and reflective of realistic usage.[6]

We have adapted a number of existing VQA (visual question answering) and OCR datasets, modifying and restructuring them into appropriate query-document pairs. For some datasets, we used structured templates and generative language models to formulate text queries to match their contents. We also created benchmarks from available data to use unconventional querying techniques. We drew heavily on Wikimedia materials and other public data sources. For example, some datasets contain encyclopedia article fragments and image descriptions as queries to match with charts and maps. We obtained multilingual documents from Wikipedia and paired them with paragraphs that reference them. We used GitHub README files to create rendered images from Markdown-formatted rich texts and paired them with LLM-generated natural language descriptions in 17 languages.

We have also curated a number of human-annotated resources to better reflect real-world use cases. These include educational materials like lecture slides, commercial catalogs, marketing materials, and institutional documents. We paired these documents with human-written queries.

We have been attentive, in constructing JVDR, to the lack of diversity that often plagues information retrieval benchmarks. We cannot commission human-annotated datasets for everything and have had recourse to generative AI to fill in the gaps.

We obtained a number of datasets from primarily European sources containing scans of historical, legal, and journalistic documents in German, French, Spanish, Italian, and Dutch, and public service documents and commercial catalogs in Hindi, Russian, and other often underrepresented languages. We used Qwen2[7] to generate queries for these documents. In several cases, we introduced cross-language queries synthesized using advanced multilingual LLMs, in order to better measure cross-language retrieval.

For a comprehensive overview of the individual benchmarks, see Appendix A.3.

6 Evaluation

Table 2 provides an overview of benchmark averages for `jina-embeddings-v4` and other embedding models.

6.1 Text Retrieval

For MTEB and MMTEB benchmarks (Enevoldsen et al., 2025), we used the asymmetric retrieval adapter except for some symmetric retrieval tasks like ArguAna,[8] where we used the text matching adapter instead. We evaluated our model on retrieval tasks that involve long text documents using the *LongEmbed* benchmark (Zhu et al., 2024). We also tested the text matching adapter on MTEB STS and MMTEB STS benchmarks.

Results for these benchmarks are tabulated in Appendix A.1. The performance of `jina-embeddings-v4` is broadly comparable with the state-of-the-art. For long document performance, `jina-embeddings-v4` significantly outpaces competing models except `voyage-3`.

6.2 Code Retrieval

To assess performance on code retrieval, we evaluated the model on the MTEB-CoIR benchmark (Li et al., 2025). The results are reported in Table A6. `jina-embeddings-v4` is competitive with the state-of-the-art in general-purpose embedding models, but the specialized `voyage-code` model has somewhat better benchmark performance.

6.3 CLIP Benchmark

To evaluate the model's performance on typical text-to-image search tasks, we used the CLIP Benchmark.[9] The results are tabulated in Appendix A.2.

`jina-embeddings-v4` generally outperforms CLIP-style models on these benchmarks, although `nllb-siglip-large` performs somewhat higher on the Crossmodal3600 benchmark (Thapliyal et al., 2022) (see Table A8) because it supports low-resource languages not included in training the `Qwen2.5-VL-3B-Instruct` backbone model.

`684831c022c53b21c313b449`

[6] See A.5 for the specific prompts.

[7] `https://huggingface.co/collections/Qwen/qwen2-6659360b33528ced941e557f`

[8] `https://huggingface.co/datasets/mteb/arguana`

[9] `https://github.com/LAION-AI/CLIP_benchmark`

Model	JVDR	ViDoRe	CLIPB	MMTEB	MTEB-en	COIR	LEMB	STS-m	STS-en
jina-embeddings-v4 (single)	73.98	84.11	84.11	66.49	55.97	71.59	67.11	72.70	85.89
jina-embeddings-v4 (multi)	80.55	90.17							
text-embedding-3-large	–	–	–	59.27	57.98	62.36	52.42	70.17	81.44
bge-m3	–	–	–	55.36			58.73		
multilingual-e5-large-instruct	–	–	–	57.12	53.47		41.76		
jina-embeddings-v3	47.82	26.02	–	58.58	54.33	55.07	55.66	75.77	85.82
voyage-3	–	–	–	66.13	53.46	67.23	74.06	68.33	78.59
gemini-embedding-001	–	–	–	67.71	64.35	73.11		78.35	85.29
jina-embedings-v2-code	–	–	–			52.24			
voyage-code	–	–	–			77.33			
nllb-clip-large-siglip			83.19						
jina-clip-v2	40.52	53.61	81.12						
colpali-v1.2 (late)	63.80	83.90							
dse-qwen2-2b-mrl-v1 (dense)	67.25	85.80							
voyage-multimodal-v3 (dense)		84.24							

Table 2: Average Retrieval Scores of Embedding Models on Various Benchmarks.

Task Acronyms: VidoRE = ViDoRe, CLIPB = CLIP Benchmark, MMTEB = MTEB(Multilingual, v2) Retrieval Tasks, MTEB-EN = MTEB(eng, v2) Retrieval Tasks, COIR = CoIR Code Retrieval, LEMB = LongEmbed, STS-m = MTEB(Multilingual, v2) Semantic Textual Similarity Tasks, STS-en = MTEB(eng, v2) Semantic Textual Similarity Tasks

Average Calculation: For JVDR and ViDoRE, we calculate the average for the multilingual tasks first and consider this as a single score before calculating the average across all tasks. Scores are nDCG@5 for JVDR and ViDoRe, Recall@5 for CLIPB, nDCG@10 for MMTEB, MTEB-en, COIR, and LEMB, and Spearman coefficient for STS-m and STS-en.

Evaluation of Text Retrieval Models on JVDR: For evaluating text retrieval models on JVDR, we used EasyOCR (https://github.com/JaidedAI/EasyOCR) and the provided extracted texts from the original ViDoRe datasets.

6.4 Visually Rich Document Benchmarks

Appendix A.4 tabulates the results of evaluating jina-embeddings-v4 on our new JVDR benchmark. Table A12 provides a comparison with other models. jina-embeddings-v4 excels at visually rich document tasks, and is currently the state-of-the-art in both single- and multi-vector mode. These results suggest that other models underperform on visually rich document tasks that do not closely resemble the ones in the ViDoRe benchmark.

6.5 Modality Gap

The so-called *modality gap* is dramatically reduced with jina-embeddings-v4 because of its cross-model encoder. We measure the cross-modal alignment score of a multimodal embedding model as the average of cosine similarities of matching pairs of image and text embeddings. Table A10 displays this score for jina-embeddings-v4 and CLIP-style models for data sampled from the Flickr30K,[10] MSCOCO, (Lin et al., 2014) and CIFAR-100[11] datasets. These results confirm that jina-embeddings-v4 generates a far better aligned cross-modal embedding space than CLIP-style models, as can be seen in Figure 2 in the appendix.

7 Conclusion

We present jina-embeddings-v4, a state-of-the-art multimodal and multilingual embedding model designed for a wide range of tasks, including semantic text retrieval, text-to-image retrieval, text-to-visually-rich document retrieval, and code search. The model achieves strong performance using single-vector representations and demonstrates even greater effectiveness with multi-vector representations, particularly in visually rich document retrieval. jina-embeddings-v4 aligns representations across modalities into a single, shared semantic space, sharply reducing structural gaps between modalities compared to CLIP-style dual-tower models, enabling more effective cross-modal retrieval.

We also present JVDR, a novel benchmark for visually rich documents that dramatically extends the ViDoRe benchmark by including much more diverse data types, more languages, and more kinds of queries and semantic similarity tests. We have made this benchmark available to the public for future work.

[10] https://www.kaggle.com/datasets/adityajn105/flickr30k

[11] https://www.kaggle.com/datasets/fedesoriano/cifar100

Limitations

jina-embeddings-v4 is a model that extends Qwen2.5-VL-3B-Instruct and is limited by its original training. As a result, its performance on many languages is not comparable to the state-of-the-art and it may not perform well on materials too far outside of its training. Furthermore, highly domain-specialized models may have significantly better performance at specific tasks.

Although this model is theoretically capable of embedding text and image input together, it has not been trained for such input. It has also not been trained for image-image retrieval or semantic similarity, and may underperform on those tasks.

JVDR is not a rigorously representative data collection. It is a significant expansion over previous related benchmarks, but this is a new area for embeddings research, and JVDR undoubtedly has gaps and shortcomings that usage will reveal.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *SEM 2012: 1st Joint Conference on Lexical and Computational Semantics (SemEval-2012)*.

Shuai Bai, Keqin Chen, et al. 2025. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*.

Yihao Ding, Soyeon Caren Han, Jean Lee, and Eduard Hovy. 2024. Deep Learning based Visually Rich Document Content Understanding: A Survey. *arXiv preprint arXiv:2408.01287*.

Kenneth Enevoldsen, Isaac Chung, et al. 2025. MMTEB: Massive Multilingual Text Embedding Benchmark. In *13th International Conference on Learning Representations (ICLR 2025)*.

Sedigheh Eslami and Gerard de Melo. 2025. Mitigate the Gap: Improving Cross-Modal Alignment in CLIP. In *13th International Conference on Learning Representations (ICLR 2025)*.

Manuel Faysse, Hugues Sibille, et al. 2025. ColPali: Efficient Document Retrieval with Vision Language Models. In *13th International Conference on Learning Representations (ICLR 2025)*.

Michael Günther, Jackmin Ong, et al. 2023. Jina Embeddings 2: 8192-Token General-Purpose Text Embeddings for Long Documents. *arXiv preprint arXiv:2310.19923*.

Dan Hendrycks, Steven Basart, et al. 2021. Measuring Coding Challenge Competence With APPS . In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the Knowledge in a Neural Network. In *28th Conference on Neural Information Processing Systems (NIPS 2014)*.

Edward J. Hu, Yelong Shen, et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *10th International Conference on Learning Representations (ICLR 2022)*.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2020. CodeSearchNet Challenge: Evaluating the State of Semantic Code Search. *arXiv preprint arXiv:1909.09436*.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*.

Aditya Kusupati, Gantavya Bhatt, et al. 2022. Matryoshka Representation Learning. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Xiangyang Li, Kuicai Dong, et al. 2025. CoIR: A Comprehensive Benchmark for Code Information Retrieval Models. In *63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*.

Xianming Li and Jing Li. 2024. AoE: Angle-optimized Embeddings for Semantic Textual Similarity. In *62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.

Zehan Li, Xin Zhang, et al. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *arXiv preprint arXiv:2308.03281*.

Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. 2022. Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Tsung-Yi Lin, Michael Maire, et al. 2014. Microsoft COCO: Common Objects in Context. In *2014 European Conference on Computer Vision (ECCV 2014)*.

Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. 2024. Unifying Multimodal Retrieval via Document Screenshot Embedding. In *2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*.

Sourab Mangrulkar, Sylvain Gugger, et al. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning Methods. Online. https://github.com/huggingface/peft.

Marco Marelli, Stefano Menini, et al. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *9th International Conference on Language Resources and Evaluation (LREC'14)*.

Alec Radford, Jong Wook Kim, et al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *38th International Conference on Machine Learning (ICML 2021)*.

Simon Schrodi, David T Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. 2025. Two Effects, One Trigger: On the Modality Gap, Object Bias, and Information Imbalance in Contrastive Vision-Language Models. In *13th International Conference on Learning Representations (ICLR 2025)*.

Saba Sturua, Isabelle Mohr, et al. 2024. jina-embeddings-v3: Multilingual Embeddings With Task LoRA. *arXiv preprint arXiv:2409.10173*.

Tarun Suresh, Revanth Gangi Reddy, et al. 2025. CoRNStack: High-Quality Contrastive Data for Better Code Retrieval and Reranking. *arXiv preprint arXiv:2412.01007*.

Ashish V. Thapliyal, Jordi Pont Tuset, Xi Chen, and Radu Soricut. 2022. Crossmodal-3600: A Massively Multilingual Multimodal Evaluation Dataset. In *2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*.

Liang Wang, Nan Yang, et al. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint arXiv:2212.03533*.

Chenghao Xiao, Isaac Chung, et al. 2025. MIEB: Massive Image Embedding Benchmark. *arXiv preprint arXiv:2504.10471*.

Tianyu Zheng, Ge Zhang, et al. 2024. OpenCodeInterpreter: Integrating Code Generation with Execution and Refinement. In *62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.

Dawei Zhu, Liang Wang, et al. 2024. LongEmbed: Extending Embedding Models for Long Context Retrieval. In *2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*.

A Appendix

A.1 MTEB and MMTEB

Table A1: Evaluation Results on MTEB Retrieval Tasks (nDCG@10%)

Model	Arg	CQG	CQU	CFHN	FEV	FiQA	HPQA	SCI	TREC	TOU	AVG
jina-embeddings-v4†	67.07	57.59	42.95	34.57	87.16	46.51	69.01	21.47	80.36	52.41	55.91
jina-embeddings-v3†	54.33	58.02	43.52	43.14	89.90	47.35	64.70	19.92	77.74	55.28	55.39
jina-embeddings-v2-base-en	44.18	56.52	38.66	23.77	73.41	41.58	63.24	19.86	65.91	63.35	49.05
jina-embedding-l-en-v1	48.30	51.68	38.66	25.93	71.16	41.02	57.26	18.54	60.34	62.34	47.52
multilingual-e5-large	54.36	58.70	39.89	26.00	83.79	43.82	70.55	17.45	71.15	49.59	51.53
e5-mistral-7b-instruct	61.65	63.52	46.75	28.50	86.99	56.81	73.21	16.32	87.03	55.44	57.62
text-embedding-3-large	57.99	65.40	50.02	30.10	88.53	55.00	71.66	23.07	79.56	58.42	57.98
gemini-embedding-001	86.44	70.68	53.69	31.06	88.98	61.78	87.01	25.15	86.32	52.39	64.35

†using the text-matching adapter

Tasks: Arg: ArguAna, CQG: CQADupstackGamingRetrieval, CQU: CQADupstackUnixRetrieval, CFHN: ClimateFEVERHardNegatives, FEV: FEVERHardNegatives, FiQA: FiQA2018, HPQA: HotpotQAHardNegatives, SCI: SCIDOCS, TREC: TRECCOVID, TOU: Touche2020Retrieval.v3

Table A2: Evaluation Results on MMTEB Retrieval Tasks (nDCG@10%)

Model	Avg	AI	Arg	Bel	Cov	Hag	PK	LB	MIR	ML	SD	SQA	SO	TC	STC	TR	TW	Wiki	WG
JinaV4	66.5	50.2	67.1	74.3	80.2	98.8	69.8	94.8	61.2	74.9	21.5	30.2	91.9	80.4	59.5	1.3	84.4	88.5	67.3
JinaV3	58.6	32.8	54.3	73.4	78.6	98.7	38.0	93.4	62.6	73.4	19.8	0.7	90.8	77.7	39.2	0.6	73.0	89.1	18.6
BGE-M3	55.4	29.0	54.0	78.2	77.5	98.8	59.0	90.3	69.6	74.8	16.3	7.5	80.6	54.9	21.9	1.0	37.8	89.9	41.7
CohV3	59.2	29.7	55.1	81.1	77.1	98.8	38.2	93.8	68.0	76.1	19.3	4.7	89.4	83.4	24.2	0.9	75.8	90.9	58.4
Gem001	68.1	48.8	86.4	90.7	79.1	99.3	38.5	96.0	70.4	84.2	25.2	10.3	96.7	86.3	51.1	3.0	98.0	94.2	60.5
TE3L	61.1	42.0	58.0	68.8	68.4	99.1	69.8	95.2	56.9	73.2	23.1	7.4	92.4	79.6	31.1	2.1	81.4	89.2	29.1
Voy3	66.0	42.5	61.0	76.5	88.5	98.6	94.8	94.5	57.7	75.7	21.4	10.7	94.3	80.5	49.2	1.2	85.7	89.7	67.7
VoyM2	–	45.0	61.8	–	–	98.9	97.0	95.9	–	–	22.5	10.2	–	80.1	–	1.4	87.3	–	39.1

Model abbreviations: JinaV4: jina-embeddings-v4, JinaV3: jina-embeddings-v3, BGE-M3: bge-m3, CohV3: cohere-embed-multilingual-v3, Gem001: gemini-embedding-001, TE3L: text-embedding-3-large, Voy3: voyage-3, VoyM2: voyage-multilingual-2.
Tasks: Avg: Mean nDCG@10% for all tasks, AI: AILAStatutes, Arg: ArguAna, Bel: BelebeleRetrieval, Cov: CovidRetrieval, Hag: HagridRetrieval, PK: LEMBPasskeyRetrieval, LB: LegalBenchCorporateLobbying, MIR: MIRACLRetrievalHardNegatives, ML: MLQARetrieval, SD: SCIDOCS, SQA: SpartQA, SO: StackOverflowQA, TC: TREC-COVID, STC: StatcanDialogueDatasetRetrieval, TR: TempReasonL1, TW: TwitterHjerneRetrieval, Wiki: WikipediaRetrievalMultilingual, WG: WinoGrande

Table A3: Retrieval performance on MTEB LongEmbed (nDCG@10%)

Model	Avg	NaQA	Needle	Passkey	QMSum	SummScreen	Wikim
jina-embeddings-v4	67.11	57.52	51.75	65.50	46.49	96.30	85.08
jina-embeddings-v3	55.66	34.30	64.00	38.00	39.34	92.33	66.02
multilingual-e5-large	40.44	24.22	28.00	38.25	24.26	71.12	56.80
multilingual-e5-large-instruct	41.76	26.71	29.50	37.75	26.08	72.75	57.79
bge-m3	58.73	45.76	40.25	59.00	35.54	94.09	77.73
cohere-embed-english-v3	42.11	25.04	30.50	38.50	23.82	75.77	59.03
text-embedding-3-large	52.42	44.09	29.25	69.75	32.49	84.80	54.16
voyage-3	74.07	54.12	57.75	94.75	51.05	97.82	88.90
voyage-3-lite	71.41	51.67	54.00	84.75	53.01	96.71	88.34
voyage-multilingual-2	79.17	64.69	75.25	97.00	51.50	99.11	87.49

Tasks: Avg: Mean nDCG@10% for all tasks, NaQA: LEMBNarrativeQARetrieval, Needle: LEMBNeedleRetrieval, Passkey: LEMBPasskeyRetrieval, QMSum: LEMBQMSumRetrieval, SummScreen: LEMBSummScreenFDRetrieval, Wikim: LEMBWikimQARetrieval

Table A4: STS performance on MTEB v2 (Spearman correlation %).

Model	Avg	BIO	SICK-R	STS12	STS13	STS14	STS15	STS17	STS22	STSB
jina-embeddings-v4	85.89	89.21	89.23	83.50	88.61	84.77	89.69	88.71	70.71	88.58
jina-embeddings-v3	85.82	88.69	89.62	82.44	89.49	84.95	89.32	90.01	68.45	89.43
multilingual-e5-large	81.39	84.57	80.23	80.02	81.55	77.72	89.31	88.12	63.66	87.29
bge-m3	80.61	–	79.72	78.73	79.60	79.00	87.81	87.13	67.99	84.87
cohere-embed-English-3	82.40	83.50	81.27	74.37	85.20	80.98	89.23	90.34	68.18	88.55
cohere-embed-multilingual-v3	83.05	85.01	82.18	77.62	85.16	80.02	88.92	90.09	69.63	88.79
gemini-embedding-001	85.29	88.97	82.75	81.55	89.89	85.41	90.44	91.61	67.97	89.08
text-embedding-3-large	81.44	84.68	79.00	72.84	86.10	81.15	88.49	90.22	66.89	83.56
voyage-3	78.59	87.92	79.63	69.52	80.56	73.33	80.39	86.81	69.60	79.53
voyage-large-2	82.63	89.13	79.78	72.94	83.11	77.21	85.30	88.77	–	84.78
voyage-multilingual-v2	76.98	87.11	78.97	67.30	80.09	71.98	78.07	86.52	67.02	75.79

Tasks: Avg: Mean Spearman Correlation % for all tasks, BIO: BIOSSES, STS22: STS22v2, STSB: STSBenchmark

Table A5: STS performance on MMTEB v2 (Spearman correlation %).

Model	Avg	Faro	FinP	Ind	JSCK	SCKR	STS12	STS13	STS14	STS15	STS17	STS22	STSB	STSES	Sem
JinaV4	72.7	72.3	14.4	35.2	80.3	89.2	83.5	88.6	84.8	89.7	88.7	70.7	88.6	75.3	56.5
JinaV3	75.8	80.8	22.4	54.7	78.2	89.6	82.4	89.5	84.9	89.3	85.9	71.1	89.4	77.9	64.6
BGE-M3	73.0	77.8	30.4	52.1	79.2	79.7	78.7	79.6	79.0	87.8	79.7	70.0	84.9	77.5	65.4
CohV3	73.8	76.0	28.2	46.7	77.2	82.2	77.6	85.2	80.0	88.9	90.1	69.4	88.8	78.8	63.8
Gem001	78.3	86.1	28.6	62.9	85.0	82.8	81.5	89.9	85.4	90.4	88.6	71.7	89.1	81.8	73.1
TE3L	70.2	75.0	23.5	12.6	81.2	79.0	72.8	86.1	81.2	88.5	90.2	69.3	83.6	74.2	65.2
Voy3	68.3	72.5	22.5	41.6	71.8	79.6	69.5	80.6	73.3	80.4	76.2	71.9	79.5	72.5	64.7
VoyM2	68.0	74.4	27.1	35.0	75.9	79.0	67.3	80.1	72.0	78.1	77.1	69.0	75.8	76.7	64.9

Model abbreviations: JinaV4: jina-embeddings-v4, JinaV3: jina-embeddings-v3, BGE-M3: bge-m3, CohV3: cohere-embed-multilingual-v3, Gem001: gemini-embedding-001, TE3L: text-embedding-3-large, Voy3: voyage-3, VoyM2: voyage-multilingual-2.
Tasks: Avg: Mean Spearman Correlation % for all tasks, Faro: FaroeseSTS, FinP: FinParaSTS, Ind: IndicCrosslingualSTS, JSCK: JSICK, SCKR: SICK-R, STS22: STS22v2, STSB: STSBenchmark, Sem: SemRel24STS

Table A6: Performance on MTEB Code Information Retrieval (MTEB-CoIR) (nDCG@10%).

Model	Avg	AppsR	CCSN	CodeMT	CodeST	CodeSN	CodeTO	CodeTD	CosQA	StackO	SynSQL
jina-embeddings-v4	71.59	76.08	84.05	70.60	85.06	83.69	89.34	44.19	31.48	93.45	70.45
jina-embeddings-v3	55.07	29.01	–	59.67	78.14	53.18	77.37	30.91	35.34	90.79	41.27
jina-embeddings-v2-code	52.24	16.37	83.97	44.40	68.66	59.62	75.68	27.25	41.92	89.26	46.99
cohere-embed-English-3	51.36	13.72	–	47.02	74.82	52.81	65.28	31.38	30.65	89.35	57.20
cohere-embed-mult.-v3	54.31	31.91	–	42.91	74.19	57.57	70.25	30.14	32.58	89.42	59.79
gemini-embedding-001	73.11	93.75	81.06	56.28	85.33	84.69	89.53	31.47	50.24	96.71	69.96
text-embedding-3-large	62.36	28.37	–	68.92	80.42	73.18	84.25	34.23	31.00	92.44	68.45
voyage-3	67.23	73.03	–	66.69	83.02	77.87	89.92	33.92	28.70	94.34	57.56
voyage-code-3	77.33	93.62	89.35	93.58	90.67	90.09	94.96	38.57	34.45	97.17	62.87

Tasks: Avg: Mean nDCG@10% for all tasks, AppsR: AppsRetrieval, COIR: COIRCodeSearchNetRetrieval, CodeMT: CodeFeedbackMT, CodeST: CodeFeedbackST, CodeSN: CodeSearchNetCCRetrieval, CodeTO: CodeTransOceanContest, CodeTD: CodeTransOceanDL, StackO: StackOverflowQA, SynSQL: SyntheticText2SQL

A.2 CLIP

Table A7: Cross-modal (Text-to-image) retrieval performance (Recall@5%) on the CLIP benchmark.

Model	Avg	flickr30k	mscoco_captions	crossmodal3600	xtd10
nllb-clip-large-siglip	83.19	**92.24**	70.84	**82.07**	87.60
jina-clip-v2	81.12	89.84	68.35	81.43	84.87
jina-embeddings-v4	**84.11**	91.36	**76.18**	79.42	**89.46**

Avg: Mean Recall@5% over all 4 tasks.

Table A8: Text-to-image retrieval performance (Recall@5%) on **crossmodal3600** for all supported languages.

Language	jina-embeddings-v4	jina-clip-v2	nllb-clip-large-siglip
average	79.42	81.43	**82.07**
ar	75.75	73.56	**78.92**
bn	57.97	63.78	**75.19**
da	80.47	85.39	**87.14**
de	**91.75**	91.25	89.56
el	66.50	75.03	**77.83**
en	**76.47**	75.83	73.11
es	**83.64**	**83.64**	82.64
fi	66.67	82.83	**86.42**
fr	88.69	**88.78**	87.86
hi	47.81	55.25	**60.31**
id	**87.41**	84.22	86.31
it	87.97	**88.33**	85.94
ja	**91.22**	87.03	86.06
ko	**82.19**	78.81	78.75
nl	81.00	**82.56**	81.69
no	71.94	81.08	**82.69**
pl	80.86	**84.00**	82.72
pt	81.42	82.42	**82.69**
ro	84.33	89.36	**90.03**
ru	**90.28**	88.97	86.44
sv	72.58	78.06	**79.33**
th	**83.36**	81.61	81.14
tr	73.08	81.31	**83.47**
uk	86.28	**88.56**	85.44
vi	**88.81**	86.64	85.56
zh	**86.67**	78.97	76.56

Table A9: Text-to-image retrieval performance (Recall@5%) on **xtd10** for all supported languages.

Language	jina-embeddings-v4	jina-clip-v2	nllb-clip-large-siglip
average	**89.46**	84.87	87.60
de	**92.10**	85.70	88.30
en	**93.10**	89.40	89.40
es	**91.50**	85.90	88.20
fr	**91.30**	85.10	87.70
it	**92.20**	85.80	89.30
ko	**86.30**	82.10	85.20
pl	89.10	86.50	**89.40**
ru	**91.50**	81.10	83.40
tr	84.70	83.70	**88.30**
zh	82.80	83.40	**86.80**

Table A10: Comparison of cross-modal alignment scores on 1K of random samples from each dataset.

Model	**Flickr30K**	**MSCOCO**	**CIFAR-100**
OpenAI-CLIP	0.15	0.14	0.20
`jina-clip-v2`	0.38	0.37	0.32
`jina-embeddings-v4`	0.71	0.72	0.56

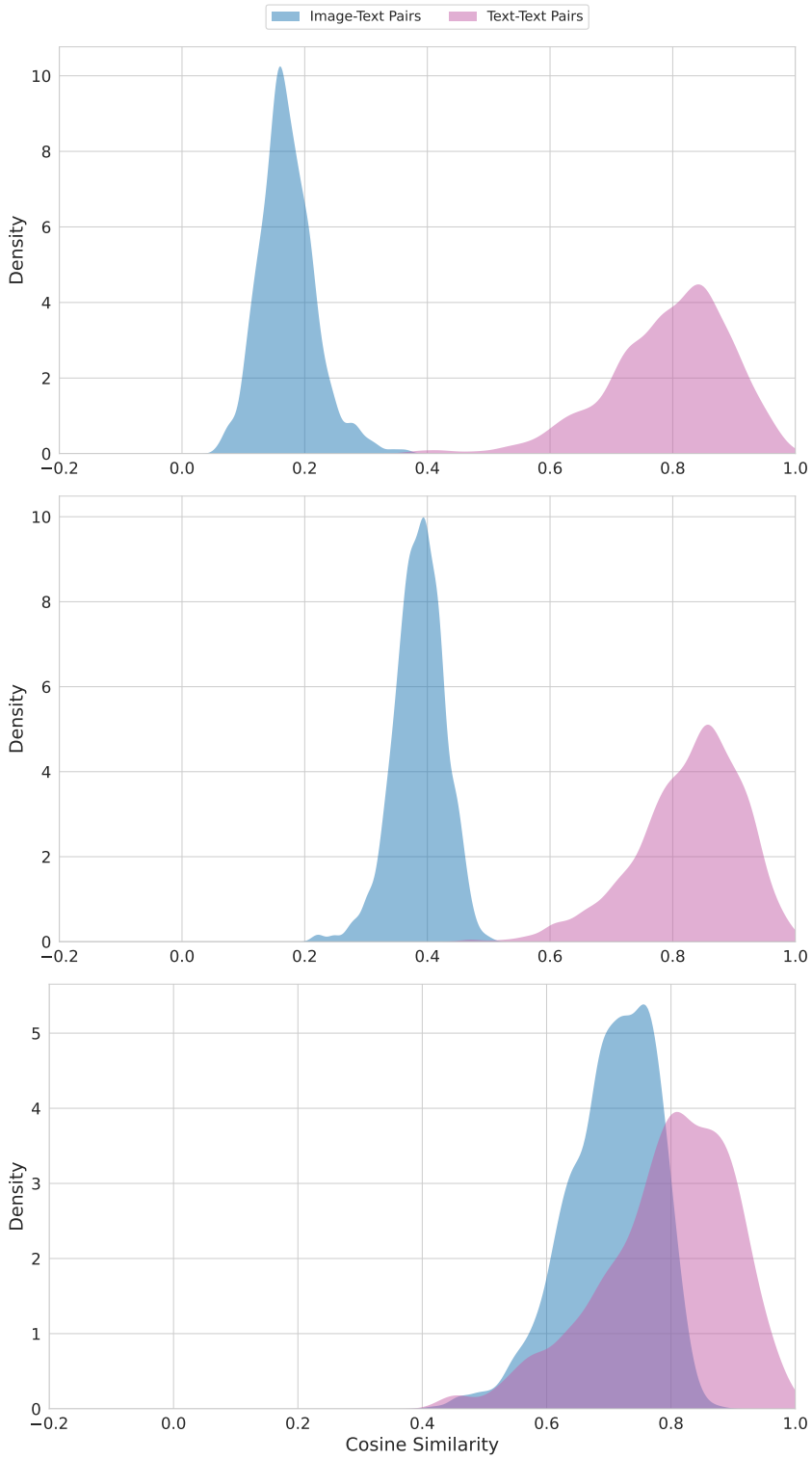

Figure 2: Distribution of the cosine similarities of the paired image-text embeddings versus paired text-text embeddings from the Flickr8K dataset. **Top**: OpenAI CLIP, **Middle**: `jina-clip-v2`, **Bottom**: `jina-embeddings-v4`

A.3 Datasets in the JVDR Benchmark

Table A11: Overview of the Dataset Collection

Dataset Name	Domain	Document Format	Query Format	Number of Queries / Documents	Languages
airbnb-synthetic-retrieval†	Housing	Tables	Instruction	4953 / 10000	ar, de, en, es, fr, hi, hu, ja ru, zh
arabic_chartqa_ar	Mixed	Charts	Question	745 / 342	ar
arabic_infographicsvqa_ar	Mixed	Illustrations	Question	120 / 40	ar
automobile_catalogue_jp	Marketing	Catalog	Question	45 / 15	ja
arxivqa	Science	Mixed	Question	30 / 499	en
beverages_catalogue_ru	Marketing	Digital Docs	Question	100 / 34	ru
ChartQA	Mixed	Charts	Question	996 / 834	en
CharXiv-en	Science	Charts	Question	999 / 1000	en
docvqa	Mixed	Scans	Question	39 / 499	en
donut_vqa	Medical	Scans / Handwriting	Question	704 / 800	en
docqa_artificial_intelligence	Software / IT	Digital Docs	Question	70 / 962	en
docqa_energy	Energy	Digital Docs	Question	69 / 971	en
docqa_gov_report	Government	Digital Docs	Question	77 / 970	en
docqa_healthcare_industry	Medial	Digital Docs	Question	90 / 961	en
europeana-de-news	Historic	Scans / News Articles	Question	379 / 137	de
europeana-es-news	Historic	Scans / News Articles	Question	474 / 179	es
europeana-fr-news	Historic	Scans / News Articles	Question	237 / 145	fr
europeana-it-scans	Historic	Scans	Question	618 / 265	it
europeana-nl-legal	Legal	Scans	Question	199 / 244	nl
github-readme-retrieval-multilingual†	Software / IT	Markdown Docs	Description	16953 / 16998	ar, bn, de, en, es, fr, hi, id, it, ja, ko, nl pt, ru, th, vi, zh
hindi-gov-vqa	Governmental	Digital Docs	Question	454 / 337	hi
hungarian_doc_qa_hu	Mixed	Digital Docs	Question	54 / 51	hu
infovqa	Mixed	Illustrations	Question	363 / 500	en
jdocqa	News	Digital Docs	Question	744 / 758	ja
jina_2024_yearly_book	Software / IT	Digital Docs	Question	75 / 33	en
medical-prescriptions	Medical	Digital Docs	Question	100 / 100	en
mpmqa-small	Manuals	Digital Docs	Question	155 / 782	en
MMTab	Mixed	Tables	Fact	987 / 906	en
openai-news	Software / IT	Digital Docs	Question	31 / 30	en
owid_charts_en	Mixed	Charts	Question	132 / 937	en
plotqa	Mixed	Charts	Question	610 / 986	en
ramen_benchmark_jp	Marketing	Catalog	Question	29 / 10	ja
shanghai_master_plan	Governmental	Digital Docs	Question / Key Phrase	57 / 23	zh, en
wikimedia-commons-documents-ml†	Mixed	Mixed	Description	15593 / 15217	ar, bn, de, en, es, fr, hi, hu, id, it, ja, ko, my, nl, pt, ru, th, ur, vi, zh
shiftproject	Environmental Documents	Digital Docs	Question	89 / 998	fr
stanford_slide	Education	Slides	Question	14 / 994	en
student-enrollment	Demographics	Charts	Question	1000 / 489	en
tabfquad	Mixed	Tables	Question	126 / 70	fr, en
table-vqa	Science	Tables	Question	992 / 385	en
tatqa	Finance	Digital Docs	Question	121 / 270	en
tqa	Education	Illustrations	Question	981 / 393	en
tweet-stock-synthetic-retrieval†	Finance	Charts	Question	6278 / 10000	ar, de, en, es, fr, hi, hu, ja, ru, zh
wikimedia-commons-maps	Mixed	Maps	Description	443 / 451	en

†For multilingual datasets, the total number of queries and documents is the sum across all language-specific splits.

A.4 JVDR (Visual Document Retrieval) Benchmark Results

Table A12: Overview of JVDR Results

Task	bm25 + OCR	jev3 + OCR	j-clip-v2	colpali-v1.2	dse-qwen2-2b-mrl-v1	jev4-single	jev4-multi
Average	46.88	48.97	40.96	65.39	68.89	75.47	81.52
medical-prescriptions	38.18	38.12	15.68	66.22	38.86	80.95	97.69
stanford_slide	81.78	95.28	91.48	100.0	100.0	100.0	97.16
donut_vqa	19.39	2.59	1.46	34.12	25.31	78.60	74.08
table-vqa	55.22	63.04	36.34	80.98	85.70	86.57	89.21
ChartQA	28.39	31.47	39.73	54.45	58.38	70.88	71.80
tqa	50.11	24.40	27.80	63.03	65.35	65.44	68.46
openai-news	76.63	87.30	70.05	94.81	93.75	93.97	96.43
europeana-de-news	11.26	12.02	11.18	35.20	44.32	48.89	63.76
europeana-es-news	51.99	43.82	12.95	45.70	60.66	60.81	80.70
europeana-it-scans	39.11	38.77	16.54	58.70	54.28	58.01	73.29
europeana-nl-legal	39.38	34.24	11.30	39.13	33.12	42.77	59.82
hindi-gov-vqa	1.83	7.51	5.21	11.43	10.19	15.32	22.49
automobile_catalogue_jp	20.92	50.39	32.54	35.72	66.44	72.22	81.32
beverages_catalogue_ru	11.05	14.09	39.66	68.47	80.32	85.68	87.73
ramen_benchmark_jp	28.02	63.37	41.28	52.03	51.66	90.77	94.65
jdocqa_jp_ocr	1.64	7.85	19.94	35.68	67.00	75.63	82.42
hungarian_doc_qa	34.28	57.84	50.44	68.83	55.25	74.64	75.56
arabic_chartqa_ar	9.32	8.63	6.62	26.92	49.35	62.16	66.64
arabic_infographicsvqa_ar	13.26	13.43	50.36	34.76	71.72	85.38	93.21
owid_charts_en	66.19	62.10	57.71	78.17	84.26	92.06	92.29
arxivqa	56.73	54.41	83.41	92.54	93.33	95.44	95.44
docvqa	81.11	50.81	45.29	90.38	86.28	83.06	92.98
shiftproject	62.42	70.25	31.85	75.18	78.54	82.55	91.13
docqa_artificial_intelligence	91.68	82.98	66.52	96.09	97.52	96.43	98.04
docqa_energy	89.97	76.97	65.56	96.03	90.08	88.66	96.28
docqa_gov_report	87.20	82.72	68.84	92.92	94.19	92.03	95.97
docqa_healthcare_industry	86.44	86.88	68.13	93.14	96.14	94.62	97.51
tabfquad	45.67	80.49	47.04	89.18	92.38	95.57	95.38
mpmqa_small	85.54	67.39	59.72	88.88	81.62	80.44	91.28
jina_2024_yearly_book	87.67	85.98	77.12	95.77	93.39	94.29	98.17
wikimedia-commons-maps	5.37	5.06	20.67	27.46	33.06	40.23	53.45
plotqa	61.13	51.44	24.05	70.58	75.99	77.48	78.75
MMTab	74.82	74.06	44.54	84.66	86.04	86.08	90.03
CharXiv-en	46.85	41.47	56.28	79.64	83.86	83.00	87.66
student-enrollment	1.05	1.30	0.70	3.95	4.09	8.04	11.55
tatqa	75.62	49.88	44.23	82.57	80.97	80.14	92.76
shanghai_master_plan	12.69	92.67	75.28	88.87	92.56	95.53	97.41
europeana-fr-news	24.55	23.69	16.43	30.33	38.23	36.66	50.16
infovqa	73.61	75.09	63.38	87.53	92.64	92.16	96.69

Models: bm25+OCR: BM25 with EasyOCR, **jev3 + OCR**: `jina-embeddings-v3` with EasyOCR, colpali-v1.2: ColPALI-v1.2, dse-qwen2- 2b-mrl-v1: DSE-QWen2-2b-MRL-V1, je4-single: `jina-embeddings-v4` single-vector, jev4-multi: `jina-embeddings-v4` multi-vector

Table A13: Retrieval performance on ViDoRe (nDCG@5%).

Model	Avg	AQA	DVQA	InfoVQA	Shift	AI	Energy	Gov	Health	TabFQ	TQA
OCR + jina-embeddings-v3	26.02	26.31	12.62	32.79	14.18	22.84	27.47	31.16	45.78	44.54	2.53
jina-clip-v2	53.61	68.33	27.62	60.60	34.12	66.55	64.69	67.47	68.38	46.89	31.43
voyage-multimodal-3	84.20	84.90	55.60	85.40	78.70	94.50	89.50	96.00	95.10	92.80	69.90
colpali-v1.2	83.90	78.00	57.20	82.80	79.10	98.10	95.20	94.80	96.70	89.70	68.10
dse-qwen2-2b-mrl-v1	85.80	85.60	57.10	88.10	82.00	97.50	92.90	96.00	96.40	93.10	69.40
OCR + bm25	65.50	31.60	36.80	62.90	64.30	92.80	85.90	83.90	87.20	46.50	62.70
siglip-so400m-patch14-384	51.40	43.20	30.30	64.10	18.70	62.50	65.70	66.10	79.10	58.10	26.20
jina-embeddings-v4 (single)	84.11	83.57	50.54	87.85	84.07	97.16	91.66	91.48	94.92	94.48	65.35
jina-embeddings-v4 (multi)	90.17	88.95	59.98	93.57	92.35	99.26	96.76	96.95	98.39	95.13	80.34

Tasks: Avg: Mean nDCG@5% over all tasks, AQA: ArxivQA, Shift: Shift Project, DVQA: DocVQA, InfoVQA: InfographicVQA, AI: Artificial Intelligence, Gov: Government Reports, Health: Healthcare Industry, TabFQ: TabFQuad, TQA: TAT-DQA

Table A14: Retrieval performance on ViDoRe V2 (nDCG@5%).

Model	Avg	Bio	ESG-En	ESG-Multi	Econ
colpali-v1.2	50.7	54.1	54.3	50.7	43.7
jina-embeddings-v4 (single)	50.4	57.0	52.6	39.5	52.6
jina-embeddings-v4 (multi)	58.2	60.9	65.1	51.8	55.1

Tasks: Avg: Mean nDCG@5% over all tasks, Bio: MIT Biomedical Multilingual, ESG-En: ESG Restaurant Human English, ESG-Multi: ESG Restaurant Synthetic Multilingual, Econ: Economics Macro Multilingual.

Table A15: Wikimedia Commons Retrieval Benchmark Results

Language	bm25 + OCR	jev3 + OCR	j-clip-v2	colpali-v1.2	dse-qwen2-2b-mrl-v1	jev4-single	jev4-multi
Average	21.99	37.43	48.63	33.60	58.67	66.04	75.63
Arabic (ar)	19.62	38.40	45.85	28.40	63.06	71.41	81.81
Bengali (bn)	22.93	44.55	49.37	26.63	52.89	66.98	76.41
German (de)	12.74	39.58	52.87	40.36	62.99	70.21	80.86
English (en)	36.45	45.24	56.58	64.98	70.23	73.55	81.66
Spanish (es)	12.75	46.10	54.85	41.34	66.43	71.68	80.82
French (fr)	15.59	36.06	35.73	43.93	41.32	53.58	59.42
Hindi (hi)	16.73	36.94	48.42	18.02	50.94	62.64	71.77
Hungarian (hu)	25.38	33.88	44.42	12.67	52.35	65.86	76.00
Indonesian (id)	28.79	39.48	50.85	40.46	62.03	66.02	73.72
Italian (it)	19.63	37.98	49.77	34.76	60.05	63.96	73.68
Japanese (jp)	21.41	30.43	44.03	28.83	63.71	66.50	77.13
Korean (ko)	34.98	35.24	47.61	29.82	68.37	71.45	81.77
Burmese (my)	22.84	29.45	54.36	10.28	37.61	56.58	65.01
Dutch (nl)	14.90	39.89	50.40	52.29	65.09	68.58	78.94
Portuguese (pt)	23.32	45.85	54.28	51.30	67.53	69.04	78.85
Russian (ru)	16.82	38.95	49.34	31.88	64.44	68.86	80.70
Thai (th)	30.00	29.64	46.25	39.13	56.41	61.68	71.02
Urdu (ur)	13.64	32.73	36.52	9.45	38.76	49.76	62.17
Vietnamese (vi)	32.40	39.80	54.59	43.72	64.62	73.30	80.24
Chinese (zh)	18.82	28.41	46.45	23.82	64.51	69.23	80.58

Table A16: GitHub Readme Retrieval Benchmark Results

Language	bm25 + OCR	jev3 + OCR	j-clip-v2	colpali-v1.2	dse-qwen2-2b-mrl-v1	jev4-single	jev4-multi
Average	50.11	65.14	39.06	72.91	72.24	85.57	85.69
Arabic (ar)	27.49	27.98	31.02	55.19	55.95	75.02	75.26
Bengali (bn)	1.29	28.27	26.96	49.25	47.30	65.70	66.08
German (de)	60.11	84.58	45.46	84.15	80.62	91.09	91.35
English (en)	87.43	91.67	48.69	91.10	90.69	96.94	97.34
Spanish (es)	78.57	83.31	43.35	84.02	78.70	89.60	90.19
French (fr)	77.55	83.54	42.42	83.73	79.11	90.25	90.45
Hindi (hi)	2.72	48.08	28.55	51.22	46.49	69.31	70.98
Indonesian (id)	78.05	82.46	38.59	79.67	74.57	88.42	88.62
Italian (it)	78.83	86.54	44.26	85.31	80.81	91.76	91.41
Japanese (jp)	14.46	63.20	42.02	69.02	75.42	89.74	90.80
Korean (ko)	40.01	35.23	37.87	64.16	68.83	87.04	86.89
Dutch (nl)	76.52	86.36	43.25	84.10	82.85	92.83	91.37
Portuguese (pt)	80.33	84.46	43.88	85.00	80.09	91.43	91.47
Russian (ru)	39.78	50.86	37.04	78.16	78.92	89.51	88.61
Thai (th)	1.47	36.67	37.62	65.29	65.45	77.61	76.67
Vietnamese (vi)	66.70	79.67	37.14	70.05	68.20	86.90	86.94
Chinese (zh)	40.52	54.53	35.89	60.05	74.05	81.44	82.26

Table A17: Tweet Stock Retrieval Benchmark Results

Language	bm25 + OCR	jev3 + OCR	j-clip-v2	colpali-v1.2	dse-qwen2-2b-mrl-v1	jev4-single	jev4-multi
Average	22.30	42.77	55.36	76.36	62.76	78.10	85.34
Arabic (ar)	0.38	1.67	49.36	77.31	52.73	66.15	77.66
German (de)	48.27	66.86	52.49	73.53	57.35	79.38	85.63
English (en)	51.38	63.66	48.35	77.13	63.47	77.92	85.36
Spanish (es)	54.28	63.44	53.44	79.02	62.57	78.68	84.62
French (fr)	51.69	64.76	54.94	76.91	62.17	78.65	85.27
Hindi (hi)	0.08	0.08	88.55	93.39	97.00	97.46	96.50
Hungarian (hu)	15.55	62.31	52.30	71.06	58.17	80.09	85.01
Japanese (jp)	0.40	47.80	54.74	70.00	57.76	77.04	85.67
Russian (ru)	0.47	3.07	47.08	70.72	57.43	76.33	83.11
Chinese (zh)	0.45	54.04	52.30	74.54	58.94	69.33	84.55

Table A18: AirBnB Retrieval Benchmark Results

Language	bm25 + OCR	jev3 + OCR	j-clip-v2	colpali-v1.2	dse-qwen2-2b-mrl-v1	jev4-single	jev4-multi
Average	7.20	1.13	2.13	10.42	11.10	8.18	37.51
Arabic (ar)	1.10	0.40	0.47	3.06	3.64	2.20	6.20
German (de)	4.03	0.71	5.54	20.17	15.09	9.27	41.94
English (en)	48.39	1.70	4.83	23.26	12.94	13.33	64.17
Spanish (es)	6.25	0.18	2.10	18.06	8.61	9.11	39.84
French (fr)	3.86	2.00	2.05	10.86	11.87	8.70	30.55
Hindi (hi)	0.16	0.86	0.82	3.19	4.93	4.05	17.44
Hungarian (hu)	5.58	0.69	3.01	7.34	11.10	6.69	27.30
Japanese (jp)	0.36	1.53	0.54	3.44	14.91	7.63	45.65
Russian (ru)	1.67	1.39	0.88	13.16	13.61	8.66	40.80
Chinese (zh)	0.58	1.84	1.04	1.62	14.28	12.14	61.19

A.5 Data Preparation Prompts

```
You are an assistant specialized in Multimodal RAG tasks.

The task is the following: given an image from a pdf page, you will have to generate questions that
    can be asked by a user to retrieve information from a large documentary corpus.

The question should be relevant to the page, and should not be too specific or too general. The
    question should be about the subject of the page, and the answer needs to be found in the page.

Remember that the question is asked by a user to get some information from a large documentary
    corpus that contains multimodal data. Generate a question that could be asked by a user without
     knowing the existence and the content of the corpus.

Generate as well the answer to the question, which should be found in the page. And the format of
    the answer should be a list of words answering the question.

Generate at most THREE pairs of questions and answers per page in a dictionary with the following
    format, answer ONLY this dictionary NOTHING ELSE:

{
   "questions": [
      {
         "question": "XXXXXX",
         "answer": ["YYYYYY"]
      },
      {
         "question": "XXXXXX",
         "answer": ["YYYYYY"]
      },
      {
         "question": "XXXXXX",
         "answer": ["YYYYYY"]
      }
   ]
}
where XXXXXX is the question and ['YYYYYY'] is the corresponding list of answers that could be as
    long as needed.

Note: If there are no questions to ask about the page, return an empty list. Focus on making
    relevant questions concerning the page.

Here is the page:
<file source="{{ (path + '/' if path else '') + image }}"/>
```

We use this prompt to generate questions for document images that do not have related text values that can be used to construct text-document pairs. This prompt follows the same formulation as the one introduced in Faysse et al. (2025)

Figure 3: Prompt for generating questions for visually-rich documents

```
Your task is to categorize each search query into one of the following two classes: VALID or INVALID
    .

Criteria for VALID queries:
1. VALID queries should not be vague or ambiguous, they must provide enough context for search
    outside a specific set of documents.
2. VALID queries should not depend on specific documents, charts, tables, but can mention known
    entities (like individuals, institutions, etc.).

Queries that do not meet the given criteria should be classified as INVALID.

Format for response:
Query: "..."
Class: VALID/INVALID
Explanation: "..."

Examples for reference:

Query: "How are concerns logged and tracked throughout the process?"
Class: INVALID
Explanation: This query does not contain enough information, it is not clear what "process" is
    being referenced.

Query: "For a married couple filing jointly, what is the withholding amount according to the Tax
    Withholding table?"
Class: INVALID
Explanation: This query depends on a specific "Tax Withholding" table.

Query: "What is the role of Gnther Oberhofer at Conrad Electronic?"
Class: VALID
Explanation: This query provides enough context by asking about a specific person at a known
    company.

Query: "Under what circumstances might the store send emails to customers?"
Class: INVALID
Explanation: This query is too vague because it does not specify which store is being referred to.

Query: "What is the premise of the story in Star Divide Ascension Series Book 2?"
CLASS: VALID
Explanation: This query provides enough context for a search by specifying the title of a
    particular book within a series.

Query: "What action will be taken regarding the trading of BROKEN HILL PROSPECTING LIMITED's
    securities?"
Class: INVALID
Explanation: The query lacks context such as timeframe, specific events, or responsible entities,
    making it vague.

Query: "What is the purpose of Tallan's Accessible Web Portal?"
Class: VALID
Explanation: This query inquires the purpose of a well known portal.

Query: "What are some examples of how pupils at Doncaster School for the Deaf are involved in
    enrichment opportunities?"
Class: VALID
Explanation: This query provides enough context for a search as it specifies a particular school (
    Doncaster School for the Deaf).

Using the guidelines and format provided above, categorize the following query: "{{ query }}".
```

We use this prompt to filter out underspecified or document-dependent questions. It ensures that only contextually self-contained queries—those not assuming prior knowledge of a specific document—are retained. This filtering is necessary in datasets with synthetic questions, where question–document relevance is annotated based on the generation source only.

Figure 4: Prompt for filtering questions