RENDERRANK: LEARNING TO RERANK TEXT WITH COMPRESSED VISUAL TOKENS

Anonymous authors
Paper under double-blind review

ABSTRACT

Rendering document text as images allows vision-language models to encode documents as visual tokens, which can reduce input sequence length compared with text input. This reduction in input length is particularly useful for reranking, where each query involves scoring multiple candidate documents and token savings apply to each candidate evaluation. We introduce RenderRank, a reranker that learns query-dependent relevance scoring from compressed visual document representations instead of the text token sequences used by conventional text-based rerankers. Training first aligns relevance scores from visual inputs with those of a text-based teacher, then refines the relative scores of positive and negative documents for the same query. Across 11 datasets from BEIR, RenderRank uses 16.5–35.5% fewer input tokens while achieving an average NDCG@10 of 55.96, outperforming all evaluated text-based baselines below 4B parameters and some larger models. Across four long-document datasets, it achieves an average NDCG@10 of 88.27 with approximately half the average input token count of the evaluated text-based rerankers. In this setting, RenderRank delivers $1.70\times$ the highest average throughput of the evaluated baselines. These results demonstrate that compressed visual representations can support accurate document relevance scoring, providing an alternative to text token representations for reranking.

1 INTRODUCTION

Retrieval-augmented generation (RAG) generates answers to queries using information retrieved from external documents as supporting evidence (Lewis et al., 2020; Izacard & Grave, 2021; Izacard et al., 2023). In this pipeline, first-stage retrieval efficiently identifies potentially relevant candidate documents from a large corpus (Karpukhin et al., 2020; Xiong et al., 2020; Khattab & Zaharia, 2020). However, retrieved candidates vary in their relevance to the query and their usefulness as evidence for answering it, so a more precise assessment is needed to determine which documents to use for subsequent generation (Yu et al., 2024; Asai et al., 2024). A reranker prioritizes candidate documents for generation based on their relevance to the query (Nogueira & Cho, 2019; Glass et al., 2022). Reranking computes a relevance score for each retrieved candidate, with the query and the corresponding document provided together as input (Nogueira et al., 2020; Zhang et al., 2025).

When reranking multiple candidates, the input length of each document affects the computational cost of evaluating the entire candidate set. Longer token sequences require more computation for relevance scoring, and these costs accumulate across retrieved candidates (Peng et al., 2025). Limiting the input length can reduce reranking computation, but may exclude evidence or context needed to assess relevance to the query (Li et al., 2023). This highlights the need to represent document content with fewer tokens while preserving the information needed for relevance assessment. Such an approach could enable relevance scoring for the same document with fewer tokens, or allow more document content to be considered within the same sequence length limit.

From this perspective, rendering document text as images and encoding them into visual tokens offers a potential way to shorten document representations. Instead of feeding the source text directly as text tokens, this approach uses the visual encoder of a vision-language model (VLM) to convert the rendered images into visual tokens for the language backbone (Wang et al., 2024). In generation tasks, such visual representations have been used to perform question answering and summarization with fewer input tokens (Li et al., 2025c). Visual text compression has further enabled models to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

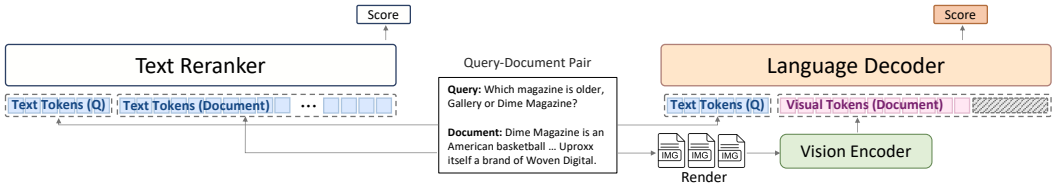


Figure 1: Overview of conventional text-based reranking (left) and RenderRank (right) for the same query–document pair. RenderRank encodes rendered document images into visual tokens, which the language decoder processes with the textual query to compute a relevance score.

process more source content within a limited context window and reduce the time spent on input processing and answer generation (Cheng et al., 2026). These results show that models can understand and use document content for downstream tasks even when it is represented as a shorter sequence of visual tokens.

Extending the benefits of visual text compression to reranking requires the ability to assess relevance to a query from compressed document content. Beyond changing the document input modality from text to images, the rendered images must enable the model to understand document content. Rendering configurations affect both the length of the visual token sequence and how readily the model can interpret the content. Shorter sequences can reduce attention and token-wise computation in the language model, accelerating relevance scoring even when each candidate is scored in a single forward pass. These configurations must therefore account for both computational efficiency and document understanding. Alongside these configurations, training must adapt relevance scoring to visual inputs by teaching the model to relate these visual representations to the query.

We introduce RenderRank, a reranker that renders document text as images, encodes them into compressed visual tokens, and scores document relevance to a textual query. Figure 1 provides an overview of RenderRank and illustrates how its document representation and relevance scoring pipeline differ from those of a text-based reranker. We employ a two-stage training approach to learn relevance scoring from visual document representations and to assign higher scores to positive documents than to negatives. First, Cross-Modal Relevance Distillation trains the model on textual queries and rendered document images to predict relevance scores that approximate those assigned by a teacher to the corresponding textual inputs. Next, Query-Local Relevance Discrimination refines relevance scoring by learning the relative scores of positive and negative documents associated with the same query. Document images are encoded in advance, independently of the query, and the resulting visual representations are used to predict relevance scores conditioned on the textual query.

RenderRank is evaluated on 11 datasets from BEIR (Thakur et al., 2021) and four long-document datasets in terms of ranking quality, input token counts, estimated computational cost, and measured throughput. RenderRank achieves an average NDCG@10 of 55.96 on BEIR, outperforming all other evaluated models with fewer than 4B parameters and some larger models. It uses 16.5–35.5% fewer input tokens than the text-based rerankers evaluated. RenderRank also achieves higher inference throughput than models of similar size and some smaller models, showing that its efficiency benefits extend beyond input token savings. For long-document reranking, it achieves high reranking performance with approximately half the average input token count of the text-based rerankers evaluated and outperforms all compared models at every tested maximum sequence length under identical length limits. These results demonstrate that representing document text as visual tokens supports high reranking effectiveness and inference efficiency while enabling more document content to be used within a limited input length.

2 Related Work

Visual Text Representation and Compression Rendering text as images allows document content to be represented through visual tokens rather than text tokens (Rust et al., 2022; Lyu et al., 2025; Tschannen et al., 2023; Xiao et al., 2024). Beyond understanding existing document images, this approach considers how source text should be visually structured for model input (Lotz et al., 2023). Prior work explores both the feasibility of understanding text through visual representations

and the potential to reduce inference cost and latency by using fewer tokens to represent documents (Li et al., 2025c; Cheng et al., 2026). Visual information preservation has also been examined in relation to rendering choices such as font size, line spacing, and page layout (Tang et al., 2026). Rendering density affects both the preservation of fine-grained textual information and the amount of source text represented by each visual token, allowing fewer tokens to encode a document or more content to fit within the same sequence length limit. Complementary approaches diversify rendering during training to reduce reliance on superficial visual cues (Yuan et al., 2026).

Document Reranking Reranking reorders candidate documents returned by a retriever according to their relevance to a query. Cross-encoder rerankers score relevance by modeling interactions between the query and document (Nogueira & Cho, 2019; Nogueira et al., 2020). Large language models have been adopted for reranking to leverage their language understanding and reasoning capabilities for improved effectiveness (Sun et al., 2023; Ma et al., 2024; Zhang et al., 2025). Building on their reranking performance, knowledge distillation approaches use these models as teachers, training other rerankers on the relative ordering of candidate documents (Baldelli et al., 2024; Schlatt et al., 2025). Distillation objectives also include directly matching the teacher's relevance score differences between documents (Hofstätter et al., 2020). As inference costs increase with model size, document length, and the number of candidates evaluated per query, reranking effectiveness is also studied alongside computational cost and throughput (Peng et al., 2025; Aarsen, 2026). In image–text reranking, precomputing visual features and compressing visual tokens have been explored to reduce the computational cost of scoring candidate pairs (Taraday et al., 2026). Efficient reranking of document page images has also been studied through pretraining on rendered text followed by further training on document images, using query-dependent visual token selection and a listwise approach that ranks the candidate set in a single forward pass (Sun et al., 2026).

RenderRank focuses on reducing the input sequence length required for reranking by encoding candidate documents originally provided as text into compressed visual tokens. To this end, it learns to approximate relevance scores from a text-based teacher using visual inputs, then refines the relative scores of candidates associated with the same query. By representing document content with fewer tokens, this approach allows longer documents to fit within the same sequence length limit for relevance scoring.

3 RENDERRANK

RenderRank renders document text as images and encodes them into visual tokens to score relevance to a textual query. To this end, we render documents with consideration for text legibility and information density, and train the model to assess relevance between visual document representations and textual queries. This section describes the document rendering procedure and the rationale for its configuration, followed by the reranking training method based on these visual representations.

3.1 RENDERING TEXT FOR VISUAL PROCESSING

Rendering document text as images requires representing sufficient content within a limited image size while preserving the character shapes and layout needed for the model to interpret the text. Smaller fonts and tighter line spacing can increase information density and reduce the number of visual tokens, but fewer pixels per character and less separation between lines may make the content harder to recognize. Rendering configurations should therefore account for both token reduction and document understanding. We use question answering and summarization to evaluate how well models understand and use rendered document content. Question answering evaluates the ability to identify and use evidence relevant to a query, whereas summarization evaluates the ability to identify and synthesize the main content of a document. We therefore use performance on these two tasks and token efficiency to determine the rendering configuration for reranking, while considering the potential reuse of rendered documents as

Table 1: Rendering configurations evaluated by F1 on QASPER and ROUGE-L on GovReport. Blue and red indicate fewer and more tokens than text input, respectively.

Font	LS	Gemma-4-26B-A4B-it		Qwen3.6-35B-A3B	
		QASPER	GovReport	QASPER	GovReport
Text	–	47.73	28.78	51.45	31.10
10pt	1.0	45.90	28.27	41.32	30.55
	1.1	45.72	28.46	43.18	30.75
	1.2	44.58	28.27	45.80	30.42
11pt	1.0	47.48	28.53	45.26	30.53
	1.1	48.32	28.69	46.46	30.55
	1.2	45.76	28.68	47.41	30.85
12pt	1.0	48.07	28.67	48.72	30.82
	1.1	47.93	28.75	45.35	30.66
	1.2	46.50	28.60	45.33	30.69

inputs to a downstream generation model. Specifically, we adopt Roboto Regular as used by Tang et al. (2026) and vary font size and line spacing to compare generation performance and input token counts.

Table 1 compares question answering and summarization performance and token reduction rates across font sizes and line spacings. With line spacing fixed at 1.0, 12pt yields less token reduction than 10pt but higher performance on both tasks for both models. For example, the QASPER F1 score of Qwen3.6 increases from 41.32 to 48.72. In contrast, increasing line spacing at 12pt increases the number of visual tokens without consistent performance gains. These results suggest that an appropriate rendering configuration can reduce token usage while largely preserving task performance.

Rendering Configuration Based on the performance and token efficiency observed in the preceding experiments, we render document text for training RenderRank in Roboto Regular at 12pt with line spacing of 1.0. Image width and height are set to multiples of 32 pixels to accommodate the 16×16-pixel patches and 2×2 spatial merging of the Qwen3-VL (Li et al., 2026) visual encoder used in RenderRank. Documents are rendered at 96 DPI with a fixed width of 896 pixels and a maximum height of 896 pixels, with the height adjusted in 32-pixel increments according to the content. Text is rendered in black on a white background without margins and wraps to the next line when it would exceed the image width. Content exceeding one page continues on the next image without overlap. Long documents are rendered into multiple images, which preserve the order of the document content and are processed as a single candidate document.

3.2 Learning to Rerank Text through Visual Representations

RenderRank is trained to assess document relevance by relating rendered document images to a textual query. Let q denote the query, d the original document, and $R(d)$ the sequence of rendered document images. Parameterized by θ, RenderRank takes the textual query and the document's visual tokens as input and produces a relevance score $s_\theta(q, R(d))$. Training proceeds in two stages: (1) Cross-Modal Relevance Distillation, which trains the model to predict a text-based teacher's relevance scores from visual inputs, and (2) Query-Local Relevance Discrimination, which refines relevance scoring to prioritize the positive document within the candidate set for each query.

Cross-Modal Relevance Distillation Using visual tokens to represent document text for reranking requires the ability to relate the content of document images to a query and assess its relevance. However, changing the input representation alone does not ensure that a text-based reranker's relevance scoring capabilities are preserved with visual inputs. We therefore perform cross-modal relevance distillation by pairing the text and image representations of the same document. The student learns to approximate a text-based teacher's relevance scores from the textual query and rendered document images.

Let $t(q, d)$ denote the relevance score produced by the teacher from the textual query q and document d, and $s_\theta(q, R(d))$ the score produced by the student from the same query and rendered document images $R(d)$. We perform cross-modal relevance distillation by minimizing the mean squared error (MSE) between these scores over the set of training pairs $\mathcal{D}$:

$$\mathcal{L}_{\text{CMRD}} = \mathbb{E}_{(q,d)\sim\mathcal{D}} \left[\big(s_\theta(q, R(d)) - t(q, d) \big)^2 \right] \tag{1}$$

This supervision encourages cross-modal alignment at the level of relevance scores without explicitly aligning text and image features in a shared representation space. Conditioned on the same query, the student learns to assign the rendered document a score consistent with the teacher's score for its textual counterpart. By operating at the score level, this objective supports distillation between textual and visual sequences of different lengths without requiring token-level correspondence.

Query-Local Relevance Discrimination Building on the cross-modal score alignment learned in the preceding stage, we train the model to assign higher scores to the positive document than to negatives within each query's candidate set, refining visual-input relevance prediction for reranking. For each query q_i, we construct a candidate set $\mathcal{C}_i = \{d_i^+, d_{i,1}^-, \ldots, d_{i,K}^-\}$, where d_i^+ is the positive document and $\{d_{i,j}^-\}_{j=1}^K$ are the K negative documents for the same query. We use an InfoNCE

contrastive loss restricted to each query's candidate set, with the objective defined as follows:

$$\mathcal{L}_{\mathrm{QLRD}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\big(s_\theta(q_i, R(d_i^+))\big)}{\exp\big(s_\theta(q_i, R(d_i^+))\big) + \sum_{j=1}^{K}\exp\big(s_\theta(q_i, R(d_{i,j}^-))\big)} \tag{2}$$

Here, N denotes the number of queries in a mini-batch. This objective trains the model to assess the relevance of rendered document content through contrasts between positive and negative documents associated with the same query. Although the loss compares candidates within each query, the model computes each relevance score using only the query and the corresponding document.

4 Experimental Setup

Training RenderRank is initialized from Qwen3-VL-Reranker-2B (Li et al., 2026) and trained in two stages. For cross-modal relevance distillation, we use the English fine-tuning data released by Sourty et al. (2026), comprising 1.57M training records with one positive document and ten negatives per query. We retain all records and select one positive and three negatives per query, yielding 6.28M query–document pairs. We use Qwen3-Reranker-4B (Zhang et al., 2025) as the teacher model to provide relevance scores for the original text query–document pairs.

For query-local relevance discrimination, we use RLHN-100K (Thakur et al., 2025), with the positive and negative documents associated with each query. Throughout both stages, we freeze the vision encoder and visual feature mergers and apply LoRA (Hu et al., 2021) only to the text decoder. Following the merge-and-reinitialize principle of ReLoRA (Lialin et al., 2023), we merge the first-stage LoRA updates into the backbone and initialize fresh adapters for the second stage. Detailed data preparation procedures and training hyperparameters are provided in the Appendix C.

Evaluation We evaluate RenderRank on 11 BEIR datasets (Thakur et al., 2021): ArguAna (Wachsmuth et al., 2018), Climate-FEVER (Diggelmann et al., 2020), DBPedia (Hasibi et al., 2017), FiQA (Maia et al., 2018), FEVER (Thorne et al., 2018), HotpotQA (Yang et al., 2018), NFCorpus (Boteva et al., 2016), SCIDOCS (Cohan et al., 2020), SciFact (Wadden et al., 2020), TREC-COVID (Voorhees et al., 2021), and Touché-2020 (Bondarenko et al., 2022). For each BEIR query, we rerank the top 100 documents retrieved by BM25. To assess long-document reranking performance, we evaluate on the English subset of MLDR (Chen et al., 2024) following the MMTEB (Enevoldsen et al., 2025), and on three LongEmbed datasets (Zhu et al., 2024): 2WikiMQA, QMSum, and SummScreenFD. For the latter, we retrieve the top eight documents per query using Qwen3-Embedding-0.6B (Zhang et al., 2025) and rerank them, consistent with the eight-candidate evaluation protocol used for MLDR. We use nDCG@10 as the evaluation metric. All baseline rerankers use text queries and documents. For RenderRank, queries remain in text form, while documents are rendered as images following Section 3. Consistent with prior work on visual feature precomputation (Taraday et al., 2026; Fan et al., 2026), We assume that visual embeddings of document images have been computed in advance.

Baselines We compare RenderRank with rerankers spanning diverse architectures and model sizes: gte-reranker-modernbert-base (Zhang et al., 2024), mxbai-rerank-large-v1 and v2 (Shakir et al., 2024; Li et al., 2025b), bge-reranker-large and bge-reranker-v2-gemma (Xiao et al., 2023; Chen et al., 2024), Qwen3-Reranker-0.6B and 4B (Zhang et al., 2025), LAMAR-600m (Hong et al., 2026), llama-nemotron-rerank-1b-v2, LightOn-rerank-PW-2B and 4B (Ananya & Chatelain, 2026), and zerank-2-reranker (Pipitone et al., 2025).

5 Experimental Results

5.1 Reranking Performance

Table 2 compares the reranking performance and average input token counts of RenderRank and a range of text-based rerankers across 11 BEIR benchmarks. RenderRank achieves an average NDCG@10 of 55.96, demonstrating competitive performance against text-based rerankers. For example, it outperforms zerank-2-reranker and LightOn-rerank-PW-4B by 7.66% and 7.26% in average NDCG@10, respectively. This performance is obtained with an average of 290.07 input tokens

Table 2: Reranking performance (NDCG@10) and input token counts on 11 BEIR datasets. Avg. and Tokens report the mean performance and input length per query–document pair, respectively, across datasets. Params counts all model parameters. For RenderRank, input length includes both textual and visual tokens processed by the language backbone. Bold and underlined values indicate the top two results across rerankers, respectively.

		BEIR												Tokens ↓
Model	**Params**	**AA**	**CFV**	**DBP**	**FQA**	**FVR**	**HQA**	**NFC**	**SD**	**SF**	**TC**	**TCH**	**Avg.**	
BM25	–	39.70	16.51	31.83	23.61	65.18	63.30	32.18	14.90	67.89	59.47	44.22	41.71	–
Rerankers with $< 2B$ parameters														
gte-reranker-modernbert-base	150M	66.14	25.80	42.10	42.54	**89.84**	76.15	34.81	18.90	75.81	79.66	33.44	53.20	359.25
mxbai-rerank-large-v1	435M	16.16	25.61	45.37	40.29	82.19	71.58	37.32	18.96	75.26	**85.84**	37.44	48.73	<u>347.45</u>
bge-reranker-large	560M	30.94	34.11	44.21	37.47	89.30	80.09	33.91	16.68	73.98	73.21	34.69	49.87	409.32
LAMAR-600m	560M	66.21	<u>36.21</u>	<u>46.23</u>	41.14	<u>89.49</u>	79.14	35.04	20.22	76.97	80.65	35.76	55.19	409.32
Qwen3-Reranker-0.6B	600M	68.04	34.34	43.76	39.74	87.13	77.25	36.60	20.44	77.00	<u>85.54</u>	30.32	54.56	438.70
llama-nemotron-rerank-1b-v2	1.2B	54.29	27.37	45.13	**47.11**	87.66	<u>80.34</u>	<u>38.14</u>	<u>21.53</u>	**79.82**	83.14	32.47	54.27	363.99
mxbai-rerank-large-v2	1.5B	42.23	23.03	38.01	28.85	70.84	69.50	35.17	17.49	78.81	67.78	**46.94**	47.15	449.76
Rerankers with $\geq 2B$ parameters														
LightOn-rerank-PW-2B	2.2B	45.52	23.21	42.60	35.66	86.51	73.05	35.61	16.64	77.70	81.24	36.91	50.42	416.22
bge-reranker-v2-gemma	2.5B	<u>74.98</u>	32.03	45.01	43.18	87.45	**80.47**	36.25	19.58	77.43	80.08	37.56	55.82	399.11
Qwen3-Reranker-4B	4B	**75.40**	**37.84**	**47.02**	44.63	89.14	79.06	37.48	**23.59**	<u>79.47</u>	85.47	<u>38.77</u>	**57.99**	438.70
zerank-2-reranker	4B	44.87	23.65	44.95	42.75	83.22	70.97	**38.50**	20.23	79.35	85.24	38.10	51.98	380.69
LightOn-rerank-PW-4B	4.5B	52.50	28.24	44.80	40.27	87.70	72.79	37.33	17.29	76.72	79.54	36.70	52.17	416.22
RenderRank (Ours)	2.1B	69.11	35.91	46.19	42.66	88.92	78.60	37.31	20.75	77.00	84.82	34.27	<u>55.96</u>	**290.07**

per query–document pair, 16.5–35.5% fewer than the text-based rerankers evaluated. These results show that representing document text as visual tokens reduces the number of tokens processed by the language backbone while effectively scoring document relevance to the query. Such token savings are particularly relevant to reranking, where each candidate document is scored separately with the query, so the reduction in input tokens applies to every candidate evaluation.

5.2 Efficiency Evaluation

To evaluate the extent to which fewer input tokens reduce computational cost and improve reranking throughput, we analyze both estimated computation and measured inference throughput. Shorter inputs can reduce the computational cost of scoring each candidate, but this does not necessarily lead to proportional gains in throughput. We therefore examine these two aspects separately. Specifically, we (1) compare computational costs based on model architecture and input length, and (2) measure the number of pairs processed per second to assess RenderRank's inference efficiency.

5.2.1 Computational Efficiency

Setup. We estimate computational cost using the approximation in E^{2}R-FLOPs (Peng et al., 2025), accounting for model architecture and individual input lengths. The computational cost of a single forward pass through a standard dense Transformer is approximated as follows:

$$F(n) = 4Lh\left[\left(1+\frac{1}{r}\right)h + f\right]n + \frac{4Lhn^2}{r}, \tag{3}$$

L: layers, h: hidden dim, f: FFN dim, n: input length, $r{=}H_Q/H_{KV}$: Q/KV head ratio

We adjust this expression according to each model's architecture and attention pattern. Let $n_{q,d}$ denote the actual number of input tokens for query q and document d after truncation to the maximum sequence length, and compute the corresponding cost as $C_{q,d} = F(n_{q,d})$. We define C_q as the sum of these costs across all candidates for a query. Here, $\mathrm{AVG}_{C_{q,d}}$ denotes the average cost across all evaluated pairs, whereas AVG_{C_q} denotes the average total cost per query. TFLOPs and QPP (Queries per PetaFLOP) are calculated as follows:

$$\mathrm{TFLOPs/pair} = \frac{\mathrm{AVG}_{C_{q,d}}}{10^{12}} \qquad \mathrm{QPP} = \frac{1}{\mathrm{AVG}_{C_q}/10^{15}} \tag{4}$$

TFLOPs measures the average computation required to score the relevance of an individual pair, with lower values indicating lower per-pair computational cost. QPP measures the number of queries for

Table 3: Reranker architectures and estimated computational efficiency, with input token counts and efficiency metrics averaged over 11 BEIR datasets. Q, KV, and d_h denote the numbers of query and key–value heads and the head dimension, respectively. G, L, SW, and Lin denote global, local, sliding-window, and linear attention. For RenderRank, Enc. and Dec. denote the vision encoder and language decoder. Its parameter count includes all model components, whereas TFLOPs and QPP exclude offline vision encoding. Bold and underlined values indicate the top two results within each parameter group, respectively. *GQA applies only to the global-attention layers.

	Architecture										Efficiency		
Model	**Params**	**Layers**	**Hidden**	**FFN**	**Q**	**KV**	d_h	**Bi.**	**Attn.**	**Attn. Pattern**	**Tokens ↓**	**TFLOPs ↓**	**QPP ↑**
Rerankers with $<$ 2B parameters													
gte-reranker-modernbert-base	150M	22	768	1152	12	12	64	O	MHA	G8+L14	<u>359.25</u>	**0.07**	**209.5**
mxbai-rerank-large-v1	435M	24	1024	4096	16	16	64	O	MHA	G24	**347.45**	<u>0.18</u>	<u>73.4</u>
bge-reranker-large	560M	24	1024	4096	16	16	64	O	MHA	G24	409.32	0.21	63.3
LAMAR-600m	560M	24	1024	4096	16	16	64	O	MHA	G24	409.32	0.28	56.8
Qwen3-Reranker-0.6B	600M	28	1024	3072	16	8	128	X	GQA	G28	438.70	0.25	53.2
llama-nemotron-rerank-1b-v2	1.2B	16	2048	8192	32	8	64	O	GQA	G16	363.99	0.52	28.2
mxbai-rerank-large-v2	1.5B	28	1536	8960	12	2	128	X	GQA	G28	449.76	0.84	15.2
Rerankers with $\geq$ 2B parameters													
LightOn-rerank-PW-2B	2.2B	24	2048	6144	8	2	256	X	GQA*	G6+Lin18	416.22	<u>0.73</u>	<u>18.3</u>
bge-reranker-v2-gemma	2.5B	18	2048	16384	8	1	256	X	MQA	G18	399.11	1.10	12.3
Qwen3-Reranker-4B	4B	36	2560	9728	32	8	128	X	GQA	G36	438.70	2.12	6.1
zerank-2-reranker	4B	36	2560	9728	32	8	128	X	GQA	G36	<u>380.69</u>	1.84	7.8
LightOn-rerank-PW-4B	4.5B	32	2560	9216	16	4	256	X	GQA*	G8+Lin24	416.22	1.73	7.7
RenderRank (Ours)	2.1B	24 (Enc.) 28 (Dec.)	1024 2048	4096 6144	16 16	16 8	64 128	O X	MHA GQA	G24 G28	**290.07**	**0.63**	**20.1**

which all candidate documents can be reranked within a fixed computational budget, representing processing capacity per unit of computation rather than processing speed over time.

Results. Table 3 compares model architectures and estimated computational costs. RenderRank requires an average of 0.63 TFLOPs per query–document pair and achieves an average QPP of 20.1. For example, compared with bge-reranker-v2-gemma, which requires 1.10 TFLOPs per pair and achieves a QPP of 12.3, RenderRank offers comparable reranking performance at a lower computational cost. Its higher QPP indicates that more queries can have their candidate sets reranked within the same computational budget. For a given model, shorter sequences reduce the computational cost of attention as well as the projections and FFNs applied to each token. Document compression through visual tokens can therefore lower the computational cost per candidate even in reranking, where relevance is scored in a single forward pass. However, differences in computational cost across models also reflect backbone size and attention architecture, which should be considered when assessing the contribution of reduced input length.

5.2.2 Inference Throughput

Setup. To evaluate reranking throughput in practice, we measure Pairs per Second (PPS) following the measurement procedure of (Aarsen, 2026). Experiments are conducted on an NVIDIA A6000 48GB using the maximum sequence length supported by each model. Starting at 8, we double the batch size until an out-of-memory (OOM) error occurs or the search limit is reached, testing smaller batch sizes when necessary. Let $\mathcal{B}$ denote the set of successfully measured batch sizes, N the total number of evaluated pairs, and T_b the total GPU forward time at batch size b. We report the highest PPS across successfully measured batch sizes:

$$\mathrm{PPS} = \max_{b \in \mathcal{B}} \frac{N}{T_b} \tag{5}$$

T_b is measured in seconds using CUDA events after warm-up at each batch size and includes only GPU forward time, excluding preprocessing and cache I/O.

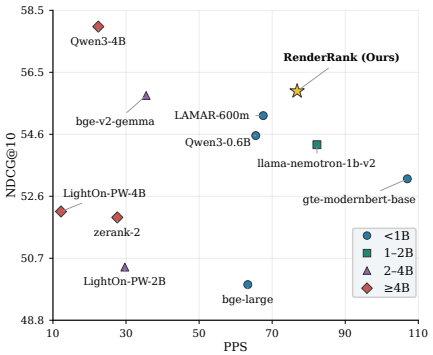


Figure 2: Reranking effectiveness and GPU forward throughput, averaged over 11 BEIR benchmarks. For each model and benchmark, PPS is measured at the batch size yielding the highest throughput.

Table 4: Reranking performance, average input token counts, and throughput on four long-document datasets. The comparison includes models officially supporting sequence lengths $\geq$ 16K tokens.

Model	MLDR			LongEmbed									Avg.	
				2WikiMQA			QMSum			SummScreenFD				
	N@10	Tokens	PPS	N@10	Tokens	PPS	N@10	Tokens	PPS	N@10	Tokens	PPS	N@10	PPS
Qwen3-Reranker-0.6B	99.63	8863.4	2.59	**94.54**	9230.2	2.23	56.43	13543.6	1.60	98.25	8606.3	4.21	87.21	2.66
LightOn-rerank-PW-2B	98.91	8878.0	1.10	71.27	9230.1	0.90	54.30	13876.6	0.57	93.05	8977.2	1.21	79.38	0.94
Qwen3-Reranker-4B	**99.85**	8863.4	0.87	**94.54**	9230.2	0.76	59.11	13543.6	0.45	**99.11**	8606.3	1.13	88.15	0.80
zerank-2-reranker	99.57	8805.6	0.88	94.42	9173.1	0.61	**60.10**	13485.9	0.34	98.97	8548.5	1.12	88.26	0.74
LightOn-rerank-PW-4B	99.68	8878.0	0.53	93.84	9230.1	0.44	58.87	13876.6	0.26	98.91	8977.2	0.52	87.82	0.44
RenderRank (Ours)	99.74	**4198.0**	**4.59**	94.30	**4262.2**	**4.25**	59.94	**6388.8**	**2.67**	99.10	**3701.9**	**6.52**	**88.27**	**4.51**

Results. Figure 2 compares average NDCG@10 and PPS across models. RenderRank achieves an average throughput of 76.83 PPS, exceeding that of larger models and several similarly sized models. For example, it achieves approximately twice the throughput of bge-reranker-v2-gemma at comparable reranking performance. The throughput advantage also extends to some smaller models. Despite its higher estimated TFLOPs per pair, RenderRank exceeds the throughput of 67.55 PPS measured for LAMAR-600m. This result indicates that lower estimated TFLOPs per pair do not necessarily correspond to higher inference throughput. RenderRank supports higher candidate evaluation throughput while maintaining high ranking performance.

5.3 Long Document Reranking

Table 4 compares reranking performance, average input token counts, and throughput (PPS) on four long-document datasets. RenderRank achieves an average NDCG@10 of 88.27, demonstrating strong reranking performance in prioritizing relevant documents even for long inputs. On MLDR, it achieves an NDCG@10 of 99.74 and uses an average of 4.20K input tokens, approximately half that of the text-based rerankers evaluated. This reduction extends to the three LongEmbed datasets, where RenderRank uses 53–57% fewer input tokens than the most token-efficient text baseline. RenderRank achieves the highest throughput among the compared models on all four datasets, averaging 4.51 PPS versus 2.66 PPS for the fastest baseline, a $1.70\times$ improvement. The corresponding speedups range from $1.55\times$ to $1.91\times$ across individual datasets. For long inputs, attention computation grows with sequence length, alongside the cost of token-wise projections and FFNs; shorter input representations can therefore reduce the computational cost even when scoring an individual pair. These results show that RenderRank provides high ranking quality for long documents while reducing input token counts and improving inference speed.

6 Analysis and Ablation Studies

6.1 Reranking under Fixed Token Budgets

To evaluate whether visual document representations can preserve more document content within the same length limit and improve reranking performance, we evaluate all models on the same MLDR samples, setting the maximum sequence length to 2K, 4K, and 8K in separate evaluations. Table 5 shows that RenderRank achieves the highest NDCG@10 across all three settings. At 2K, RenderRank achieves an NDCG@10 of 97.9, surpassing the scores of 95.8 and 97.3 obtained by gte-reranker and LightOn-PW-4B at 4K, respectively. At 4K, it reaches 99.5, comparable to the performance of text-based rerankers at 8K, providing strong long-document reranking performance with half the maximum sequence length. Performance differences between models narrow at 8K, whereas differences across document representations are more pronounced at shorter lengths. These results suggest that RenderRank can encode and leverage more document content within the same length

Table 5: Reranking performance on MLDR under different maximum sequence lengths. Bold indicates the best result at each length.

Model	2K	4K	8K
gte-reranker-modernbert	93.2	95.8	98.5
llama-nemotron-rerank-1b-v2	97.4	98.5	99.4
Qwen3-Reranker-4B	96.9	98.0	99.6
LightOn-rerank-PW-4B	95.1	97.3	99.5
RenderRank	**97.9**	**99.5**	**99.7**

limit for long-document reranking, providing ranking quality comparable to text-based rerankers with fewer input tokens.

6.2 Effect of Rendering Font Size

We vary the rendering font size to examine how the visual information density of documents affects reranking quality and inference efficiency. As shown in Figure 3, increasing the font size from 10pt to 14pt raises the average input length from approximately 236 to 370 tokens and reduces throughput from approximately 94 to 63 PPS, while improving NDCG@10 from 55.04 to 56.43. Smaller fonts fit more content within the same image area, shortening the input sequence, but allocate fewer pixels to each character, potentially making details needed for relevance assessment harder to distinguish. The default 12pt setting achieves a score of 55.96 with approximately 290 input tokens and 77 PPS, offering fewer tokens and higher throughput than 14pt with a relative difference of approximately 0.83%. These results show that document compression through visual tokens can improve throughput while allowing the balance between ranking quality and inference efficiency to be adjusted through the degree of compression.

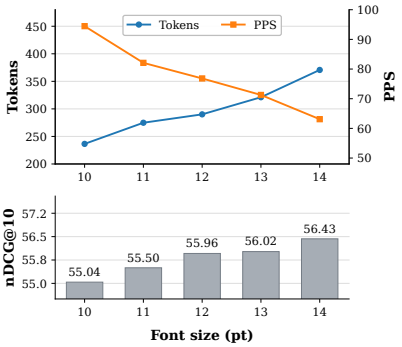


Figure 3: Effect of rendering font size on reranking performance, input token counts, and throughput, averaged over 11 BEIR datasets.

6.3 Component Ablation

We examine how the training stages and inference settings affect reranking performance and throughput, respectively. As shown in Table 6(a), Cross-Modal Relevance Distillation improves average NDCG@10 with image inputs from 50.20 to 54.86. This improvement demonstrates the contribution of learning relevance scores from a text-based teacher to reranking with visual inputs. Query-Local Relevance Discrimination further increases performance to 55.96, showing the additional benefit of learning relative priorities among candidates after distillation. Table 6(b) compares inference throughput across input and caching settings. Cached image inputs achieve 76.83 PPS, compared with 37.31 PPS when visual encoding is included in the measurement. In settings where document visual embeddings are stored in advance, reranking can directly use these representations to improve throughput. These results show that the two training stages improve relevance scoring for documents represented as compressed visual tokens, while inference throughput varies with the execution setting even for the same model.

Table 6: Training-stage effectiveness and inference throughput on 11 BEIR datasets. Effectiveness is averaged across datasets, and throughput is measured using a 12pt font.

(a) Reranking effectiveness

Model	**N@10**
RenderRank	**55.96**
w/o Stage 2	54.86
w/o Stage 1 & 2	50.20

(b) Inference throughput

Inference setting	**PPS ↑**
RenderRank	**76.83**
w/o caching	37.31

7 Conclusion

We presented RenderRank, which renders document text as images and encodes it into compressed visual tokens for query-dependent relevance scoring. Cross-modal distillation and query-local discrimination enable the model to learn relevance from compressed visual inputs and refine the relative scores of candidate documents. RenderRank achieves high reranking performance with fewer input tokens, while its compressed representations allow more document content to contribute to relevance scoring within the same sequence length limit. Our analyses show that both training stages improve relevance scoring, while rendering density controls the trade-off between reranking performance and inference efficiency. This work establishes compressed visual representations as an effective basis for document reranking and opens a path toward representing and reranking longer text documents.

AI USE STATEMENT

We used generative AI tools to assist with translating the paper into English. All authors reviewed the translated text, revised it as needed to preserve the intended meaning, and approved the final paper. We take responsibility for its content.

ETHICS STATEMENT

This work investigates document reranking performance and efficiency using publicly available datasets and models. We describe the data sources and experimental procedures to support transparency and reproducibility. Within this research scope, this work does not raise specific ethical concerns beyond those generally associated with information retrieval systems.

REPRODUCIBILITY STATEMENT

The document rendering procedure and the two training objectives are described in Section 3. Section 4 specifies the training data, evaluation datasets, and baseline models, with computational cost estimation and throughput measurement procedures detailed in Section 5.2. Additional details on data preparation, rendering specifications, training hyperparameters, and hardware are provided in Appendix C.

REFERENCES

Tom Aarsen. Introducing the ettin reranker family, 2026. URL https://huggingface.co/blog/ettin-reranker.

Ishrat Jahan Ananya and Amelie Chatelain. One adapter, both modalities: Field notes from building and serving a multimodal reranker. https://huggingface.co/blog/lightonai/lighton-rerank, 2026.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Davide Baldelli, Junfeng Jiang, Akiko Aizawa, and Paolo Torroni. Twolar: a two-step llm-augmented distillation method for passage reranking. In *European Conference on Information Retrieval*, pp. 470–485. Springer, 2024.

Alexander Bondarenko, Maik Fröbe, Johannes Kiesel, Shahbaz Syed, Timon Gurcke, Meriem Beloucif, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, et al. Overview of touché 2022: argument retrieval. In *International conference of the cross-language evaluation forum for European languages*, pp. 311–336. Springer, 2022.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. In *European Conference on Information Retrieval*, pp. 716–722. Springer, 2016.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.

Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, et al. Glyph: Scaling context windows via visual-text compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37145–37158, 2026.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. Specter: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 2270–2282, 2020.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610, 2021.

Hervé Déjean and Stéphane Clinchant. Efficient listwise reranking with compressed document representations. *arXiv preprint arXiv:2604.26483*, 2026.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. Climate-fever: A dataset for verification of real-world climate claims. *arXiv preprint arXiv:2012.00614*, 2020.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, Ömer Çağatan, Akash Kundu, Martin Bernstorff, Shitao Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf, Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Mariya Hendriksen, Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin Lee, Marek Šuppa, Crystina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail, John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Michael Günther, Mengzhou Xia, Weijia Shi, Xing Han Lù, Jordan Clive, Gayatri Krishnakumar, Anna Maksimova, Silvan Wehrli, Maria Tikhonova, Henil Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide, Lester James Miranda, Alena Fenogenova, Guangyu Song, Ruqiya Bin Safi, Wen-Ding Li, Alessia Borghini, Federico Cassano, Hongjin Su, Jimmy Lin, Howard Yen, Lasse Hansen, Sara Hooker, Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. Mmteb: Massive multilingual text embedding benchmark, 2025. URL https://arxiv.org/abs/2502.13595.

Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1074–1084, 2019.

Yingqi Fan, Xuan Lu, Anhao Zhao, Junlong Tong, Ping Nie, Kai Zou, Yunpu Ma, Wei Zhang, and Xiaoyu Shen. minireranker: Efficient multimodal reranking through visual cache reuse and interaction sparsity. *arXiv preprint arXiv:2606.10759*, 2026.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2701–2715, 2022.

Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. Dbpedia-entity v2: a test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1265–1268, 2017.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666*, 2020.

Seongtae Hong, Youngjoon Jang, Jungseob Lee, Seungyoon Lee, and Heuiseok Lim. Lamar: An open language-aware multilingual alignment reranker, 2026. URL https://arxiv.org/abs/2607.22042.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In *Proceedings of the 2021 conference of the north American chapter of the association for computational linguistics: Human language technologies*, pp. 1419–1436, 2021.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pp. 874–880, 2021.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251): 1–43, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 39–48, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. Parade: Passage representation aggregation fordocument reranking. *ACM Transactions on Information Systems*, 42(2):1–26, 2023.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Defu Lian, Yingxia Shao, and Zheng Liu. Making text embedders few-shot learners. In *International Conference on Learning Representations*, volume 2025, pp. 38107–38124, 2025a.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Chen Keqin, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Xianming Li, Aamir Shakir, Rui Huang, Julius Lipp, Benjamin Clavié, and Jing Li. Prorank: Prompt warmup via reinforcement learning for small language models reranking. *arXiv preprint arXiv:2506.03487*, 2025b.

Yanhong Li, Zixuan Lan, and Jiawei Zhou. Text or pixels? evaluating efficiency and understanding of llms with visual text inputs. In *EMNLP (Findings)*, pp. 10564–10578, 2025c.

Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. Relora: High-rank training through low-rank updates, 2023. *URL https://arxiv. org/abs/2307.05695*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jonas F Lotz, Elizabeth Salesky, Phillip Rust, and Desmond Elliott. Text rendering strategies for pixel language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10155–10172, 2023.

Xuan Lu, Haohang Huang, Yingqi Fan, Junlong Tong, Yuxuan Zhang, Ping Nie, Rui Meng, and Xiaoyu Shen. Comprank: Efficient llm reranking via token-level compression and decoding-free scoring. *arXiv preprint arXiv:2606.11700*, 2026.

Zhiheng Lyu, Xueguang Ma, and Wenhu Chen. Pixelworld: How far are we from perceiving everything as pixels? *arXiv preprint arXiv:2501.19339*, 2025.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2421–2425, 2024.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. Www'18 open challenge: financial opinion mining and question answering. 2018.

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. *arXiv preprint arXiv:2407.15831*, 2024.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the association for computational linguistics: EMNLP 2020*, pp. 708–718, 2020.

Zhiyuan Peng, Ting-Ruen Wei, Tingyu Song, and Yilun Zhao. Efficiency-effectiveness reranking FLOPs for LLM-based rerankers. In Saloni Potdar, Lina Rojas-Barahona, and Sebastien Montella (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2782–2791, Suzhou (China), November 2025. Association for Computational Linguistics. ISBN 979-8-89176-333-3. doi: 10.18653/v1/2025.emnlp-industry.186. URL https://aclanthology.org/2025.emnlp-industry.186/.

Nicholas Pipitone, Ghita Houir Alami, Advaith Avadhanam, Anton Kaminskyi, and Ashley Khoo. zelo: Elo-inspired training method for rerankers and embedding models, 2025. URL https://arxiv.org/abs/2509.12541.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Phillip Rust, Jonas F Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. Language modelling with pixels. *arXiv preprint arXiv:2207.06991*, 2022.

Ferdinand Schlatt, Maik Fröbe, Harrisen Scells, Shengyao Zhuang, Bevan Koopman, Guido Zuccon, Benno Stein, Martin Potthast, and Matthias Hagen. Rank-distillm: Closing the effectiveness gap between cross-encoders and llms for passage re-ranking. In *European Conference on Information Retrieval*, pp. 323–334. Springer, 2025.

Aamir Shakir, Darius Koenig, Julius Lipp, and Sean Lee. Boost your search with the crispy mixedbread rerank models, 2024. URL https://www.mixedbread.ai/blog/mxbai-rerank-v1.

Raphaël Sourty, Antoine Chaffin, Paulo Roberto Moura Junior, and Amélie Chatelain. Denseon with the lateon: Fully open dense and late-interaction models for multilingual, long-context, and code search, 2026. URL https://arxiv.org/abs/2607.27178.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 14918–14937, 2023.

Yiqun Sun, Pengfei Wei, and Lawrence B Hsieh. Very efficient listwise multimodal reranking for long documents. *arXiv preprint arXiv:2605.11864*, 2026.

Lv Tang, Tianyi Zheng, Yang Liu, Bo Li, and Xingyu Li. Visual text compression as measure transport. *arXiv preprint arXiv:2605.06708*, 2026.

Mitchell Keren Taraday, Shahaf Wagner, and Chaim Baskin. Efficient discriminative joint encoders for large scale vision-language reranking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=UXtTBAyqVB.

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

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

Nandan Thakur, Crystina Zhang, Xueguang Ma, and Jimmy Lin. Fixing data that hurts performance: Cascading llms to relabel hard negatives for robust information retrieval, 2025. URL https://arxiv.org/abs/2505.16967.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, 2018.

Michael Tschannen, Basil Mustafa, and Neil Houlsby. Clippo: Image-and-language understanding from pixels only. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11006–11017. IEEE, 2023.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. Trec-covid: constructing a pandemic information retrieval test collection. In *ACM SIGIR Forum*, volume 54, pp. 1–12. ACM New York, NY, USA, 2021.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 241–251, 2018.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 7534–7550, 2020.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Yu Wang, Shengyao Zhuang, Xueguang Ma, Zongyu Wu, Jimmy Lin, Vivek Srikumar, and Zhichao Xu. Tevatron-elastic: A unified abstraction for training elastic retrievers and rerankers. *arXiv preprint arXiv:2608.08809*, 2026.

Chenghao Xiao, Zhuoxu Huang, Danlu Chen, G Thomas Hudson, Yizhi Li, Haoran Duan, Chenghua Lin, Jie Fu, Jungong Han, and Noura Al Moubayed. Pixel sentence representation learning. *arXiv preprint arXiv:2402.08183*, 2024.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in neural information processing systems*, 37:121156–121184, 2024.

Chaohao Yuan, Ruifeng Yuan, Zhuoxu Huang, Yu Rong, Hong Cheng, Hou Pong Chan, and Chenghao Xiao. On the design fundamentals of pixel text representation learning. *arXiv preprint arXiv:2609.01147*, 2026.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1393–1412, 2024.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. Longembed: Extending embedding models for long context retrieval. *arXiv preprint arXiv:2404.12096*, 2024.

Shengyao Zhuang, Zhichao Xu, and Ivano Lauriola. Layer-wise token compression for efficient document reranking. *arXiv preprint arXiv:2605.20683*, 2026.

A Further Related Work

Learned compressed representations have been explored as substitutes for document token sequences to reduce the computational cost of reranking (Déjean & Clinchant, 2026). In this approach, a separate compressor encodes each document into representations associated with a fixed number of memory tokens, and the compressor and reranker are jointly trained with a ranking objective. After training, document representations are computed in advance, allowing the reranker to process the compressed representations alongside the query instead of the original text.

Other approaches reduce token computation within the reranker. One approach encodes documents independently, selects the document key–value states accessible to query-side attention, and directly supervises relevance scores derived by aggregating attention weights (Lu et al., 2026). Another pools token representations at an intermediate layer to shorten the sequences processed by subsequent layers, training the reranker to adapt to the compressed representations (Zhuang et al., 2026). Subsequent work trains a single model across multiple compression ratios, allowing the ratio to be adjusted at inference time (Wang et al., 2026).

Rather than jointly training a text compressor with the reranker or selecting or merging tokens within the language model used for reranking, RenderRank represents rendered documents through the visual encoder of an existing VLM. We freeze the visual encoder and feature mergers and train the language model for relevance scoring, with the number of visual tokens varying according to the rendering configuration and document length. Our focus is on adapting existing visual text representations to document reranking through query-dependent relevance training, and on characterizing the resulting trade-offs between reranking performance and inference efficiency.

B Generation Experiments for Rendering Evaluation

We evaluate rendering configurations in terms of task performance and token efficiency using Gemma-4-26B-A4B-it (Team, 2026) and Qwen3.6-35B-A3B (Qwen Team, 2026) on the QASPER (Dasigi et al., 2021), GovReport (Huang et al., 2021), and Multi-News (Fabbri et al., 2019) datasets in LongBench (Bai et al., 2024). We report F1 for QASPER and ROUGE-L for both summarization tasks. Documents are rendered in Roboto Regular, combining font sizes of 10pt, 11pt, and 12pt with line spacing of 1.0, 1.1, and 1.2. Images have a fixed width of 896 pixels and a height that adjusts to the content, up to 896 pixels. Content exceeding this height continues onto the next image without overlap, and the resulting images are supplied in document order. We use the recommended sampling settings for each model and keep them unchanged across all rendering configurations and the text baseline. Thinking mode is disabled for both models.

Table 7 supplements the main-text results with average document-token counts and an additional evaluation on Multi-News. Document-token counts exclude queries and instructions, and token reduction is calculated relative to the text baseline for each model and dataset. The additional results

Table 7: Performance and document-token counts across rendering configurations. QASPER is evaluated using F1, and GovReport and Multi-News using ROUGE-L. Tokens denotes the average document-token count. Blue shading indicates a reduction in document-token count relative to text input, while red shading indicates an increase.

Font	LS	Gemma-4-26B-A4B-it						Qwen3.6-35B-A3B					
		QASPER		GovReport		Multi-News		QASPER		GovReport		Multi-News	
		F1	Tokens	ROUGE-L	Tokens	ROUGE-L	Tokens	F1	Tokens	ROUGE-L	Tokens	ROUGE-L	Tokens
Text	–	47.73	4,998	28.78	10,551	22.57	2,748	51.45	5,023	31.10	10,579	23.41	2,652
10pt	1.0	45.90	3,064	28.27	6,223	22.24	1,843	41.32	1,911	30.55	4,197	22.29	1,050
	1.1	45.72	3,434	28.46	7,090	22.02	2,046	43.18	2,192	30.75	4,823	22.80	1,202
	1.2	44.58	3,566	28.27	7,449	22.21	2,100	45.80	2,303	30.42	5,079	22.74	1,253
11pt	1.0	47.48	3,867	28.53	8,117	22.40	2,274	45.26	2,528	30.53	5,571	22.59	1,378
	1.1	48.32	4,361	28.69	9,125	22.24	2,572	46.46	2,846	30.55	6,303	23.12	1,535
	1.2	45.76	4,579	28.68	9,696	22.14	2,691	47.41	3,019	30.85	6,678	23.19	1,637
12pt	1.0	48.07	4,339	28.67	9,072	22.20	2,539	48.72	2,829	30.82	6,262	22.98	1,521
	1.1	47.93	4,848	28.75	10,266	21.90	2,764	45.35	3,223	30.66	7,136	23.35	1,725
	1.2	46.50	5,274	28.60	11,401	22.24	3,063	45.33	3,557	30.69	7,943	23.41	1,912

Table 8: An example from Touché-2020 showing the query, document text, and two rendered images.

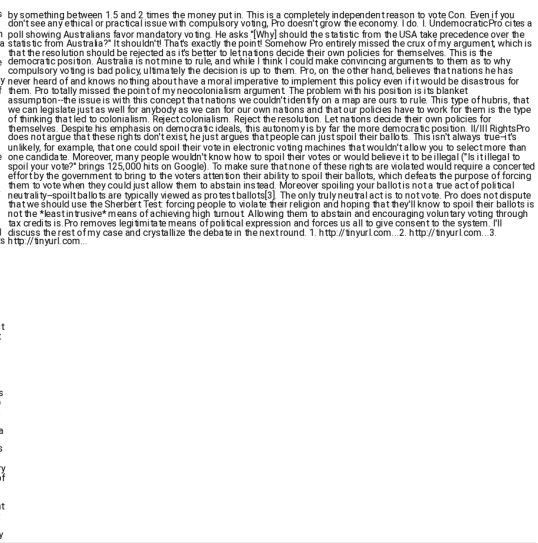
Query: Should election day be a national holiday?

Document: Autumn Regular Tournament: In a democracy, voting ought to be compulsory Thanks, Philo. =Pro case=FrameworkPro argues that the only concern that should govern policy is if that policy is "democratic". This is absurd–democracies are governments and as I articulated in my framework, governance is fundamentally a balancing act between individual rights and the common good. Good governments are ones which make acceptable trade offs on this scale. ... The only truly neutral act is to not vote. Pro does not dispute that we should use the Sherbert Test: forcing people to violate their religion and hoping that they'll know to spoil their ballots is not the *least intrusive* means of achieving high turnout. Allowing them to abstain and encouraging voluntary voting through tax credits is.Pro removes legitimitate means of political expression and forces us all to give consent to the system. I'll discuss the rest of my case and crystallize the debate in the next round. 1. http://tinyurl.com...2. http://tinyurl.com...3. http://tinyurl.com...

Image 1



Image 2



show that changes in token count do not consistently translate into corresponding changes in performance. For example, increasing line spacing from 1.0 to 1.2 at 12pt improves the Multi-News score of Qwen from 22.98 to 23.41, but reduces token savings from 42.66% to 27.92%. In contrast, Gemma at 12pt with line spacing of 1.2 uses more document tokens than the text baseline without reaching its performance. These results highlight that the effects of rendering configurations vary across models and tasks, underscoring the importance of considering both performance and token efficiency when determining the rendering configuration.

C Training Details

Rendering We render document text into images while retaining each query as text. Before rendering, all whitespace characters, including line breaks, tabs, and consecutive spaces, are normalized to a single space; original paragraph boundaries and manual line breaks are therefore not preserved. Documents are rendered in black text on a white background using Roboto Regular at 12 pt (16 px at 96 DPI), with zero margins and Pillow's default text antialiasing. Each image has a fixed width of 896 px. Text is wrapped according to its measured pixel width under the selected font, rather than a fixed character count, and words exceeding the image width are split at character boundaries.

We use a line spacing of 1.0, corresponding to a 16 px vertical advance between consecutive lines. Each image contains at most 56 lines, yielding a maximum height of 896 px. Longer documents continue onto subsequent images without overlap or repeated text. The height of each image is rounded up to the nearest multiple of 32 px, with a minimum of 32 px and a maximum of 896 px; thus, the final image is cropped in height to accommodate its remaining lines rather than padded to a full square. Table 8 presents an example query–document pair and its corresponding rendered document images.

Table 9: Dataset composition for the two training stages.

Stage 1			Stage 2	
Dataset	**Records**	**Pairs**	**Dataset**	**Records**
FiQA	13,008	52,032	ArguAna	4,065
HotpotQA	142,537	570,148	FEVER	8,151
Natural Questions	114,030	456,120	FiQA	5,496
MS MARCO	518,851	2,075,404	HotpotQA	10,246
FEVER	126,852	507,408	MS MARCO Passage	46,926
SQuAD v2	129,241	516,964	Natural Questions	6,043
TriviaQA	511,074	2,044,296	SciDocsRR	12,654
MIRACL	6,191	24,764		
MLDR	9,086	36,344		
Total	**1,570,870**	**6,283,480**	**Total**	**93,581**

Dataset Table 9 summarizes the source dataset composition for both training stages. For Cross-Modal Relevance Distillation in Stage 1, we use the English retrieval fine-tuning data[1] released as part of the DenseOn and LateOn work (Ananya & Chatelain, 2026). This release contains retrieval training data with mined hard negatives filtered using the NV-Retriever (Moreira et al., 2024) procedure, including English examples from the MLDR training split. The source data comprise 1.57M training records, each containing a query, one positive document, and ten negative documents. Since this stage minimizes the discrepancy between student and teacher relevance scores for individual query–document pairs, we convert each record into separate pairs by pairing the query with the positive document and the first three negatives in the provided candidate order. We retain all records, yielding 6.28M training pairs. For Query-Local Relevance Discrimination in Stage 2, we use RLHN-100K[2] (Thakur et al., 2025), which provides positive and negative documents for each query. It is derived from seven datasets in the BGE training collection (Li et al., 2025a), with cascading LLMs used to identify and relabel false negatives while leaving the ArguAna subset unchanged. This stage learns relative relevance scores within the same query, so we organize the provided documents into candidate sets. We use all records and retain the first positive document and the first seven negatives in the provided order for each record. Each query and its candidate set form a training instance, with contrastive comparisons restricted to that set.

Hyperparams Both stages are trained for one epoch using AdamW (Loshchilov & Hutter, 2017) with a learning rate of 5×10^{-5}, no weight decay, and a linear learning rate scheduler. The warmup ratios are 0.10 and 0.05 for Stage 1 and Stage 2, respectively. We clip the gradient norm to 1.0 and enable BF16 mixed precision and gradient checkpointing. Stage 1 uses 2 pairs per GPU with 64 gradient accumulation steps, resulting in a total batch size of 1,024 pairs. Stage 2 uses 8 pairs per GPU with 2 gradient accumulation steps, resulting in a total batch size of 128 pairs. Each query is paired with one positive and seven negative documents. The temperature for the Stage 2 contrastive loss is set to 0.1. The maximum sequence length is set to 32,768 for both training stages and inference. Both stages apply LoRA to the text decoder with a rank of 64, an alpha of 128, and a dropout rate of 0.05, while keeping the vision encoder and visual feature mergers frozen.

Reproducibility We implement both training stages using Sentence Transformers[3] (Reimers & Gurevych, 2019) and set the random seed to 42 for training, including data shuffling. The reported results are obtained from a single training run for each configuration. Data preparation procedures and training hyperparameters are specified above to facilitate reproduction of the experiments.

Hardware We trained RenderRank on eight NVIDIA RTX A6000 GPUs, each with 48GB of memory, using a server equipped with two Intel Xeon Gold 6230R processors, each with 26 cores. Inference throughput was measured on a single GPU.

[1] https://huggingface.co/datasets/lightonai/embeddings-fine-tuning-filtered-en
[2] https://huggingface.co/datasets/rlhn/rlhn-100K
[3] https://github.com/huggingface/sentence-transformers

D Detailed Results

We provide the full results for each of the 11 BEIR datasets. Tables 10 and 11 report the average number of input tokens and inference throughput for each evaluated reranker. Tables 12, 13, and 14 present the reranking performance, average number of input tokens, and throughput of RenderRank across rendering font sizes. Table 15 provides the full performance results for the training-stage ablations.

Table 10: Full results for average input token counts of rerankers on 11 BEIR datasets.

		BEIR											
Model	**Params**	**AA**	**CFV**	**DBP**	**FQA**	**FVR**	**HQA**	**NFC**	**SD**	**SF**	**TC**	**TCH**	**Avg.**
gte-reranker-modernbert-base	150M	590.82	349.43	97.31	388.50	224.47	125.54	357.15	287.82	365.39	345.06	820.29	359.25
mxbai-rerank-large-v1	435M	563.76	329.37	89.29	383.63	205.76	116.99	342.54	276.43	354.01	341.52	818.69	347.45
bge-reranker-large	560M	642.82	417.03	104.19	420.74	255.46	134.94	420.11	339.02	448.81	408.58	910.87	409.32
Qwen3-Reranker-0.6B	596M	670.01	429.17	169.26	460.64	303.84	199.44	449.36	353.84	464.23	435.73	890.13	438.70
LAMAR-600m	600M	642.82	417.03	104.19	420.74	255.46	134.94	420.11	339.02	448.81	408.58	910.87	409.32
llama-nemotron-rerank-1b-v2	1.2B	591.56	353.67	99.53	387.35	228.67	128.74	370.61	284.69	387.86	357.02	814.20	363.99
mxbai-rerank-large-v2	1.5B	681.26	440.21	180.27	471.69	314.92	210.46	460.37	364.93	475.25	446.78	901.25	449.76
LightOn-rerank-PW-2B	2.2B	647.73	406.94	146.84	439.11	281.40	177.41	426.36	331.34	441.63	411.52	868.13	416.22
bge-reranker-v2-gemma	2.5B	631.43	388.00	133.51	432.04	264.31	163.39	401.57	316.95	410.03	389.95	859.04	399.11
Qwen3-Reranker-4B	4B	670.01	429.17	169.26	460.64	303.84	199.44	449.36	353.84	464.23	435.73	890.13	438.70
zerank-2-reranker	4B	612.11	371.28	111.10	402.19	246.37	141.67	391.17	295.13	406.51	377.76	832.26	380.69
LightOn-rerank-PW-4B	4.5B	647.73	406.94	146.84	439.11	281.40	177.41	426.36	331.34	441.63	411.52	868.13	416.22
RenderRank (Ours)	2.1B	518.35	283.33	128.82	280.62	197.20	150.52	278.68	257.07	292.15	284.13	519.92	290.07

Table 11: Full throughput results (PPS) of rerankers on 11 BEIR datasets.

		BEIR											
Model	**Params**	**AA**	**CFV**	**DBP**	**FQA**	**FVR**	**HQA**	**NFC**	**SD**	**SF**	**TC**	**TCH**	**Avg.**
gte-reranker-modernbert-base	150M	45.35	60.35	378.30	44.93	88.48	235.90	83.25	72.61	67.88	83.00	16.89	106.99
mxbai-rerank-large-v1	435M	77.62	72.92	442.98	72.03	101.50	229.39	82.19	102.99	80.61	83.36	77.97	129.42
bge-reranker-large	560M	48.05	48.46	167.70	35.59	53.84	101.97	42.00	52.59	49.08	49.06	48.27	63.33
Qwen3-Reranker-0.6B	596M	32.87	41.48	183.17	37.38	60.97	127.17	55.55	56.16	51.67	55.43	18.70	65.50
LAMAR-600m	600M	31.45	32.28	202.17	32.72	75.28	150.04	48.65	58.68	44.76	49.08	17.98	67.55
llama-nemotron-rerank-1b-v2	1.24B	40.33	52.24	259.71	42.13	72.90	163.05	64.35	64.48	59.70	64.75	21.09	82.25
mxbai-rerank-large-v2	1.5B	35.20	50.82	133.88	49.34	70.29	94.71	48.11	56.99	47.71	45.03	24.95	59.73
LightOn-rerank-PW-2B	2.2B	15.38	22.57	66.08	23.53	34.34	53.68	20.50	31.10	22.40	24.18	12.72	29.68
bge-reranker-v2-gemma	2.51B	18.14	23.56	97.04	18.46	45.49	67.15	28.25	28.85	26.17	20.32	17.19	35.51
Qwen3-Reranker-4B	4B	12.59	18.62	50.42	17.60	25.75	40.33	17.14	23.35	14.91	18.51	7.33	22.41
zerank-2-reranker	4.02B	11.42	21.06	70.42	20.14	30.58	53.61	19.50	27.80	19.27	20.69	9.20	27.61
LightOn-rerank-PW-4B	4.5B	7.00	9.34	25.01	9.89	14.54	22.25	9.45	13.02	8.52	10.05	5.22	12.21
RenderRank (Ours)	2.1B	35.53	68.97	151.77	64.43	87.86	113.36	66.41	79.16	68.79	70.79	38.12	76.83

Table 12: Full reranking performance results (NDCG@10) for RenderRank with different rendering font sizes on BEIR.

	BEIR											
Input	**AA**	**CFV**	**DBP**	**FQA**	**FVR**	**HQA**	**NFC**	**SD**	**SF**	**TC**	**TCH**	**Avg.**
10 pt	67.42	33.79	45.43	40.83	88.20	77.94	36.26	20.09	75.97	83.23	36.24	55.04
11 pt	68.41	34.42	45.97	41.92	88.87	78.68	36.49	20.69	76.18	83.73	35.16	55.50
12 pt	69.11	35.91	46.19	42.66	88.92	78.60	37.31	20.75	77.00	84.82	34.27	55.96
13 pt	68.82	35.68	46.38	42.44	89.31	79.05	36.99	20.97	77.82	84.31	34.46	56.02
14 pt	69.54	36.24	47.32	42.92	89.30	79.17	37.30	21.09	78.08	85.06	34.67	56.43

Table 13: Full results for average input token counts of RenderRank with different rendering font sizes on BEIR.

	BEIR											
Input	**AA**	**CFV**	**DBP**	**FQA**	**FVR**	**HQA**	**NFC**	**SD**	**SF**	**TC**	**TCH**	**Avg.**
10 pt	460.10	228.64	121.91	223.08	167.75	139.79	218.55	207.48	231.83	225.19	376.91	236.47
11 pt	502.14	267.68	124.87	264.16	189.13	147.21	263.43	244.60	276.35	268.96	474.97	274.86
12 pt	518.35	283.33	128.82	280.62	197.20	150.52	278.68	257.07	292.15	284.13	519.92	290.07
13 pt	549.75	313.62	147.08	312.76	219.96	168.11	310.62	284.64	323.66	315.89	586.41	321.14
14 pt	605.08	364.90	151.26	366.89	248.68	178.39	367.26	333.28	380.95	371.74	709.15	370.69

Table 14: Full throughput results (PPS) for RenderRank with different rendering font sizes on BEIR.

	BEIR											
Input	**AA**	**CFV**	**DBP**	**FQA**	**FVR**	**HQA**	**NFC**	**SD**	**SF**	**TC**	**TCH**	**Avg.**
10 pt	37.89	74.65	167.24	89.46	116.13	133.59	93.31	96.62	86.20	89.76	54.18	94.46
11 pt	35.94	72.68	159.66	75.50	101.63	115.95	76.49	83.03	72.76	67.07	42.30	82.09
12 pt	35.53	68.97	151.77	64.43	87.86	113.36	66.41	79.16	68.79	70.79	38.12	76.83
13 pt	33.60	62.36	126.00	64.40	88.58	113.85	66.10	72.04	62.47	64.50	30.08	71.27
14 pt	31.70	47.73	134.06	55.04	70.06	107.84	56.17	61.72	47.09	55.35	27.33	63.10

Table 15: Full training-stage ablation results (NDCG@10) with text and image inputs on 11 BEIR datasets.

		BEIR											
Model	**Input**	**AA**	**CFV**	**DBP**	**FQA**	**FVR**	**HQA**	**NFC**	**SD**	**SF**	**TC**	**TCH**	**Avg.**
Qwen3-VL-Reranker-2B	Text	57.31	32.10	45.33	43.22	86.85	77.86	37.47	20.71	77.39	82.87	35.17	54.21
	Image	58.68	22.92	40.84	36.32	79.57	73.23	35.27	18.84	77.67	79.13	29.69	50.20
Stage 1	Text	57.06	32.68	46.43	44.37	88.88	77.88	37.67	20.99	78.32	84.67	40.75	55.43
	Image	60.56	36.17	45.49	41.77	89.00	78.19	36.72	20.14	76.83	83.47	35.15	54.86
Stage 2 (RenderRank)	Text	71.23	33.34	46.76	45.01	89.10	78.74	37.85	22.18	78.42	85.34	41.37	57.21
	Image	69.11	35.91	46.19	42.66	88.92	78.60	37.31	20.75	77.00	84.82	34.27	55.96