MM-DIFFEMBED: WHEN MULTIMODAL DIFFUSION LLMS MEET MULTIMODAL DOCUMENT RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal document retrieval (MDR) requires representations that preserve fine-grained page evidence while contextualizing it across text, tables, and layout. Most multi-vector retrievers use autoregressive multimodal language models, leaving the role of diffusion-pretrained contextualization in this setting unclear. We introduce MM-DIFFEMBED, a framework that adapts multimodal diffusion language models (MDLMs) into single-pass multi-vector retrievers, instantiated as COLLAVIDA and COLSDAR-VL. The framework retains contextualized token states, applies a shared projection and `MaxSim` scorer, and supports prompt-denoising or query-vector decorrelation during adaptation. Across ViDoRe-v1, ViDoRe-v2, and the reasoning-intensive MRMR benchmark, COLLAVIDA and COLSDAR-VL exceed their strongest respective LLaVA- and Qwen-family baselines on MRMR by 3.79 and 4.76 points. ViDoRe results show a backbone-dependent trade-off rather than a universal diffusion advantage, and efficiency measurements reveal that single-pass encoding does not eliminate the costs of dense page representations. Overall, the results establish a concrete testbed for studying diffusion-pretrained multi-vector encoders in future MDR research.

1 INTRODUCTION

Multimodal document retrieval (MDR) seeks evidence distributed across a page's text, tables, diagrams, and layout. A financial query may depend on a value and its header, while a scientific query may link a diagram to its explanation. Encoding page images avoids structure loss from plain text conversion (Faysse et al., 2025; Yu et al., 2025), and multi-vector representations keep local evidence available until scoring. Unlike natural-image retrieval, where a global concept often determines relevance, MDR must preserve small decisive details together with their surrounding context (Yan et al., 2026a). This makes MDR a demanding testbed for token-level representations rather than only for image–caption similarity alone.

Large language models (LLMs) equipped with vision form multimodal large language models (MLLMs), and the dominant LLaVA and Qwen-VL families use autoregressive (AR) generation (Liu et al., 2023; 2024; Wang et al., 2024; Bai et al., 2025). Their visual and linguistic priors have produced strong multi-vector retrievers, but causal contextualization introduces a structural mismatch with MDR: a page is an observed object whose relevant evidence may occur before or after the position being represented. This does not imply that AR retrievers are inadequate. Instead, it motivates asking whether a different contextualization prior can complement token-level matching when the multi-vector projection and `MaxSim` scorer are held fixed.

Diffusion language models (DLMs) reconstruct masked tokens and therefore expose bidirectional context during representation learning; multimodal instances include LaViDa and SDAR-VL (Nie et al., 2025; Li et al., 2025; Cheng et al., 2025b). However, an embedding-space study reports no uniform advantage for diffusion VLMs over AR alternatives (Wang et al., 2026d), and recent diffusion work has largely focused on model understanding or generation rather than MDR. The open question is consequently not whether diffusion is universally better, but whether its contextualized states remain useful after retrieval adaptation and token-level scoring. Recent MDLMs make such states accessible in a single observed-input pass (Li et al., 2026d; Sun et al., 2026), enabling a direct study with native LaViDa and SDAR-VL backbones.

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.

We introduce MM-DIFFEMBED, a retrieval adaptation framework for MDLMs in multi-vector MDR. Its COLLAVIDA and COLSDAR-VL instantiations use LaViDa-L and SDAR-VL-8B, project contextualized hidden states into a shared 128-dimensional space, and rank pages with `MaxSim`. The main configurations retain all valid token states and use prompt denoising or query-vector decorrelation as model-specific training regularization. At inference, each observed query or page is encoded once; no iterative reverse-diffusion decoding is required.

We compare both instantiations with four AR multi-vector retrievers on ViDoRe-v1, ViDoRe-v2, and reasoning-intensive MRMR (Faysse et al., 2025; Macé et al., 2025; Zhang et al., 2026). Under the shared MRMR protocol, COLLAVIDA improves over ColLLaVA-Llama3 by 3.79 points and COLSDAR-VL over ColQwen2-7B by 4.76 points. On ViDoRe, the outcome is backbone-dependent: COLLAVIDA is 0.91 points above ColLLaVA-Llama3 on ViDoRe-v2 but remains lower on ViDoRe-v1, while COLSDAR-VL trails Qwen on both versions. Readout analysis favors retaining all valid token states, whereas efficiency measurements expose a quality–cost trade-off rather than an automatic speed benefit. Because the pretrained checkpoints, visual frontends, and adaptation recipes also differ, these comparisons characterize the evaluated systems and do not isolate a causal effect of the diffusion backbone. Our contributions are:

❶ **Framework Design.** We establish MM-DIFFEMBED as a single-pass multi-vector MDR framework for diffusion-pretrained multimodal backbones, instantiated with COLLAVIDA and COLSDAR-VL under a shared projection and `MaxSim` interface.

❷ **Retrieval Adaptation.** We develop and analyze retrieval adaptation choices for diffusion contextual states, including full-token readout, prompt denoising, and query-vector decorrelation, and quantify their effects on representation granularity and benchmark performance.

❸ **Systematic Evaluation.** We provide a systematic comparison of diffusion- and autoregressive-based multi-vector retrievers across semantic and reasoning-intensive MDR benchmarks, together with readout, hyperparameter, and efficiency analyses that expose their quality–cost trade-offs.

2 RELATED WORK

2.1 MULTIMODAL DOCUMENT RETRIEVAL

General multimodal retrieval often asks whether a natural image and a text description refer to the same concept, as in CLIP-style image–caption matching (Radford et al., 2021; Zhai et al., 2023), or learns a universal image–text space for several downstream tasks (Jiang et al., 2025; Zhang et al., 2025b; Günther et al., 2025). MDR has a different failure mode: a small number in a table, a chart axis, a footnote, or the spatial relation between a heading and a figure can determine relevance even when the page and query have similar global topics, as illustrated in Figure 1. Document Screenshot Embedding and VisRAG retain rendered pages rather than relying exclusively on OCR (Ma et al., 2024; Yu et al., 2025). ColPali (Faysse et al., 2025) made multi-vector matching over page-image tokens practical, followed by ColQwen2/2.5, Jina-embeddings-v4, and Llama Nemoretriever ColEmbed (Günther et al., 2025; Xu et al., 2025b). ViDoRe-v2 broadens the collections and query phenomena used to assess MDR (Macé et al., 2025), while MRMR adds expert-domain, theorem-oriented, and contradiction retrieval in which lexical overlap is unreliable (Zhang et al., 2026). These benchmarks make token-level evidence preservation important, and motivate our study of diffusion-pretrained contextual tokens as an MDR alternative to AR retrievers in this evidence-centric setting. This distinction defines the scope of our backbone comparison.

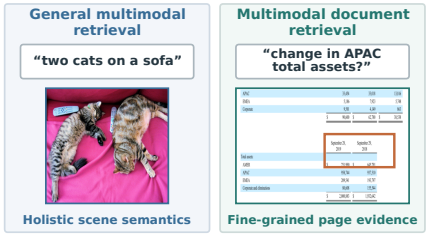


Figure 1: General multimodal retrieval versus MDR; orange box is illustrative.

2.2 MULTIMODAL LLMS

Most MLLMs use an AR decoder (Wu et al., 2023; Yin et al., 2024; Ma et al., 2026). LLaVA established visual instruction tuning through this interface (Liu et al., 2023; 2024).

Table 1: Diffusion embedding studies. Base is the backbone, Readout the retained states, and Multi indicates multi-vector scoring.

Relevant Work	Modality	Base	Readout	Multi
DiffEmbed (Zhang et al., 2025a)	Text	DLM	Mean	✗
DiffRetriever (Wang et al., 2026a)	Text	DLM	Mask	✓
pplx-embed (Eslami et al., 2026)	Text	DLM	Mean	✗
Wang et al. (2026d)	Image+text	MDLM	Mean	✗
MM-DIFFEMBED (Ours)	MDR	MDLM	All	✓

Later LLaVA-family systems improved resolution handling and visual task transfer (Li et al., 2024). Qwen2-VL and Qwen2.5-VL similarly emphasize resolution-aware perception and document understanding (Wang et al., 2024; Bai et al., 2025). DLMs provide another pretraining objective: LLaDA reconstructs masked tokens with bidirectional context (Nie et al., 2025), while LaViDa extends this paradigm to multimodal understanding and SDAR-VL combines within-block denoising with causal dependencies across blocks (Li et al., 2025; Cheng et al., 2025b). Their native generation procedures are distinct from the single-pass encoding used by a retriever. Wang et al. (2026d) compare diffusion and AR embeddings across 32 classification, VQA, and retrieval datasets using a single-vector interface; our MDR setting instead retains multiple contextual tokens, applies `MaxSim`, and includes both LaViDa and SDAR-VL. The comparison is summarized in Table 1; it also situates text-only DiffEmbed, DiffRetriever, and pplx-embed, which respectively use mean-pooled or masked-position representations of DLMs (Zhang et al., 2025a; Wang et al., 2026a; Eslami et al., 2026). Thus, our question is complementary to the single-vector study: does a document-oriented multi-vector interface expose useful local evidence that sequence pooling can discard, without assuming that bidirectionality alone guarantees better embeddings? We use this distinction to isolate the representation interface studied in this paper.

2.3 MULTI-VECTOR RETRIEVAL

Let a query and a candidate page be represented by token matrices $\mathbf{Q} \in \mathbb{R}^{m \times r}$ and $\mathbf{D} \in \mathbb{R}^{n \times r}$, where m and n are the retained token counts and r is the shared embedding dimension. Their token-similarity matrix is $\mathbf{S} = \mathbf{Q}\mathbf{D}^\top \in \mathbb{R}^{m \times n}$, and the pair score is $s(q, d) = \sum_{a=1}^{m} \max_{1 \le b \le n} \mathbf{S}_{ab}$. ColBERT introduced this decomposition for text, matching each query token to its strongest passage token (Khattab & Zaharia, 2020); ColBERTv2 improved its training and residual-compressed storage (Santhanam et al., 2022). MDR systems such as ColPali apply the same idea to visual page tokens (Faysse et al., 2025), while ColQwen2/2.5 and ColEmbed provide additional VLM-based multi-vector examples in released model families (Xu et al., 2025b). Compared with a single pooled vector, the matrix formulation can retain separate evidence for a chart label and its nearby value, but it also increases index size and `MaxSim` scoring cost. Crucially, this scoring stage only aggregates information already present in $\mathbf{Q}$ and $\mathbf{D}$; it does not perform a new cross-modal reasoning pass. MM-DIFFEMBED therefore keeps this interface fixed and studies the backbone contextualization, readout, and adaptation choices that determine what evidence each token can carry. A detailed discussion is provided in Appendix A, which includes more recent relevant works.

3 METHODOLOGY

Figure 2 summarizes the shared independent encoding, projection, multi-vector, and `MaxSim` path, contrasting causal AR contextualization with masked-token diffusion contextualization.

3.1 TASK FORMULATION

Let $\mathcal{C} = \{d_j\}_{j=1}^{N}$ be a corpus of N document pages. Each page d_j is observed as an RGB image $\mathbf{I}_{d_j} \in \mathbb{R}^{u_j \times v_j \times 3}$, where u_j and v_j are its pixel height and width. Given a query q, the task is to produce a ranking π_q of the corpus according to relevance $y(q, d_j) \ge 0$. In the principal setting q is text; the same document encoder also supports the image-only queries used in our Negation evaluation. Section 4.1 specifies query handling; Appendix B illustrates tasks and prompts. Training uses paired queries and relevant pages, $\mathcal{S} = \{(q_i, d_i^+)\}_{i=1}^{M}$, where M is the number of training pairs and d_i^+ is the positive page for q_i. Query and page are encoded independently, so the page representation can be computed before a query arrives.

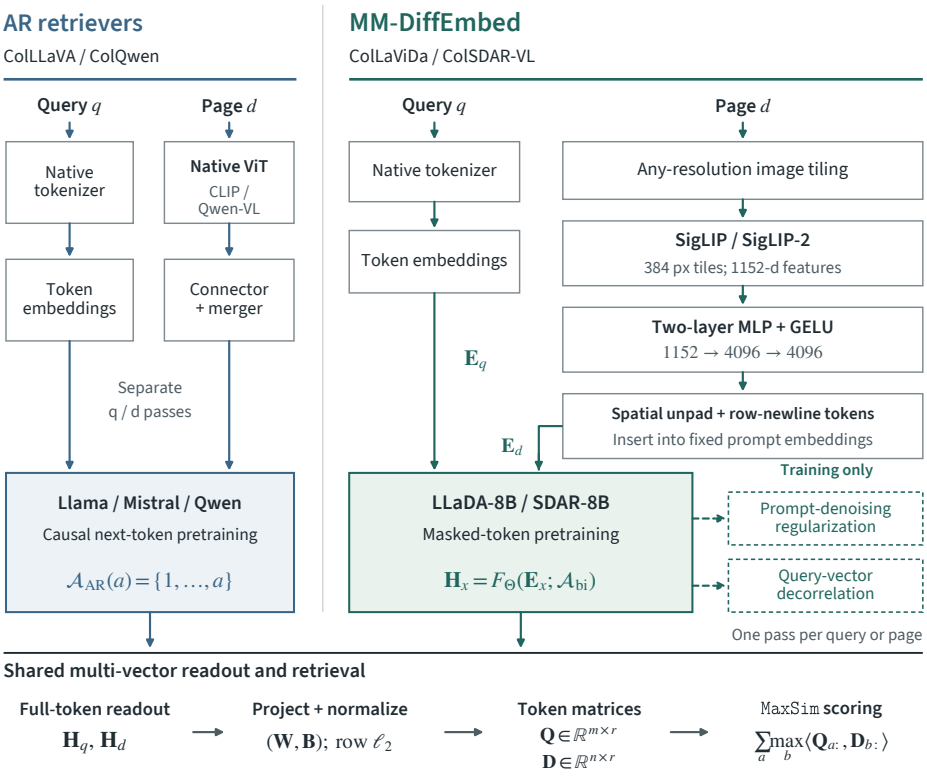


Figure 2: Architecture of MM-DIFFEMBED and its AR comparison. The larger right panel expands the native any-resolution SigLIP-family visual path, two-layer connector, spatial merge, and diffusion-pretrained backbone. Dashed branches show model-specific training-only regularization, while both families share the projection and `MaxSim` retrieval path. The diagram is schematic and does not imply that diffusion pretraining alone guarantees better representations.

3.2 DIFFUSION-PRETRAINED CONTEXTUALIZATION

Let F_Θ denote the multimodal backbone, with trainable parameters collectively denoted by Θ. A vision encoder and connector map a page to visual embeddings $\mathbf{V}_d \in \mathbb{R}^{p_d \times h}$, where p_d is the number of visual tokens after image tiling and feature merging and h is the language backbone's hidden width. These embeddings are combined with a fixed retrieval prompt to form the backbone input. Text queries use their tokenized text and the processor's query-augmentation tokens. We retain the pretrained input-processing conventions of each backbone.

For an input of ℓ valid positions, an AR decoder normally uses the attention support $\mathcal{A}_{\mathrm{AR}}(a) = \{1, \ldots, a\}$ at position a. Our retrieval encoders instead use

$$\mathcal{A}_{\mathrm{bi}}(a) = \{1, \ldots, \ell\}, \qquad a \in \{1, \ldots, \ell\}, \tag{1}$$

excluding padding. This permits each language-layer token representation to incorporate evidence from either side of the observed sequence. It does not make a page representation query-conditioned: query–page interaction occurs only after independent encoding.

Two instantiations. COLLAVIDA uses LaViDa-L, comprising a SigLIP vision encoder, an MLP connector, and the LLaDA-8B language backbone (Zhai et al., 2023; Nie et al., 2025; Li et al., 2025). COLSDAR-VL uses SDAR-VL-8B with its SigLIP-2 visual frontend (Tschannen et al., 2025; Cheng et al., 2025b). Although SDAR-VL is pretrained for block-wise diffusion, we repurpose its bidirectional encoding path over the complete observed input. We do not perform block-wise generation or iterative reverse diffusion when extracting the main model's embeddings. Thus, the shared design is *diffusion-pretrained, single-pass contextualization*.

3.3 Token Readout and Late Interaction

The backbone produces hidden-state matrices $\mathbf{H}_q \in \mathbb{R}^{m\times h}$ and $\mathbf{H}_d \in \mathbb{R}^{n\times h}$, where m and n count retained query and page positions. Both main configurations retain all non-padding positions, including visual, prompt, and augmentation positions where applicable. A shared affine projection uses $\mathbf{W} \in \mathbb{R}^{h\times r}$ and bias $\mathbf{B} \in \mathbb{R}^r$ to obtain r-dimensional retrieval vectors:

$$\mathbf{Q}_{a:} = \frac{\mathbf{H}_{q,a:}\mathbf{W} + \mathbf{B}^\top}{\|\mathbf{H}_{q,a:}\mathbf{W} + \mathbf{B}^\top\|_2}, \qquad \mathbf{D}_{b:} = \frac{\mathbf{H}_{d,b:}\mathbf{W} + \mathbf{B}^\top}{\|\mathbf{H}_{d,b:}\mathbf{W} + \mathbf{B}^\top\|_2}, \tag{2}$$

where $\mathbf{Q} \in \mathbb{R}^{m\times r}$ and $\mathbf{D} \in \mathbb{R}^{n\times r}$ denote the query and document embedding matrices. Numerical normalization uses a small positive denominator floor. We set $h = 4096$ and $r = 128$ for both diffusion instantiations. Following late-interaction retrieval (Khattab & Zaharia, 2020; Faysse et al., 2025), the ranking score is

$$s(q,d) = \sum_{a=1}^{m} \max_{1\le b\le n} \langle \mathbf{Q}_{a:}, \mathbf{D}_{b:} \rangle. \tag{3}$$

The maximum identifies the strongest page evidence for each query vector; the sum aggregates these matches. Any contextual reasoning must already be encoded in the vectors, because this scorer introduces no cross-encoder.

Alternative readouts. To study representation granularity, the implementation also supports mean pooling into one vector and readout from k appended mask positions. A refinement variant repeatedly updates those mask positions before extracting their representations, using either argmax or Gumbel-based token selection for intermediate updates. Here k is the output-position budget, distinct from the number of input visual tokens p_d. These are experimental alternatives, not additional stages of the main full-token pipeline. Figure 5 illustrates the four readout designs compared in Section 4.3.

3.4 Retrieval Adaptation Objectives

Contrastive retrieval. For a batch of B_{tr} paired examples, let $s_{ij} = s(q_i, d_j^+)$ and let m_i be the number of valid query positions in example i. Query-length-normalized scores define a query-to-document objective

$$\mathcal{L}_{q\to d} = -\frac{1}{B_{\text{tr}}} \sum_{i=1}^{B_{\text{tr}}} \log \frac{\exp(s_{ii}/(m_i\tau))}{\sum_{j=1}^{B_{\text{tr}}} \exp(s_{ij}/(m_i\tau))}, \tag{4}$$

where $\tau > 0$ is the temperature and other paired pages serve as negatives. Distributed training extends the denominator to the gathered candidate pool. The reverse objective $\mathcal{L}_{d\to q}$ applies the same operation with the document and query token matrices exchanged, normalizing by the source document length. Because `MaxSim` is asymmetric, this is not merely transposing the forward score matrix. The standard symmetric recipe uses $(\mathcal{L}_{q\to d} + \mathcal{L}_{d\to q})/2$.

The cached recipe stores document embeddings across microbatches and re-encodes queries for backpropagation, replacing $\mathbf{D}$ by $\text{sg}(\mathbf{D})$, where sg preserves a value while stopping its gradient. This is a query-side cached optimization recipe inspired by gradient caching (Gao et al., 2021), not an exact two-sided GradCache implementation. Since the encoder parameters are shared, query-side updates still change the document encoding function. The two instantiated recipes are reported separately in Section 4.1; implementation-level padding conventions are given in Appendix C.

Prompt-denoising regularization. We use an auxiliary denoising objective in CoLLaViDa. Let $\mathbf{X}_d \in \mathcal{V}^{t_d}$ denote the t_d discrete prompt tokens, where $\mathcal{V}$ is the backbone vocabulary, and let $\mathcal{M}_d \subseteq \{1,\dots,t_d\}$ contain positions independently selected with masking probability ρ. The corruption operator $C_{\rho,\sigma}$ replaces selected text embeddings with the mask embedding and adds zero-mean Gaussian noise of standard deviation σ to the expanded visual-prefix region. Let $\widetilde{\mathbf{E}}_d \in \mathbb{R}^{\ell_d\times h}$ be the resulting input matrix, where ℓ_d is the expanded input length. Let $p_{\Theta,a}$ be the backbone vocabulary distribution at masked position a, let B_{pd} be the auxiliary microbatch size, and let $Z = \sum_{i=1}^{B_{\text{pd}}} |\mathcal{M}_{d_i}|$ count all masked prompt positions in that microbatch. The auxiliary loss is

$$\mathcal{L}_{\text{pd}} = \mathbb{E}_{C_{\rho,\sigma}} \left[-\frac{1}{Z} \sum_{i=1}^{B_{\text{pd}}} \sum_{a\in\mathcal{M}_{d_i}} \log p_{\Theta,a}(\mathbf{X}_{d_i,a} \mid \widetilde{\mathbf{E}}_{d_i}) \right], \tag{5}$$

with zero loss when $Z = 0$. Visual features are detached in this auxiliary branch. Importantly, the reconstructed text is the fixed prompt *Describe the image.*, not OCR content, a page caption, or a reasoning trace. This is a local adaptation regularizer, not the original full diffusion pretraining objective or a direct reasoning-supervision signal.

Query-vector decorrelation. Late interaction can allocate multiple vectors to nearly the same direction. To discourage excessive average similarity, COLSDAR-VL regularizes the query representations. Let $\overline{\mathbf{Q}}_i \in \mathbb{R}^{\bar{m}\times r}$ be a query matrix zero-padded to the current batch width $\bar{m}$. Its nonzero rows are normalized, and its padding rows remain zero. For target similarity c, the implemented regularizer is

$$\mathcal{L}_{\mathrm{qd}} = \frac{1}{B_{\mathrm{tr}}} \sum_{i=1}^{B_{\mathrm{tr}}} \mathrm{ReLU}\left(\frac{2}{\bar{m}(\bar{m}-1)} \sum_{1\le a<a'\le \bar{m}} \langle \overline{\mathbf{Q}}_{i,a:}, \overline{\mathbf{Q}}_{i,a':} \rangle - c \right). \tag{6}$$

Sequences with fewer than two positions contribute zero. ReLU is applied *after* averaging the pairwise similarities: this penalty constrains their mean, not every individual pair, and does not enforce orthogonality.

Model-specific adaptation. The framework combines a retrieval loss $\mathcal{L}_{\mathrm{ret}}$ with nonnegative auxiliary weights λ_{pd} and λ_{qd}:

$$\mathcal{L} = \mathcal{L}_{\mathrm{ret}} + \lambda_{\mathrm{pd}}\mathcal{L}_{\mathrm{pd}} + \lambda_{\mathrm{qd}}\mathcal{L}_{\mathrm{qd}}. \tag{7}$$

The COLLAVIDA instantiation uses cached retrieval adaptation with prompt denoising; the COLSDAR-VL instantiation uses standard symmetric retrieval adaptation with query decorrelation. We apply low-rank adaptation (Hu et al., 2021) and retain the same token projection and `MaxSim` interface at inference. Auxiliary losses are training-only: neither adds a retrieval-time generation pass. This property does not imply a latency advantage over AR embedding models, which can also encode observed inputs in one pass.

3.5 ENCODING AND TRAINING PROCEDURE

At inference, each page is encoded once and cached; a query is encoded on demand and scored against the cache with `MaxSim`. Training uses the same independent encoders, while the selected recipe determines whether page states are cached with stopped gradients and which auxiliary objective is used. Algorithm 1 in Appendix C.5 gives the complete training, indexing, and retrieval procedure.

3.6 THEORETICAL ANALYSIS

Our analysis isolates three properties that are necessary for the proposed design but do not, by themselves, imply a universal accuracy improvement. First, bidirectional contextualization makes suffix evidence available to every retained position. Second, normalized `MaxSim` has a simple ranking stability guarantee. Third, the query decorrelation term controls only the mean pairwise similarity, which explains both its intended anti-collapse role and its limitations. Formal statements, proof sketches, and computational implications are provided in Appendix D.

Proposition 1 (Context availability)**.** *In a deterministic transformer with a causal mask, the hidden state at position a depends only on positions $1{:}a$. With the full attention support in Eq. 1, it may also depend on positions $a{+}1{:}\ell$. Hence two inputs sharing their prefix through a have identical causal states at a, while their bidirectional states may differ if their suffixes differ.*

Proposition 2 (`MaxSim` stability)**.** *Let $\mathbf{Q}, \mathbf{Q}' \in \mathbb{R}^{m\times r}$ and $\mathbf{D}, \mathbf{D}' \in \mathbb{R}^{n\times r}$ have unit-norm rows. If corresponding rows differ by at most ϵ_q and ϵ_d in Euclidean norm, then*

$$|s(\mathbf{Q}, \mathbf{D}) - s(\mathbf{Q}', \mathbf{D}')| \le m(\epsilon_q + \epsilon_d).$$

A score gap exceeding twice this bound therefore preserves the pairwise ranking under perturbation.

Proposition 3 (Meaning of query decorrelation)**.** *For $m \ge 2$ unit query rows $\{\mathbf{Q}_{a:}\}_{a=1}^{m} \subset \mathbb{R}^r$,*

$$\frac{2}{m(m-1)} \sum_{a<a'} \langle \mathbf{Q}_{a:}, \mathbf{Q}_{a':} \rangle = \frac{\|\sum_a \mathbf{Q}_{a:}\|_2^2 - m}{m(m-1)}.$$

Thus the implemented ReLU penalty reduces a positive mean similarity (equivalently, the query-vector centroid norm), but cannot guarantee that every pair is orthogonal. Zero padding further changes the denominator as specified in Eq. 6.

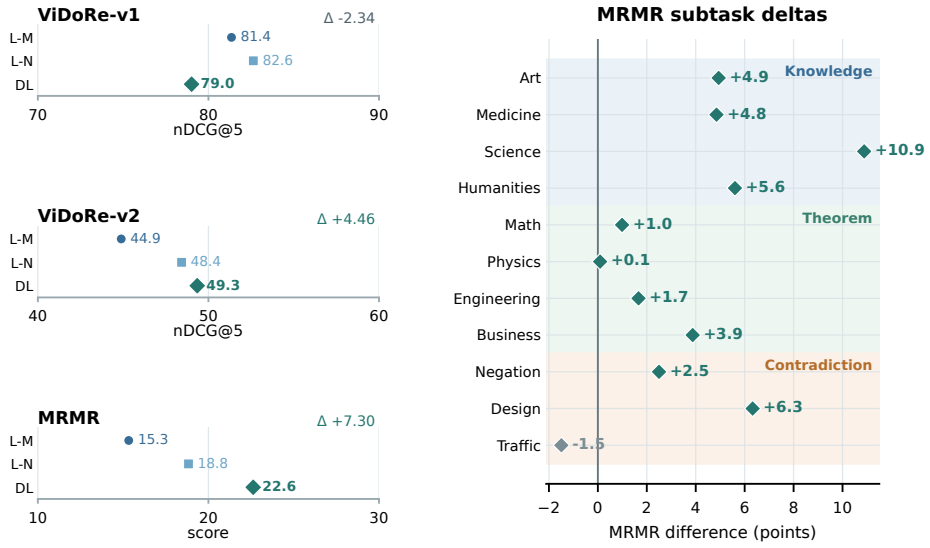


Figure 3: LLaVA-family results. Left: benchmark means (L-M, L-N, and DL denote the two AR baselines and COLLAVIDA). Right: COLLAVIDA's MRMR subtask differences from the strongest AR member in the family. Shaded bands indicate Knowledge, Theorem, and Contradiction. Figure 33 shows the Qwen-family comparison; Table 3 reports all subset scores.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Training Configuration. We train COLLAVIDA and COLSDAR-VL on 118,195 ColPali query–page pairs (Faysse et al., 2025) for five epochs, using AdamW with a learning rate of 2×10^{-4}, rank-64 LoRA, and temperature $\tau = 0.02$. COLLAVIDA combines cached retrieval adaptation with prompt denoising ($\lambda_{\text{pd}} = 0.5$), while COLSDAR-VL uses symmetric retrieval adaptation with query decorrelation ($\lambda_{\text{qd}} = 0.1$). The AR comparisons are ColLLaVA-1.6–Mistral-7B, ColLLaVA-Llama3, ColQwen2-7B, and ColQwen2.5-7B. All experiments run on a cluster of NVIDIA A100-80GB GPUs. Appendix C details training and visual processing.

Evaluation. We report macro nDCG@5 over ten ViDoRe-v1 and four selected ViDoRe-v2 subsets (Faysse et al., 2025; Macé et al., 2025). On MRMR (Zhang et al., 2026), we average nDCG@10 for ten subtasks and Hit@1 for Negation equally across all eleven subtasks. All six models share a modified MRMR protocol: mixed-modal query images are omitted, image-only Negation queries are retained, and retrieval uses the full task corpus, including 800 candidates for Negation rather than four. Appendix C.3 specifies the complete evaluation and aggregation protocol, and Appendix E provides additional ViDoRe metrics.

4.2 MAIN RESULTS

Figure 3 and Table 3 summarize the benchmark and subset results. We use ColLLaVA-Llama3 and ColQwen2, the strongest AR models on MRMR within each family, as references. Across ViDoRe-v1, ViDoRe-v2, and MRMR, COLLAVIDA scores 79.01, 49.34, and 22.63, with changes of -3.62, $+0.91$, and $+3.79$ points relative to ColLLaVA-Llama3. COLSDAR-VL scores 83.05, 52.27, and 29.08, with changes of -7.01, -6.89, and $+4.76$ points relative to ColQwen2. Both diffusion retrievers therefore improve on the reasoning-intensive benchmark, while their ViDoRe results depend on the backbone: COLLAVIDA slightly exceeds ColLLaVA-Llama3 on ViDoRe-v2 but trails it on ViDoRe-v1, whereas ColQwen2 leads COLSDAR-VL on both versions. Figure 33 in Appendix E provides the Qwen-family benchmark means and subtask-level differences, enabling the same comparison across both backbone families.

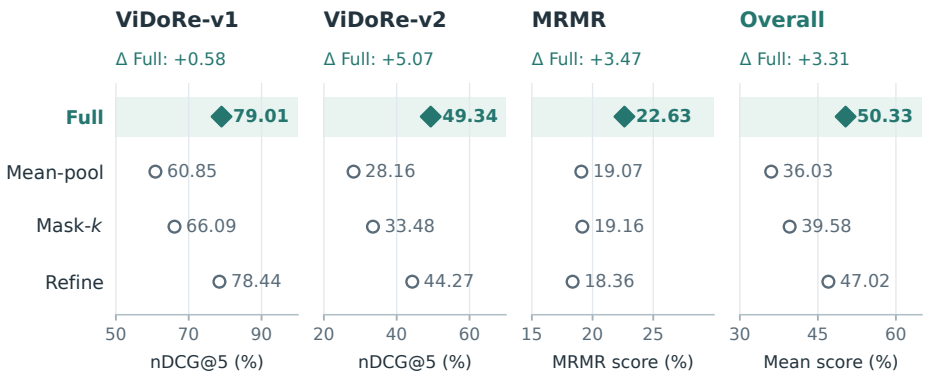


Figure 4: Comparison of four ColLaViDa readout designs. Overall averages the three benchmark scores. Each mode uses one model throughout; Δ is Full minus the strongest displayed alternative. Axes differ; Appendix F details selection, and Section 4.1 specifies the MRMR protocol.

MRMR gains vary across subtasks (Figures 3 and 33). Relative to each subtask's strongest AR member, ColLaViDa improves most on Knowledge: Science (+10.9 points) and Design (+6.3), but trails on Traffic (−1.5). ColSDAR-VL gains most on Design (+27.8), with gains on Medicine (+4.9) and Physics (+3.9), while remaining below the Qwen maximum on Mathematics (−1.8) and Art (−0.5). The largest gains thus arise in knowledge and contradiction tasks. A plausible explanation is that contextualized token vectors preserve complementary local cues that jointly identify relevant evidence. The variation across subtasks also highlights the value of inspecting local performance differences alongside benchmark averages when comparing backbone families.

4.3 Token Readout and Representation Granularity

For page encoding, *Full* retains all valid token states, *mean-pool* averages states into one vector, and *mask-k* and *refine* target fixed mask positions without and with iterative token updates, respectively (Figure 5; Algorithm 2 in Appendix C.6). Figure 4 compares these four designs on the three benchmarks and their equally weighted mean, *Overall*. Full obtains 50.33 Overall, exceeding Refine by 3.31 points, Mask-k by 10.75 points, and Mean-pool by 14.30 points. Relative to the strongest displayed alternative on each benchmark, Full improves ViDoRe-v1, ViDoRe-v2, and MRMR by 0.58, 5.07, and 3.47 points, respectively. Within this comparison, retaining all valid states provides the strongest retrieval performance without either pooling them into a single vector or restricting the output to a fixed set of appended positions.

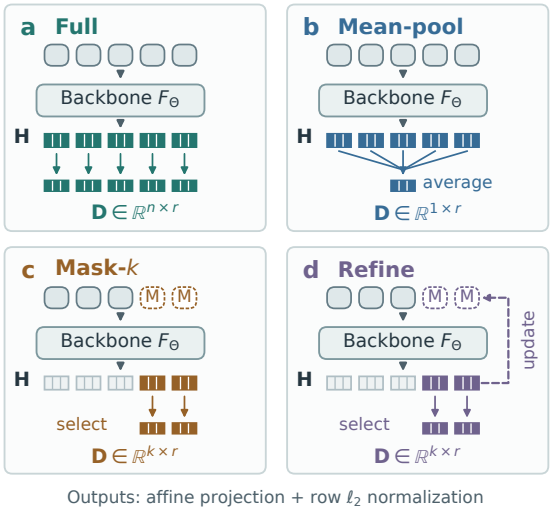


Figure 5: Readouts: **(a)** all states; **(b)** averaging; **(c)** mask slots; **(d)** slot updates. M marks appended masks; outputs use Eq. 2. Appendix F details readout configurations.

Mean-pool trails Full by 18.16 and 21.19 points on the two ViDoRe versions. This pattern is consistent with the value of preserving independently matchable local evidence: pooling removes the token-level matching choices available to `MaxSim` (Khattab & Zaharia, 2020; Faysse et al., 2025). Mask-k also falls below Full on all three benchmarks, with an Overall gap of 10.75 points. Full exceeds Refine by 0.58/5.07/4.27 points on ViDoRe-v1/ViDoRe-v2/MRMR without iterative token updates, suggesting that repeated token prediction does not necessarily improve retrieval-oriented representations. Appendix F details candidate selection and comparison settings.

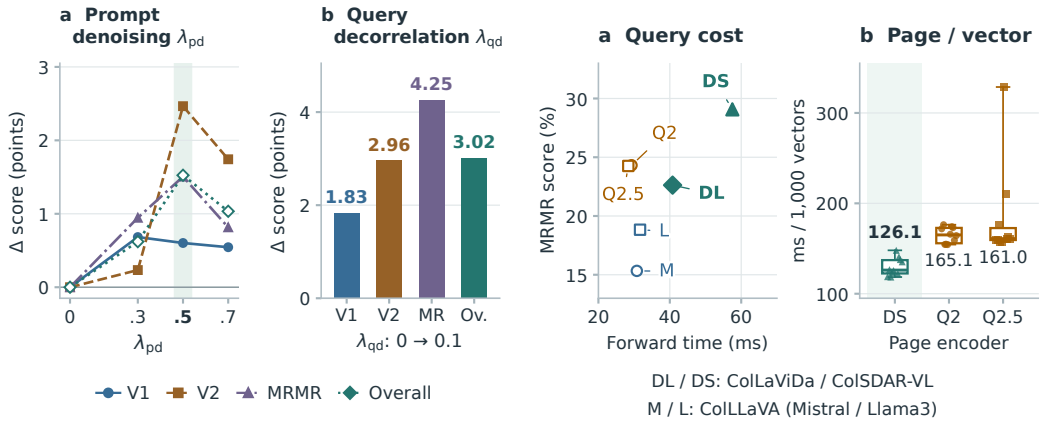


Figure 6: Hyperparameter analysis across benchmarks. **(a)** ColLaViDa: 0.5 is shaded. **(b)** ColSDAR-VL: 0 to 0.1. Changes are relative to zero weight (points). V1/V2 denote ViDoRe-v1/v2; MR is MRMR; Ov. is the three-benchmark mean. Appendix F.2 details the comparison conventions.

Figure 7: Efficiency analysis. **(a)** MRMR quality versus median ArxivQA query-forward time. **(b)** Page cost per 1,000 vectors: quartiles, medians, full ranges, and ten observations per model. Q2/Q2.5 denote ColQwen2/2.5; Appendix G provides absolute costs and full distributions for all models.

4.4 Hyperparameter Analysis

Figure 6 varies ColLaViDa's prompt-denoising weight λ_{pd} and ColSDAR-VL's query-decorrelation weight λ_{qd} (Eq. 7). For ColLaViDa, 0.5 maximizes Overall among the tested weights (50.33 versus 48.80 at zero), improving ViDoRe-v2 and MRMR by 2.47 and 1.51 points. ViDoRe-v1 instead peaks at 0.3, only 0.08 points above 0.5, while 0.7 lowers Overall to 49.84. This nonmonotonic response suggests a balance between retrieval adaptation and prompt reconstruction. For ColSDAR-VL, changing λ_{qd} from 0 to 0.1 raises Overall from 51.78 to 54.80, with ViDoRe-v1/v2 and MRMR gains of 1.83/2.96/4.25 points. These gains are consistent with discouraging redundant query features during multi-vector matching. Appendix F.2 specifies the coefficient values and benchmark aggregation conventions used in these comparisons.

4.5 Efficiency Analysis

Figure 7 pairs MRMR quality with ArxivQA encoding times on one A100-80GB GPU. Median query-forward times are 40.8 and 57.5 ms for ColLaViDa and ColSDAR-VL, adding 9.1 and 28.2 ms relative to ColLLaVA-Llama3 and ColQwen2. All six models use one encoding pass. For page encoding, ColSDAR-VL uses a median of 126.1 ms per 1,000 retained vectors, versus 165.1 and 161.0 ms for ColQwen2 and ColQwen2.5, reductions of 23.6% and 21.6%. This output-normalized advantage accompanies a denser representation: 4,792 vectors per page at the median, versus 755 for both AR models. Its absolute page-forward time is higher, at 594.8 ms versus 123.3 and 121.7 ms; ColLaViDa also incurs a higher offline cost. These measurements characterize the native token budgets and attention backends. Appendix G details absolute costs, memory, storage, batching, and scoring for all six models.

5 Conclusion

We presented MM-DiffEmbed, a single-pass multi-vector MDR framework with full-token readout and model-specific regularization. ColLaViDa and ColSDAR-VL improve reasoning-intensive MRMR retrieval over their AR baselines, with backbone-dependent trade-offs on ViDoRe. Our analyses support retaining contextualized token evidence and balancing the auxiliary objectives. Efficiency measurements identify lower per-vector page cost for ColSDAR-VL than for Qwen AR models, alongside higher absolute latency. These findings motivate diffusion pretraining as a source of complementary local evidence for complex retrieval. Appendices H, I, J, and K cover limitations and outlook, AI use, ethics, and reproducibility.

REFERENCES

Abdelrahman Abdallah, Mahmoud Abdalla, Mohammed Ali, and Adam Jatowt. Argus-retriever: Vision-llm late-interaction retrieval with region-aware query-conditioned moe for visual document retrieval, 2026. URL https://arxiv.org/abs/2606.04300.

Duong Bach. Hierarchical patch compression for colpali: Efficient multi-vector document retrieval with dynamic pruning and quantization, 2025. URL https://arxiv.org/abs/2506.21601.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report, 2025. URL https://arxiv.org/abs/2502.13923.

Yanqi Chen, Juelin Liu, Alexandra Meliou, and Xiao Yan. Chimera: Efficient multi-vector retrieval via gpu-cpu co-processing, 2026. URL https://arxiv.org/abs/2608.23553.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and Bowen Zhou. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation, 2025a. URL https://arxiv.org/abs/2510.06303.

Shuang Cheng, Yuhua Jiang, Zineng Zhou, Dawei Liu, Wang Tao, Linfeng Zhang, Biqing Qi, and Bowen Zhou. SDAR-VL: Stable and Efficient Block-wise Diffusion for Vision-Language Understanding, 2025b. URL https://arxiv.org/abs/2512.14068.

Gabriel de Souza P. Moreira, Ronay Ak, Mengyao Xu, Oliver Holworthy, Benedikt Schifferer, Zhiding Yu, Yauhen Babakhin, Radek Osmulski, Jiarui Cai, Ryan Chesler, Bo Liu, and Even Oldridge. Nemotron colembed v2: Top-performing late interaction embedding models for visual document retrieval, 2026. URL https://arxiv.org/abs/2602.03992.

DiffusionGemma Team, Adrien Ali Taïga, James Assiene, Daniele Calandriello, Rahma Chaabouni, João Gante, Tamara von Glehn, Nate Keating, Chris Knutsen, Martin Kukla, Tianlin Liu, Ivan Lobov, Ofir Nabati, João Gabriel Oliveira, Nicolas Perez-Nieves, Nastasia Prutianova, Bobak Shahriari, Jean Tarbouriech, Pavel Tyletski, Çağlar Ünlü, Cindy Wu, Glenn Cameron, Jerome Connor, Sertan Girgin, Maarten Grootendorst, Alon Levkovitch, Eliya Nachmani, Omar Sanseviero, Piotr Stanczyk, Quentin Berthet, Andrew Campbell, Clément Crepy, Valentin De Bortoli, Arnaud Doucet, Romuald Elie, Alexandre Galashov, Klaus Greff, Alexis Jacq, David Ruhe, Yu-Han Wu, Sebastian Flennerhag, Brendan O'Donoghue, George Scrivener, and Shantanu Thakoor. Diffusiongemma technical report, 2026. URL https://arxiv.org/abs/2608.00146.

Yaoxuan Dou and Yang Shu. Faster but different: Diagnosing and controlling content drift in accelerated multimodal diffusion language models, 2026. URL https://arxiv.org/abs/2607.29079.

Sedigheh Eslami, Maksim Gaiduk, Markus Krimmel, Louis Milliken, Bo Wang, and Denis Bykov. Diffusion-Pretrained Dense and Contextual Embeddings. *arXiv preprint arXiv:2602.11151*, 2026. URL https://arxiv.org/abs/2602.11151.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. ColPali: Efficient Document Retrieval with Vision Language Models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 61424–61449, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/99e9e141aafc314f76b0ca3dd66898b3-Paper-Conference.pdf.

Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model, 2025. URL https://arxiv.org/abs/2502.09622.

Yonggan Fu, Lexington Whalen, Zhifan Ye, Xin Dong, Shizhe Diao, Jingyu Liu, Chengyue Wu, Hao Zhang, Enze Xie, Song Han, Maksim Khadkevich, Jan Kautz, Yingyan Celine Lin, and Pavlo Molchanov. Efficient-dlm: From autoregressive to diffusion language models, and beyond in speed, 2026. URL https://arxiv.org/abs/2512.14067.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. Scaling deep contrastive learning batch size under memory limited setup. In Anna Rogers, Iacer Calixto, Ivan Vulić, Naomi Saphra, Nora Kassner, Oana-Maria Camburu, Trapit Bansal, and Vered Shwartz (eds.), *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pp. 316–321, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.repl4nlp-1.31. URL https://aclanthology.org/2021.repl4nlp-1.31/.

Tiancheng Gu, Kaicheng Yang, Kaichen Zhang, Xiang An, Ziyong Feng, Yueyi Zhang, Weidong Cai, Jiankang Deng, and Lidong Bing. Unime-v2: Mllm-as-a-judge for universal multimodal embedding learning, 2025. URL https://arxiv.org/abs/2510.13515.

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-embeddings-v4: Universal Embeddings for Multimodal Multilingual Retrieval, 2025. URL https://arxiv.org/abs/2506.18902.

Lixuan Guo, Yifei Wang, Tiansheng Wen, Aosong Feng, Stefanie Jegelka, and Chenyu You. No more k-means: Single-stage sparse coding for efficient multi-vector retrieval, 2026. URL https://arxiv.org/abs/2605.30120.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Zhanqiu Hu, Jian Meng, Yash Akhauri, Mohamed S. Abdelfattah, Jae sun Seo, Zhiru Zhang, and Udit Gupta. Flashdlm: Accelerating diffusion language model inference via efficient kv caching and guided diffusion, 2025. URL https://arxiv.org/abs/2505.21467.

Jiahao Huo, Yu Huang, Yibo Yan, Ye Pan, Kening Zheng, Wei-Chieh Huang, Yi Cao, Mingdong Ou, Philip S. Yu, and Xuming Hu. Causalembed: Auto-regressive multi-vector generation in latent space for visual document embedding, 2026. URL https://arxiv.org/abs/2601.21262.

Rajesh Jayaram. Multi-vector embeddings are provably more expressive than single vector embeddings, 2026. URL https://arxiv.org/abs/2606.23475.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training Vision-Language Models for Massive Multimodal Embedding Tasks. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 1255–1279, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/04261fce1705c4f02f062866717d592a-Paper-Conference.pdf.

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*, SIGIR '20, pp. 39–48. ACM, July 2020. doi: 10.1145/3397271.3401075. URL http://dx.doi.org/10.1145/3397271.3401075.

Jaeik Kim, Woojin Kim, Jihwan Hong, Yejoon Lee, Sieun Hyeon, Mintaek Lim, Yunseok Han, Dogeun Kim, Hoeun Lee, Hyunggeun Kim, and Jaeyoung Do. Dynin-omni: Omnimodal unified large diffusion language model, 2026a. URL https://arxiv.org/abs/2604.00007.

Juyeon Kim, Geon Lee, Dongwon Choi, Taeuk Kim, and Kijung Shin. Hybrid-vector retrieval for visually rich documents: Combining single-vector efficiency and multi-vector accuracy, 2026b. URL https://arxiv.org/abs/2510.22215.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy Visual Task Transfer, 2024. URL https://arxiv.org/abs/2408.03326.

Lijiang Li, Zuwei Long, Yunhang Shen, Heting Gao, Haoyu Cao, Xing Sun, Caifeng Shan, Ran He, and Chaoyou Fu. Omni-diffusion: Unified multimodal understanding and generation with masked discrete diffusion, 2026a. URL https://arxiv.org/abs/2603.06577.

Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. LaViDa: A Large Diffusion Language Model for Multimodal Understanding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 105101–105134. Curran Associates, Inc., 2025. doi: 10.52202/085713-3511. URL https://doi.org/10.52202/085713-3511.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. Lavida-o: Elastic large masked diffusion models for unified multimodal understanding and generation, 2026b. URL https://arxiv.org/abs/2509.19244.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. Sparse-lavida: Sparse multimodal discrete diffusion language models, 2026c. URL https://arxiv.org/abs/2512.14008.

Shufan Li, Yuchen Zhu, Jiuxiang Gu, Kangning Liu, Zhe Lin, Yongxin Chen, Molei Tao, Aditya Grover, and Jason Kuen. Lavida-r1: Advancing reasoning for unified multimodal diffusion language models, 2026d. URL https://arxiv.org/abs/2602.14147.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023. doi: 10.52202/075280-1516. URL https://doi.org/10.52202/075280-1516.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, June 2024.

Xinze Liu, Lei Yang, Dayan Wu, Hengjie Zhu, Zihao Zhang, Hanqi Wu, Tianzhu Hu, Peng Fu, Zheng Lin, and Weiping Wang. Adaptiveembed: Sample-adaptive multi-vector representation for multimodal retrieval, 2026a. URL https://arxiv.org/abs/2608.25412.

Yaoyang Liu, Junlin Li, Yinjun Wu, and Zhen Chen. Poqd: Performance-oriented query decomposer for multi-vector retrieval, 2025. URL https://arxiv.org/abs/2505.19189.

Zhuchenyang Liu, Ziyu Hu, Yao Zhang, and Yu Xiao. Structural anchor pruning: Training-free multi-vector compression for visual document retrieval, 2026b. URL https://arxiv.org/abs/2601.20107.

Xu Ma, Yitian Zhang, and Yun Fu. Towards unified multimodal large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37212–37230. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1853.

Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. Unifying multimodal retrieval via document screenshot embedding. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6492–6505, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.373. URL https://aclanthology.org/2024.emnlp-main.373/.

Yuxin Ma, Lun Du, Lanning Wei, Kun Chen, Qian Xu, Kangyu Wang, Guofeng Feng, Guoshan Lu, Lin Liu, Xiaojing Qi, Xinyuan Zhang, Zhen Tao, Haibo Feng, Ziyun Jiang, Ying Xu, Zenan Huang, Yihong Zhuang, Haokai Xu, Jiaqi Hu, Zhenzhong Lan, Junbo Zhao, Jianguo Li, and Da Zheng. dinfer: An efficient inference framework for diffusion language models, 2025. URL https://arxiv.org/abs/2510.08666.

Quentin Macé, António Loison, and Manuel Faysse. ViDoRe Benchmark V2: Raising the Bar for Visual Retrieval, 2025. URL https://arxiv.org/abs/2505.17166.

Debanjan Mahata, Atharva Tendle, Daniel Preotiuc-Pietro, Yong Zhuang, and Ozan Irsoy. Midr: Enrichment-augmented indexing for multimodal document retrieval, 2026. URL https://arxiv.org/abs/2609.01316.

Ailar Mahdizadeh, Aria Salari, Sohail Rajabi, Shahriar Mirabbasi, Panos Nasiopoulos, and Alireza Morsali. Coverage matters: Marginmerge for compressing multi-vector visual document retrievers, 2026. URL https://arxiv.org/abs/2608.02969.

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, Yingbo Zhou, Wenhu Chen, and Semih Yavuz. Vlm2vec-v2: Advancing multimodal embedding for videos, images, and visual documents, 2025. URL https://arxiv.org/abs/2507.04590.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large Language Diffusion Models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 50608–50646. Curran Associates, Inc., 2025. doi: 10.52202/085713-1689. URL https://doi.org/10.52202/085713-1689.

Chunyi Peng, Zhipeng Xu, Yukun Yan, Zhenghao Liu, Shi Yu, Sen Mei, Yubo Sun, Yongheng Zhang, Jie Zhou, Yu Gu, Ge Yu, and Maosong Sun. Conceptformer: Learning adaptive latent concepts for query-document alignment in visual document retrieval, 2026. URL https://arxiv.org/abs/2608.15698.

Hanxiang Qin, Alexander Martin, Rohan Jha, Chunsheng Zuo, Reno Kriz, and Benjamin Van Durme. Multi-vector index compression in any modality, 2026. URL https://arxiv.org/abs/2602.21202.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Litu Rout, Constantine Caramanis, and Sanjay Shakkottai. Anchored diffusion language model, 2025. URL https://arxiv.org/abs/2505.18456.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.272. URL https://aclanthology.org/2022.naacl-main.272/.

Jan Luca Scheerer, Matei Zaharia, Christopher Potts, Gustavo Alonso, and Omar Khattab. Warp: An efficient engine for multi-vector retrieval, 2025. URL https://arxiv.org/abs/2501.17788.

Qingyu Shi, Jinbin Bai, Zhuoran Zhao, Wenhao Chai, Kaidong Yu, Jianzong Wu, Yunhai Tong, Xiangtai Li, Xuelong Li, and Shuicheng Yan. Muddit: Liberating generation beyond text-to-image with a unified discrete diffusion model, 2026. URL https://arxiv.org/abs/2505.23606.

Yingte Shu, Yuchuan Tian, Chao Xu, Yunhe Wang, and Hanting Chen. Deferred commitment decoding for diffusion language models, 2026. URL https://arxiv.org/abs/2601.02076.

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

Yueyi Sun, Yuhao Wang, Jason Li, Ye Tian, Tao Zhang, Jacky Mai, Yihan Wang, Haochen Wang, Jinbin Bai, Ling Yang, and Yunhai Tong. Perceptiondlm: Parallel region perception with multimodal diffusion language models, 2026. URL https://arxiv.org/abs/2606.19534.

Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. Tess 2: A large-scale generalist diffusion language model, 2025. URL https://arxiv.org/abs/2502.13917.

Yao Tian, Zhoujin Tian, Xi Zhao, Ruiyuan Zhang, and Xiaofang Zhou. Gem: A native graph-based index for multi-vector retrieval, 2026a. URL https://arxiv.org/abs/2603.20336.

Ye Tian, Ling Yang, Jiongfan Yang, Anran Wang, Yu Tian, Jiani Zheng, Haochen Wang, Zhiyang Teng, Zhuochen Wang, Yinjie Wang, Yunhai Tong, Mengdi Wang, and Xiangtai Li. Mmada-parallel: Multimodal large diffusion language models for thinking-aware editing and generation, 2025. URL https://arxiv.org/abs/2511.09611.

Zeyue Tian, Binxin Yang, Zhaoyang Liu, Jiexuan Zhang, Ruibin Yuan, Hubery Yin, Qifeng Chen, Chen Li, Jing Lyu, Wei Xue, and Yike Guo. Audio-omni: Extending multi-modal understanding to versatile audio generation and editing, 2026b. URL https://arxiv.org/abs/2604.10708.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features, 2025. URL https://arxiv.org/abs/2502.14786.

João Veneroso, Rajesh Jayaram, Jinmeng Rao, Gustavo Hernández Ábrego, Majid Hadian, and Daniel Cer. Crisp: Clustering multi-vector representations for denoising and pruning, 2025. URL https://arxiv.org/abs/2505.11471.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution, 2024. URL https://arxiv.org/abs/2409.12191.

Shuai Wang, Yu Yin, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. DiffRetriever: Parallel Representative Tokens for Retrieval with Diffusion Language Models. *arXiv preprint arXiv:2605.07210*, 2026a.

Xiao Wang, Sean MacAvaney, and Craig Macdonald. Plaid-prf: Pseudo-relevance feedback with centroid-like tokens in plaid, 2026b. URL https://arxiv.org/abs/2607.18626.

Yuchi Wang, Dingkang Yang, Haiyang Yu, Weikang Bian, Jiefeng Long, Xiao Liang, Chao Feng, and Hongsheng Li. Mmemb-r1: Reasoning-enhanced multimodal embedding with pair-aware selection and adaptive control, 2026c. URL https://arxiv.org/abs/2604.06156.

Zihang Wang, Siyue Zhang, Yilun Zhao, Jingyi Yang, Tingyu Song, Anh Tuan Luu, and Chen Zhao. Analyzing Diffusion and Autoregressive Vision Language Models in Multimodal Embedding Space, 2026d. URL https://arxiv.org/abs/2602.06056.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S. Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 2247–2256. IEEE, 2023. doi: 10.1109/BigData59044.2023.10386743.

Peixi Wu, Biao Yang, Feipeng Ma, Bosong Chai, Bo Lin, Wei Yuan, Fan Yang, Tingting Gao, Hebei Li, and Xiaoyan Sun. Lame: Learning to think in latent space for multimodal embedding via information bottleneck, 2026. URL https://arxiv.org/abs/2606.13061.

Haowen Xiang, Yibo Yan, Jiahao Huo, Yu Huang, Yi Cao, Mingdong Ou, and Xuming Hu. Mmmatryoshka: Towards budget-elastic visual document retrieval via a 2d multimodal matryoshka training framework, 2026. URL https://arxiv.org/abs/2606.07654.

Han Xiao. Embedding inversion via conditional masked diffusion language models, 2026. URL https://arxiv.org/abs/2602.11047.

Zhihui Xie, Jiacheng Ye, Lin Zheng, Jiahui Gao, Jingwei Dong, Zirui Wu, Xueliang Zhao, Shansan Gong, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream-coder 7b: An open diffusion language model for code, 2025. URL https://arxiv.org/abs/2509.01142.

Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, Jinbin Bai, Qian Yu, Dengyang Jiang, Yuandong Pu, Haoxing Chen, Le Zhuo, Junjun He, Gen Luo, Tianbin Li, Ming Hu, Jin Ye, Shenglong Ye, Bo Zhang, Chang Xu, Wenhai Wang, Hongsheng Li, Guangtao Zhai, Tianfan Xue, Bin Fu, Xiaohong Liu, Yu Qiao, and Yihao Liu. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding, 2025. URL https://arxiv.org/abs/2510.06308.

Yi Xin, Siqi Luo, Tianxiang Xu, Qi Qin, Haoxing Chen, Kaiwen Zhu, Zhiwei Zhang, Yangfan He, Rongchao Zhang, Jinbin Bai, Shuo Cao, Bin Fu, Junjun He, Yihao Liu, Yuewen Cao, and Xiaohong Liu. dmllm-tts: Self-verified and efficient test-time scaling for diffusion multi-modal large language models, 2026. URL https://arxiv.org/abs/2512.19433.

Jingqi Xu, Jingxi Lu, Chenghao Li, Sreetama Sarkar, Souvik Kundu, and Peter A. Beerel. Redvtp: Training-free acceleration of diffusion vision-language models inference via masked token-guided visual token pruning, 2025a. URL https://arxiv.org/abs/2511.12428.

Mengyao Xu, Gabriel Moreira, Ronay Ak, Radek Osmulski, Yauhen Babakhin, Zhiding Yu, Benedikt Schifferer, and Even Oldridge. Llama Nemoretriever Colembed: Top-Performing Text-Image Retrieval Model, 2025b. URL https://arxiv.org/abs/2507.05513.

Yibo Yan, Guangwei Xu, Xin Zou, Shuliang Liu, James Kwok, and Xuming Hu. Docpruner: A storage-efficient framework for multi-vector visual document retrieval via adaptive patch-level embedding pruning, 2025. URL https://arxiv.org/abs/2509.23883.

Yibo Yan, Jiahao Huo, Guanbo Feng, Mingdong Ou, Yi Cao, Xin Zou, Shuliang Liu, Yuanhuiyi Lyu, Yu Huang, Jungang Li, Kening Zheng, Xu Zheng, Philip S. Yu, James Kwok, and Xuming Hu. Unlocking multimodal document intelligence: From current triumphs to future frontiers of visual document retrieval, 2026a. URL https://arxiv.org/abs/2602.19961.

Yibo Yan, Mingdong Ou, Yi Cao, Jiahao Huo, Xin Zou, Shuliang Liu, James Kwok, and Xuming Hu. Visual late chunking: An empirical study of contextual chunking for efficient visual document retrieval, 2026b. URL https://arxiv.org/abs/2604.10167.

Yibo Yan, Mingdong Ou, Yi Cao, Xin Zou, Jiahao Huo, Shuliang Liu, James Kwok, and Xuming Hu. Sculpting the vector space: Towards efficient multi-vector visual document retrieval via prune-then-merge framework. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 24883–24925. Association for Computational Linguistics, July 2026c. doi: 10.18653/v1/2026.findings-acl.1247. URL https://aclanthology.org/2026.findings-acl.1247/.

Yibo Yan, Mingdong Ou, Yi Cao, Xin Zou, Shuliang Liu, Jiahao Huo, Yu Huang, James Kwok, and Xuming Hu. Beyond the grid: Layout-informed multi-vector retrieval with parsed visual document representations, 2026d. URL https://arxiv.org/abs/2603.01666.

Binhan Yang, Yuxiang Zeng, Hengxin Zhang, Zhuanglin Zheng, Yunzhen Chi, Yongxin Tong, and Ke Xu. Unified and efficient approach for multi-vector similarity search, 2026. URL https://arxiv.org/abs/2604.02815.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models, 2025. URL https://arxiv.org/abs/2505.15809.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025. URL https://arxiv.org/abs/2508.15487.

Ara Yeroyan. Visual rag toolkit: Scaling multi-vector visual retrieval with training-free pooling and multi-stage search, 2026. URL https://arxiv.org/abs/2602.12510.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024. doi: 10.1093/nsr/nwae403.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. VisRAG: Vision-based Retrieval-augmented Generation on Multi-modality Documents. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 21074–21098, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3640a1997a4c9571cea9db2c82e1fc35-Paper-Conference.pdf.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, October 2023.

Siyue Zhang, Yilun Zhao, Liyuan Geng, Arman Cohan, Anh Tuan Luu, and Chen Zhao. Diffusion vs. Autoregressive Language Models: A Text Embedding Perspective, 2025a. URL https://arxiv.org/abs/2505.15045.

Siyue Zhang, Yuan Gao, Xiao Zhou, Yilun Zhao, Tingyu Song, Arman Cohan, Anh Tuan Luu, and Chen Zhao. MRMR: A Realistic and Expert-Level Multidisciplinary Benchmark for Reasoning-Intensive Multimodal Retrieval. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 100517–100555, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a3a7387e49f4de290c23beea2dfcdc75-Paper-Conference.pdf.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Bridging Modalities: Improving Universal Multimodal Retrieval by Multimodal Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9274–9285, June 2025b.

Jiaming Zhou, Xuxin Cheng, Shiwan Zhao, Yuhang Jia, Cao Liu, Ke Zeng, Xunliang Cai, and Yong Qin. Diffa-2: A practical diffusion large language model for general audio understanding, 2026. URL https://arxiv.org/abs/2601.23161.

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, Hongrui Guo, Jiaqi Hu, Wentao Ye, Tieyuan Chen, Chenchen Li, Chengfu Tang, Haibo Feng, Jun Hu, Jun Zhou, Xiaolu Zhang, Zhenzhong Lan, Junbo Zhao, Da Zheng, Chongxuan Li, Jianguo Li, and Ji-Rong Wen. Llada-moe: A sparse moe diffusion language model, 2025. URL https://arxiv.org/abs/2509.24389.

Fengqi Zhu, Shaoxuan Xu, Jingyang Ou, Zebin You, Yipeng Xing, Huabin Liu, Xiaolu Zhang, Jun Zhou, Zhenzhong Lan, Yankai Lin, Wayne Xin Zhao, Jianguo Li, Chongxuan Li, and Ji-Rong Wen. Llada moe v2: Scaling mixture-of-experts diffusion language models, 2026a. URL https://arxiv.org/abs/2608.03457.

Yuhan Zhu, Xiangyu Zeng, Chenting Wang, Xinhao Li, Chunxu Liu, Yicheng Xu, Ziang Yan, Yi Wang, and Limin Wang. Freeret: Mllms as training-free retrievers, 2026b. URL https://arxiv.org/abs/2509.24621.

Haoyu Zuo, Yibo Yan, Xin Zou, Shuliang Liu, Yi Cao, Mingdong Ou, and Xuming Hu. Anchorfold: A focus-then-fold framework via recursive attention propagation for efficient multi-vector visual document retrieval, 2026. URL https://arxiv.org/abs/2608.08732.

APPENDIX CONTENTS

The supplementary material follows the order of its first citation in the main text. Section titles and page numbers below link to their destinations.

A More Related Works 18

B More Illustrative Examples and Task Prompts 21

MRMR task prompts for text-to-page retrieval 21

ViDoRe-v1 examples 22

ViDoRe-v2 examples 27

MRMR examples 29

C Implementation and Reproducibility Details 34

Training data, models, and optimization 34

Visual processing and implementation 35

Evaluation protocol and reporting 35

Source-code map 36

Training algorithm 37

Four readout algorithms 38

D More Theoretical Analysis 40

E Additional Retrieval Results 42

Benchmark-level and subtask results 42

Additional ViDoRe retrieval metrics 44

F Readout and Adaptation Configurations 46

G Additional Efficiency Results and Protocol 47

H Limitations and Outlook 51

I AI Use Statement 51

J Ethics Statement 51

K Reproducibility Statement 51

A More Related Works

This appendix broadens the survey in Section 2, emphasizing work released from 2025 onward while retaining the foundational formulations. We use "multi-vector" for an encoder that stores a set of vectors per input and aggregates their similarities at query time; this term includes token, patch, latent, and representative-vector designs.

A.1 Multi-Vector Retrieval

From contextualized text tokens to general set scoring. ColBERT (Khattab & Zaharia, 2020) established the token-level formulation for text retrieval. For a query matrix $\mathbf{Q} \in \mathbb{R}^{m\times r}$ and document matrix $\mathbf{D} \in \mathbb{R}^{n\times r}$, the interaction matrix is $\mathbf{S} = \mathbf{Q}\mathbf{D}^\top \in \mathbb{R}^{m\times n}$ and the score is

$$s(q,d) = \sum_{a=1}^{m} \max_{1\le b\le n} \mathbf{S}_{ab}. \tag{8}$$

ColBERTv2 (Santhanam et al., 2022) improved supervision, residual compression, and indexing. Recent work studies the same set-to-set interface more formally: multi-vector embeddings can express similarity functions that a single inner product cannot represent at the same dimension (Jayaram, 2026). CausalEmbed (Huo et al., 2026) instead generates a compact sequence of document vectors autoregressively in a latent space, illustrating that vector count, not only backbone size, is an explicit design variable.

Visual documents and multimodal inputs. ColPali (Faysse et al., 2025) transferred Eq. 8 to rendered pages, retaining patch-aligned vectors so that tables, typography, and spatial relations need not be serialized by OCR. ColQwen2/2.5, Jina-embeddings-v4, Llama Nemoretriever ColEmbed, and VLM2Vec provide further VLM-based retrieval systems (Xu et al., 2025b; Günther et al., 2025; Jiang et al., 2025). VLM2Vec-V2 broadens this line to a single training recipe and benchmark spanning images, videos, and visual documents, while UniME-V2 uses MLLM-judged soft labels to mine harder and less ambiguous negatives (Meng et al., 2025; Gu et al., 2025). Nemotron ColEmbed V2 reports a stronger ColPali-style retriever through cluster-based sampling, hard-negative mining, bidirectional attention, and model merging, while Argus-Retriever adds query-conditioned region experts to focus computation on relevant page areas (de Souza P. Moreira et al., 2026; Abdallah et al., 2026). Training-free baselines such as FreeRet probe whether an MLLM can be used as a retriever without contrastive adaptation, whereas reasoning-oriented embedding models such as MMEmb-R1 and LaME introduce explicit pair selection or latent deliberation before producing a compact representation (Zhu et al., 2026b; Wang et al., 2026c; Wu et al., 2026). Recent document-specific studies target the evidence mismatch directly: POQD (Liu et al., 2025) optimizes query decomposition for multi-vector ranking, ConceptFormer (Peng et al., 2026) learns query-conditioned latent concepts, and AdaptiveEmbed (Liu et al., 2026a) allocates a content-adaptive number of vectors rather than using one fixed budget for every page. ColChunk studies contextual late chunking to reduce the number of stored visual vectors while preserving page-level context (Yan et al., 2026b). These methods complement, rather than replace, the independent page and query encoding used by MM-DiffEmbed.

Compression and adaptive representation budgets. The index for N pages stores $r\sum_{j=1}^{N} n_j$ scalars, and exact scoring costs $O(mr\sum_j n_j)$ per query. A generic document-side compressor can be written as $\mathcal{C} : \mathbb{R}^{n\times r} \to \mathbb{R}^{k\times r}$ with $k \ll n$, where quality depends on preserving query-relevant coverage. HPC-ColPali combines quantization and attention-guided pruning (Bach, 2025); DocPruner (Yan et al., 2025) performs adaptive patch-level pruning; CRISP clusters representations before denoising and pruning (Veneroso et al., 2025); and Structural Anchor Pruning (Liu et al., 2026b) provides a training-free, query-agnostic compression diagnostic. MarginMerge (Mahdizadeh et al., 2026) and AnchorFold (Zuo et al., 2026) use coverage-aware anchors or recursive attention propagation for visual documents. Prune-then-Merge explicitly separates token selection from vector fusion, while MM-Matryoshka trains representations that remain useful across multiple vector and dimension budgets (Yan et al., 2026c; Xiang et al., 2026). Beyond the Grid (ColParse) replaces a dense patch grid with parsed, layout-informed regions plus a global vector (Yan et al., 2026d). Visual RAG Toolkit applies training-free spatial pooling and staged search, and HEAVEN combines a single-vector first stage with exact multi-vector reranking (Yeroyan, 2026; Kim et al., 2026b). Multi-Vector

Index Compression in Any Modality (Qin et al., 2026) studies resizing, memory tokens, hierarchical pooling, and attention-guided clustering under a fixed budget, while No More K-means (Guo et al., 2026) explores sparse coding as an alternative to large-scale clustering.

Systems and retrieval-time acceleration. Compression is only one part of deployment. WARP (Scheerer et al., 2025) combines dynamic similarity imputation, implicit decompression, and staged score reduction for fast multi-vector search. MV-HNSW (Yang et al., 2026) and GEM (Tian et al., 2026a) construct native graph indexes instead of filtering with a single-vector proxy; Chimera (Chen et al., 2026) co-processes vector data between GPU and CPU memory. PLAID-PRF augments candidate scoring with pseudo-relevance centroid tokens, whereas UEmbed jointly exposes sparse and dense multimodal signals (Wang et al., 2026b; Song et al., 2026). MIDR studies enrichment-augmented indexing for multimodal documents, highlighting that index construction and representation learning are coupled design choices (Mahata et al., 2026). These systems expose a quality–storage–latency frontier rather than a free efficiency gain. Our experiments intentionally keep the full-token interface fixed, so the measured differences concern contextualization and adaptation, not post-hoc index compression. Taken together, the 2025–2026 literature is converging on three deployment patterns: retain dense patch evidence for exact reranking, learn or select a small representative set for candidate generation, and let the index choose the representation granularity. This separation is useful for interpreting our results: MM-DIFFEMBED changes the contextual encoder and its retrieval adaptation, while leaving the stored vector budget and exact aggregation unchanged. The recent MDR survey provides a broader taxonomy of representation, indexing, and benchmark-design choices (Yan et al., 2026a); this appendix focuses on the subset most directly connected to multi-vector encoding and diffusion contextualization.

A.2 DIFFUSION LANGUAGE MODELS

Text diffusion and representation learning. Autoregressive language modeling factorizes $p(x_{1:\ell}) = \prod_{a=1}^{\ell} p(x_a \mid x_{<a})$, whereas a masked diffusion model corrupts an input sequence and predicts selected positions. For $\mathbf{E} \in \mathbb{R}^{\ell \times h}$, mask set $\mathcal{M}$, and corruption $C_{\rho,\sigma}$, a generic objective is

$$\mathcal{L}_{\mathrm{DLM}}(\Theta) = -\mathbb{E}_{\widetilde{\mathbf{E}} = C_{\rho,\sigma}(\mathbf{E}),\, a \in \mathcal{M}} \log p_\Theta(x_a \mid \widetilde{\mathbf{E}}, a). \tag{9}$$

LLaDA (Nie et al., 2025) demonstrated large-scale masked diffusion pretraining. Theoretical analyses clarify when masked diffusion can achieve short-step perplexity gains and where sequence-level error remains difficult (Feng et al., 2025). Anchored Diffusion Language Models protect high-information tokens during corruption (Rout et al., 2025), while TESS 2 shows that careful AR-to-diffusion adaptation and controllable inference can close some instruction-following gaps (Tae et al., 2025). Dream 7B and Dream-Coder (Ye et al., 2025; Xie et al., 2025) explored general and code generation with arbitrary-order denoising, while LLaDA-MoE (Zhu et al., 2025) and LLaDA-MoE v2 (Zhu et al., 2026a) study sparse expert scaling. SDAR converts an AR model into a block-wise diffusion generator (Cheng et al., 2025a); dInfer and Efficient-DLM target the resulting systems and AR-to-diffusion conversion efficiency (Ma et al., 2025; Fu et al., 2026); DiffusionGemma provides an open large-scale engineering reference (DiffusionGemma Team et al., 2026). Text embedding works such as DiffEmbed, DiffRetriever, and pplx-embed (Zhang et al., 2025a; Wang et al., 2026a; Eslami et al., 2026) show that the denoising backbone alone does not determine the readout: pooled, masked-position, and representative-token embeddings make different trade-offs. FlashDLM and Deferred Commitment Decoding reduce repeated computation or avoid boundary-induced context truncation during denoising (Hu et al., 2025; Shu et al., 2026). Embedding Inversion further shows that masked diffusion can decode an embedding-conditioned sequence in parallel (Xiao, 2026).

Multimodal diffusion language models. Multimodal models insert visual tokens before applying a denoising objective. With visual features $\mathbf{V} \in \mathbb{R}^{p \times h}$ and expanded sequence $\mathbf{E}_{\mathrm{mm}} \in \mathbb{R}^{\ell \times h}$, the retrieval-time encoder in our setting is the deterministic map $\mathbf{H} = F_\Theta(\mathbf{E}_{\mathrm{mm}}; \mathcal{A}_{\mathrm{bi}}) \in \mathbb{R}^{\ell \times h}$. MMaDA (Yang et al., 2025) and Muddit (Shi et al., 2026) unify text and image understanding or generation under discrete diffusion. LaViDa, LaViDa-O, and LaViDa-R1 extend masked multimodal modeling from understanding to elastic unified generation and reasoning (Li et al., 2025; 2026b;d); Sparse-LaViDa prunes masked visual tokens during sampling (Li et al., 2026c), and RedVTP uses response-driven visual-token pruning (Xu et al., 2025a). MMaDA- Parallel studies bidirectional cross-modal denoising throughout a generation trajectory (Tian et al., 2025), and PerceptionDLM

explicitly exploits parallel region perception for image understanding (Sun et al., 2026). These methods primarily optimize generation or multimodal reasoning, whereas MM-DIFFEMBED uses their observed-input contextualization as a multi-vector encoder without iterative decoding. The recent PerceptionDLM result is representative of a broader shift from sequence-level generation toward parallel, region- or span-level perception. In that setting, a denoising model exposes a family of contextual states $\{\mathbf{h}_a\}_{a=1}^{\ell}$ rather than a single causal prefix state. Our retrieval adaptation treats these states as evidence vectors and learns their query-document geometry directly, so no claim about generation speed or sampling quality is required for the MDR comparison.

Omni-modal diffusion and scope of this work. The latest direction models more than text and images in one diffusion space. Lumina-DiMOO (Xin et al., 2025), Omni-Diffusion (Li et al., 2026a), and Dynin-Omni (Kim et al., 2026a) target unified text, image, speech, and video understanding or generation. DIFFA-2 extends diffusion language modeling to general audio understanding (Zhou et al., 2026), while Audio-Omni combines a frozen MLLM with a diffusion transformer for audio generation and editing (Tian et al., 2026b). MMaDA-Parallel and dMLLM-TTS further explore parallel multimodal trajectories and test-time scaling (Tian et al., 2025; Xin et al., 2026). Acceleration work such as RedVTP and Faster but Different shows that fewer denoising computations can change content, so speed must be evaluated jointly with representation fidelity (Xu et al., 2025a; Dou & Shu, 2026). A recent study of diffusion and AR VLMs compares single-vector embeddings across 32 datasets (Wang et al., 2026d); it does not evaluate token-level page evidence with `MaxSim`. Our scope is the complementary, narrower question of whether diffusion-pretrained contextual states, adapted with retrieval losses and retained as multi-vectors, are useful for MDR and especially reasoning-intensive retrieval.

B More Illustrative Examples and Task Prompts

This section connects the retrieval task to concrete benchmark inputs. We first summarize MRMR task instructions, then present one publicly released example from every evaluated subset.

B.1 MRMR Task Prompts for Text-to-Page Retrieval

These templates adapt the *Task* fields in MRMR's Appendix G (Zhang et al., 2026) to text-to-page retrieval. Knowledge and Theorem each share an instruction across four subsets; Contradiction uses three distinct criteria. Braces mark supplied query text, and the output is a ranking of page images, not a generated answer. The templates explain task semantics rather than prefixes used in our evaluations (Appendix C.3). The Negation template requires a supplied scene description; the recorded evaluation instead retains the original image query without text substitution.

Knowledge: Art, Medicine, Science, and Humanities

```
Task: Retrieve document-page images containing evidence that helps
answer the question.
Question: {question}
```

Theorem: Mathematics, Physics, Engineering, and Business

```
Task: Retrieve document-page images presenting theorems, laws, or
principles needed to solve the problem.
Problem: {problem}
```

Contradiction: Negation

```
Task: Given a scene description, retrieve document-page images
containing statements that contradict that description.
Scene description: {scene description}
```

Contradiction: Design

```
Task: Given a vehicle design description, retrieve document-page
images stating the design requirements it violates.
Vehicle design: {vehicle design description}
```

Contradiction: Traffic

```
Task: Given a traffic case description, retrieve document-page images
stating the driving rules it violates.
Traffic case: {traffic case description}
```

B.2 Benchmark Examples

Figures 8–32 show the query and document image for each of the ten ViDoRe-v1, four ViDoRe-v2, and eleven MRMR subsets. Each example occupies a full-width panel, preserving legibility across appendix pages.

ArXivQA

Based on the graph, what is the impact of correcting for fspec not equal to 1 on the surface density trend?

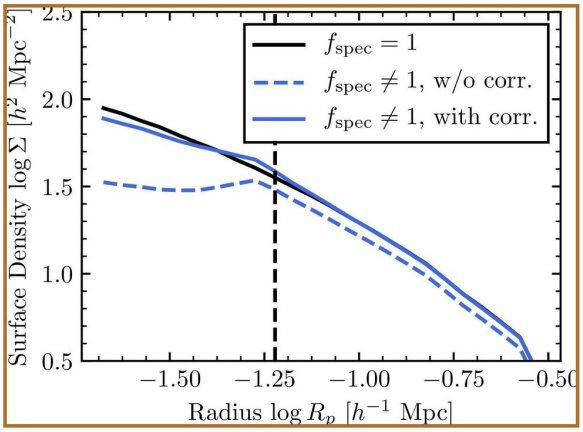


Figure 8: Representative example from the ArXivQA subset.

DocVQA

What is the dividend payout in 2012?



Figure 9: Representative example from the DocVQA subset.

InfoVQA

Which company collects only phone number, email address and location (time zone)?

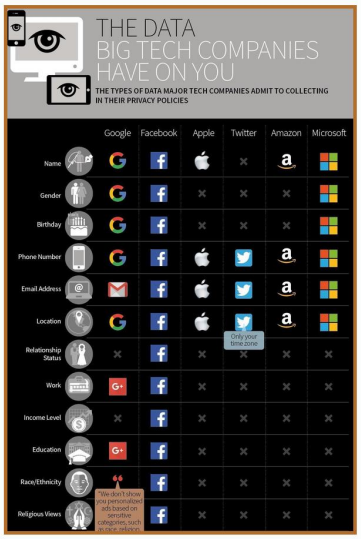


Figure 10: Representative example from the InfoVQA subset.

TabFQuAD

Quelles étaient les principales composantes du bénéfice net par action d'une entreprise pour les années 2016 et 2017?



Bénéfice net retenu pour le calcul du BNPA courant [1]			
Résultat net part du Groupe		862	(527
Éléments exclus :			
• amortissement des incorporels liés aux acquisitions, net d'impôt		55	5
• perte de valeur, nette d'impôt		115	1 383
• effet net de la réforme fiscale aux États-Unis		(61)	
• réévaluation des compléments de prix d'acquisition		66	108
Résultat net courant part du Groupe	E	1 037	1 015
Impact des instruments dilutifs :			
• économies de frais financiers liés à la conversion des instruments de dette, nettes d'impôt		-	
Résultat net courant part du Groupe – dilué	F	1 037	1 015
Nombre d'actions retenu pour le calcul du BNPA			
Nombre d'actions au 1er janvier		225 945 387	222 540 740
Actions créées sur la période		2 529 801	1 771 86
Actions propres à déduire (moyenne sur l'exercice)		(2 090 481)	(813 730
Nombre d'actions moyen retenu pour le calcul	C	226 384 707	223 498 87
Impact des instruments dilutifs :			
• actions gratuites et stock-options dilutifs [2]		3 682 435	3 488 040
• bons de souscription d'actions (BSA) [2]		606 436	718 168
Nombre d'actions – dilué	D	230 673 578	227 705 075
(en euros)			
BÉNÉFICE NET COURANT PAR ACTION [1]	E/C	4,58	4,54
BÉNÉFICE NET COURANT PAR ACTION – DILUÉ [1]	F/D	4,50	4,46

Figure 11: Representative example from the TabFQuAD subset.

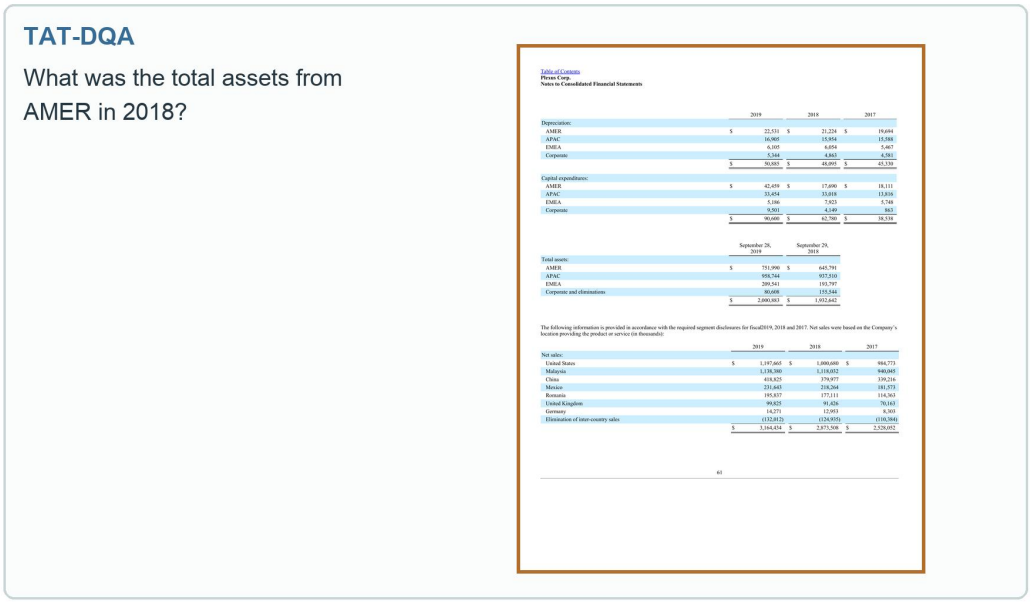


Table of Contents
Pirsas Corp.
Notes to Consolidated Financial Statements

	2019	2018	2017
Depreciation:			
AMER	$ 22,531	$ 21,224	$ 19,694
APAC	16,905	15,954	15,588
EMEA	6,105	6,054	5,467
Corporate	5,344	4,863	4,581
	$ 50,885	$ 48,095	$ 45,330
Capital expenditures:			
AMER	$ 42,459	$ 17,690	$ 18,111
APAC	33,454	33,018	13,816
EMEA	5,186	7,923	5,748
Corporate	9,501	4,149	863
	$ 90,600	$ 62,780	$ 38,538

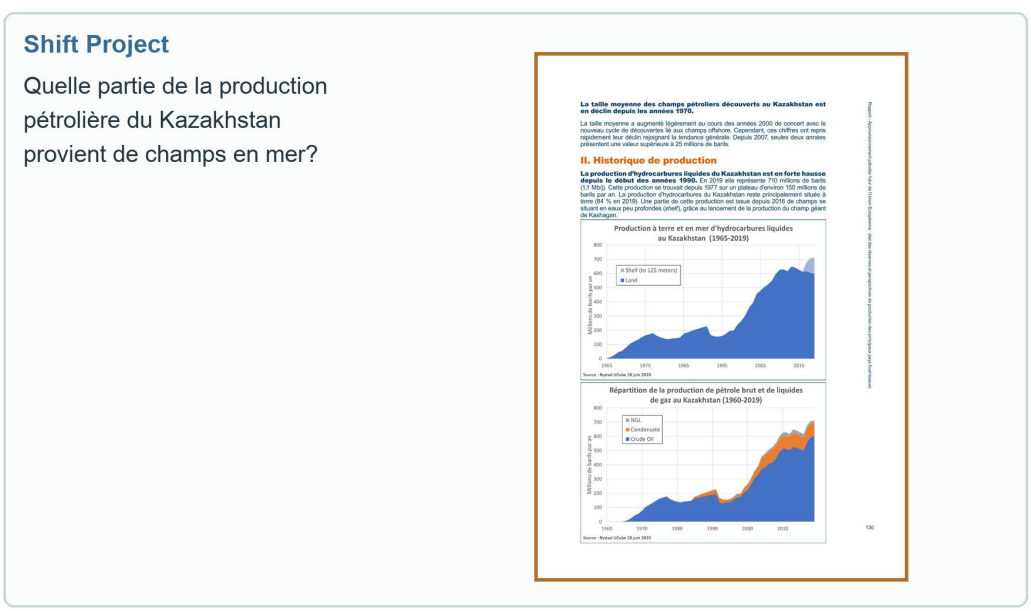
	September 28, 2019	September 29, 2018
Total assets:		
AMER	$ 751,990	$ 645,791
APAC	958,744	937,510
EMEA	209,541	193,797
Corporate and eliminations	80,608	155,544
	$ 2,000,883	$ 1,932,642

The following information is provided in accordance with the required segment disclosures for fiscal2019, 2018 and 2017. Net sales were based on the Company's location providing the product or service (in thousands):

	2019	2018	2017
Net sales:			
United States	$ 1,197,665	$ 1,000,680	$ 984,773
Malaysia	1,138,380	1,118,032	940,045
China	418,825	379,977	339,216
Mexico	231,443	218,264	181,573
Romania	195,837	177,111	114,363
United Kingdom	99,825	91,426	70,163
Germany	14,271	12,953	8,303
Elimination of inter-country sales	(132,812)	(124,935)	(110,384)
	$ 3,164,434	$ 2,873,508	$ 2,528,052

61

Figure 12: Representative example from the TAT-DQA subset.



La taille moyenne des champs pétroliers découverts au Kazakhstan est en déclin depuis les années 1970.

La taille moyenne a augmenté légèrement au cours des années 2000 de concert avec le nouveau cycle de découvertes lié aux champs offshore. Cependant, ces chiffres ont repris rapidement leur déclin rejoignant la tendance générale. Depuis 2007, seules deux années présentent une valeur supérieure à 25 millions de barils.

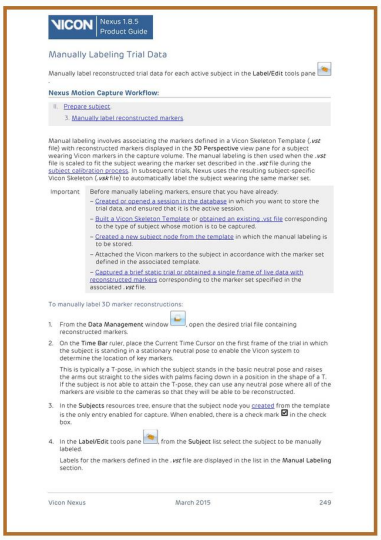
II. Historique de production

La production d'hydrocarbures liquides du Kazakhstan est en forte hausse depuis le début des années 1990. En 2019 elle représente 710 millions de barils (1,1 Mb/j). Cette production se trouvait depuis 1977 sur un plateau d'environ 150 millions de barils par an. La production d'hydrocarbures du Kazakhstan reste principalement située à terre (84 % en 2019). Une partie de cette production est issue depuis 2016 de champs se situant en eaux peu profondes (shelf), grâce au lancement de la production du champ géant de Kashagan.

Figure 13: Representative example from the Shift Project subset.

Synthetic: AI

What is the process for manually labeling trial data in Vicon Nexus?



VICON Nexus 1.8.5 Product Guide

Manually Labeling Trial Data

Manually label reconstructed trial data for each active subject in the **Label/Edit** tools pane .

Nexus Motion Capture Workflow:

4. Prepare subject
5. Manually label reconstructed markers

Manual labeling involves associating the markers defined in a Vicon Skeleton Template (*.vst* file) with reconstructed markers displayed in the **3D Perspective** view pane for a subject wearing Vicon markers in the capture volume. The manual labeling is then used when the *.vst* file is scaled to fit the subject wearing the marker set described in the *.vsk* file during the subject calibration process. In subsequent trials, Nexus uses the resulting subject-specific Vicon Skeleton (*.vsk* file) to automatically label the subject wearing the same marker set.

Important: Before manually labeling markers, ensure that you have already:
- Created or opened a session in the database in which you want to store the trial data, and ensured that it is the active session.
- Built a Vicon Skeleton Template or obtained an existing *.vst* file corresponding to the type of subject whose motion is to be captured.
- Created a new subject node from the template in which the manual labeling is to be stored.
- Attached the Vicon markers to the subject in accordance with the marker set defined in the associated template.
- Captured a brief static trial or obtained a single frame of live data with reconstructed markers corresponding to the marker set specified in the associated *.vst* file.

To manually label 3D marker reconstructions:

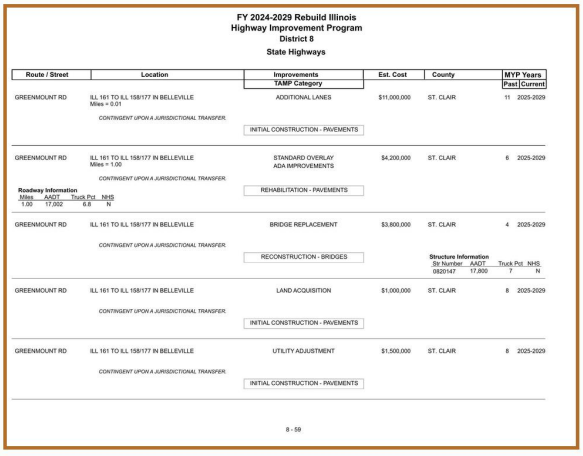
1. From the **Data Management** window , open the desired trial file containing reconstructed markers.
2. On the **Time Bar** ruler, place the Current Time Cursor on the first frame of the trial in which the subject is standing in a stationary neutral pose to enable the Vicon system to determine the location of key markers.
   This is typically a T-pose, in which the subject stands in the basic neutral pose and raises the arms out straight to the sides with palms facing down in a position in the shape of a T. If the subject is not able to attain the T-pose, they can use any neutral pose where all of the markers are visible to the cameras so that they will be able to be reconstructed.
3. In the **Subjects** resources tree, ensure that the subject node you created from the template is the only entry enabled for capture. When enabled, there is a check mark in the check box.
4. In the **Label/Edit** tools pane , from the **Subject** list select the subject to be manually labeled.
   Labels for the markers defined in the *.vst* file are displayed in the list in the **Manual Labeling** section.

Vicon Nexus March 2015 249

Figure 14: Representative example from the Synthetic: AI subset.

Synthetic: Energy

What types of road improvements are planned for Greenmount Road in Belleville, Illinois?



FY 2024-2029 Rebuild Illinois
Highway Improvement Program
District 8
State Highways

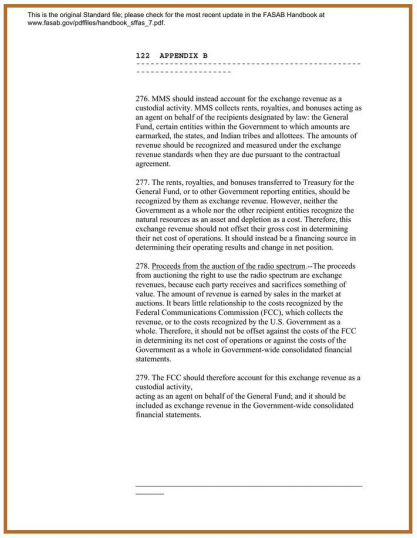
Route / Street	Location	Improvements / TAMP Category	Est. Cost	County	MYP Years Past	MYP Years Current
GREENMOUNT RD	ILL 161 TO ILL 158/177 IN BELLEVILLE Miles = 0.01	ADDITIONAL LANES	$11,000,000	ST. CLAIR	11	2025-2029
	CONTINGENT UPON A JURISDICTIONAL TRANSFER.	INITIAL CONSTRUCTION - PAVEMENTS				
GREENMOUNT RD	ILL 161 TO ILL 158/177 IN BELLEVILLE Miles = 1.00	STANDARD OVERLAY ADA IMPROVEMENTS	$4,200,000	ST. CLAIR	6	2025-2029
	CONTINGENT UPON A JURISDICTIONAL TRANSFER.	REHABILITATION - PAVEMENTS				
	Roadway Information Miles 1.00, AADT 17,002, Truck Pct 6.8, NHS N					
GREENMOUNT RD	ILL 161 TO ILL 158/177 IN BELLEVILLE	BRIDGE REPLACEMENT	$3,600,000	ST. CLAIR	4	2025-2029
	CONTINGENT UPON A JURISDICTIONAL TRANSFER.	RECONSTRUCTION - BRIDGES				
				Structure Information Str Number 082014T, AADT 17,800, Truck Pct 7, NHS N		
GREENMOUNT RD	ILL 161 TO ILL 158/177 IN BELLEVILLE	LAND ACQUISITION	$1,000,000	ST. CLAIR	8	2025-2029
	CONTINGENT UPON A JURISDICTIONAL TRANSFER.	INITIAL CONSTRUCTION - PAVEMENTS				
GREENMOUNT RD	ILL 161 TO ILL 158/177 IN BELLEVILLE	UTILITY ADJUSTMENT	$1,500,000	ST. CLAIR	8	2025-2029
	CONTINGENT UPON A JURISDICTIONAL TRANSFER.	INITIAL CONSTRUCTION - PAVEMENTS				

8 - 59

Figure 15: Representative example from the Synthetic: Energy subset.

Synthetic: Government

What types of revenue should MMS account for as a custodial activity?



This is the original Standard file; please check for the most recent update in the FASAB Handbook at www.fasab.gov/pdffiles/handbook_sffas_7.pdf.

122 APPENDIX B

276. MMS should instead account for the exchange revenue as a custodial activity. MMS collects rents, royalties, and bonuses acting as an agent on behalf of the recipients designated by law: the General Fund, certain entities within the Government to which amounts are earmarked, the states, and Indian tribes and allottees. The amounts of revenue should be recognized and measured under the exchange revenue standards when they are due pursuant to the contractual agreement.

277. The rents, royalties, and bonuses transferred to Treasury for the General Fund, or to other Government reporting entities, should be recognized by them as exchange revenue. However, neither the Government as a whole nor the other recipient entities recognize the natural resources as an asset and depletion as a cost. Therefore, this exchange revenue should not offset their gross cost in determining their net cost of operations. It should instead be a financing source in determining their operating results and change in net position.

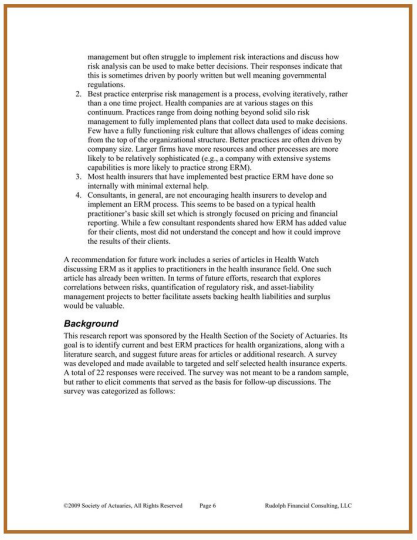
278. Proceeds from the auction of the radio spectrum.--The proceeds from auctioning the right to use the radio spectrum are exchange revenues, because each party receives and sacrifices something of value. The amount of revenue is earned by sales in the market at auctions. It bears little relationship to the costs recognized by the Federal Communications Commission (FCC), which collects the revenue, or to the costs recognized by the U.S. Government as a whole. Therefore, it should not be offset against the costs of the FCC in determining its net cost of operations or against the costs of the Government as a whole in Government-wide consolidated financial statements.

279. The FCC should therefore account for this exchange revenue as a custodial activity,
acting as an agent on behalf of the General Fund; and it should be included as exchange revenue in the Government-wide consolidated financial statements.

Figure 16: Representative example from the Synthetic: Government subset.

Synthetic: Healthcare

What are some challenges faced by health companies in implementing enterprise risk management?



management but often struggle to implement risk interactions and discuss how risk analysis can be used to make better decisions. Their responses indicate that this is sometimes driven by poorly written but well meaning governmental regulations.

2. Best practice enterprise risk management is a process, evolving iteratively, rather than a one time project. Health companies are at various stages on this continuum. Practices range from doing nothing beyond solid silo risk management to fully implemented plans that collect data used to make decisions. Few have a fully functioning risk culture that allows challenges of ideas coming from the top of the organizational structure. Better practices are often driven by company size. Larger firms have more resources and other processes are more likely to be relatively sophisticated (e.g., a company with extensive systems capabilities is more likely to practice strong ERM).
3. Most health insurers that have implemented best practice ERM have done so internally with minimal external help.
4. Consultants, in general, are not encouraging health insurers to develop and implement an ERM process. This seems to be based on a typical health practitioner's basic skill set which is strongly focused on pricing and financial reporting. While a few consultant respondents shared how ERM has added value for their clients, most did not understand the concept and how it could improve the results of their clients.

A recommendation for future work includes a series of articles in Health Watch discussing ERM as it applies to practitioners in the health insurance field. One such article has already been written. In terms of future efforts, research that explores correlations between risks, quantification of regulatory risk, and asset-liability management projects to better facilitate assets backing health liabilities and surplus would be valuable.

Background

This research report was sponsored by the Health Section of the Society of Actuaries. Its goal is to identify current and best ERM practices for health organizations, along with a literature search, and suggest future areas for articles or additional research. A survey was developed and made available to targeted and self selected health insurance experts. A total of 22 responses were received. The survey was not meant to be a random sample, but rather to elicit comments that served as the basis for follow-up discussions. The survey was categorized as follows:

©2009 Society of Actuaries, All Rights Reserved Page 6 Rudolph Financial Consulting, LLC

Figure 17: Representative example from the Synthetic: Healthcare subset.

ESG

Quelles sont les marques de RBI mentionnées dans le rapport?

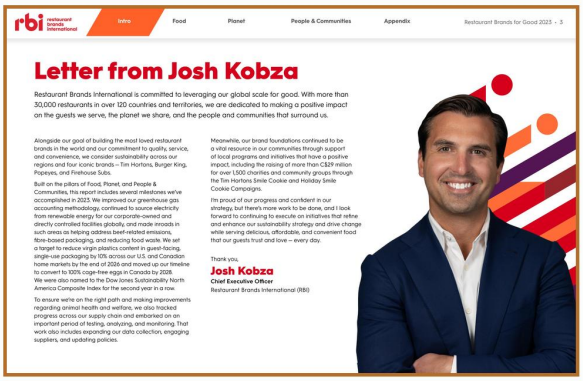


Figure 18: Representative example from the ESG subset.

Biomedical

What are the specific outcomes of using autologous chondrocyte implantation in canine studies?

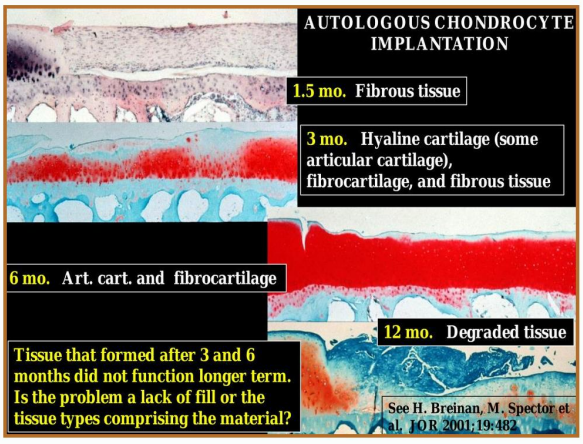
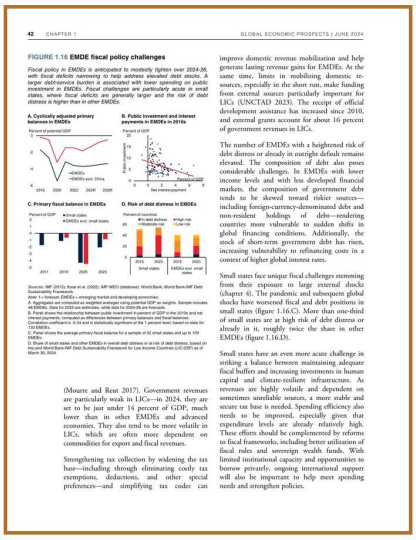


Figure 19: Representative example from the Biomedical subset.

Economics

How do the fiscal challenges in small states compare to those in larger economies?



42 CHAPTER 1 GLOBAL ECONOMIC PROSPECTS | JUNE 2024

FIGURE 1.16 EMDE fiscal policy challenges

Fiscal policy in EMDEs is anticipated to modestly tighten over 2024-26, with fiscal deficits narrowing to help address elevated debt stocks. A larger debt-service burden is associated with lower spending on public investment in EMDEs. Fiscal challenges are particularly acute in small states, where fiscal deficits are generally larger and the risk of debt distress is higher than in other EMDEs.

A. Cyclically adjusted primary balances in EMDEs

B. Public investment and interest payments in EMDEs in 2010s

C. Primary fiscal balance in EMDEs

D. Risk of debt distress in EMDEs

(Mourre and Reut 2017). Government revenues are particularly weak in LICs—in 2024, they are set to be just under 14 percent of GDP, much lower than in other EMDEs and advanced economies. They also tend to be more volatile in LICs, which are often more dependent on commodities for export and fiscal revenues.

Strengthening tax collection by widening the tax base—including through eliminating costly tax exemptions, deductions, and other special preferences—and simplifying tax codes can improve domestic revenue mobilization and help generate lasting revenue gains for EMDEs. At the same time, limits in mobilizing domestic resources, especially in the short run, make funding from external sources particularly important for LICs (UNCTAD 2023). The receipt of official development assistance has increased since 2010, and external grants account for about 16 percent of government revenues in LICs.

The number of EMDEs with a heightened risk of debt distress or already in outright default remains elevated. The composition of debt also poses considerable challenges. In EMDEs with lower income levels and with less developed financial markets, the composition of government debt tends to be skewed toward riskier sources—including foreign-currency-denominated debt and non-resident holdings of debt—rendering countries more vulnerable to sudden shifts in global financing conditions. Additionally, the stock of short-term government debt has risen, increasing vulnerability to refinancing costs in a context of higher global interest rates.

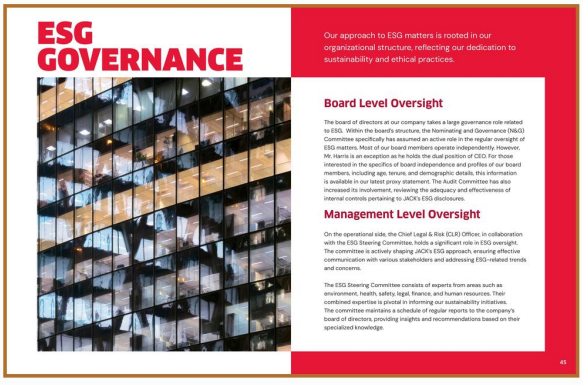
Small states face unique fiscal challenges stemming from their exposure to large external shocks (chapter 4). The pandemic and subsequent global shocks have worsened fiscal and debt positions in small states (figure 1.16.C). More than one-third of small states are at high risk of debt distress or already in it, roughly twice the share in other EMDEs (figure 1.16.D).

Small states have an even more acute challenge in striking a balance between maintaining adequate fiscal buffers and increasing investments in human capital and climate-resilient infrastructure. As revenues are highly volatile and dependent on sometimes unreliable sources, a more stable and secure tax base is needed. Spending efficiency also needs to be improved, especially given that expenditure levels are already relatively high. These efforts should be complemented by reforms to fiscal frameworks, including better utilization of fiscal rules and sovereign wealth funds. With limited institutional capacity and opportunities to borrow privately, ongoing international support will also be important to help meet spending needs and strengthen policies.

Figure 20: Representative example from the Economics subset.

ESG-Human

Who is responsible for integrating climate consideration into Jack in the box governance?



ESG GOVERNANCE

Our approach to ESG matters is rooted in our organizational structure, reflecting our dedication to sustainability and ethical practices.

Board Level Oversight

The board of directors at our company takes a large governance role related to ESG. Within the board's structure, the Nominating and Governance (N&G) Committee specifically has assumed an active role in the regular oversight of ESG matters. Most of our board members operate independently. However, Mr. Harris is an exception as he holds the dual position of CEO. For those interested in the specifics of board independence and profiles of our board members, including age, tenure, and demographic details, this information is available in our latest proxy statement. The Audit Committee has also increased its involvement, reviewing the adequacy and effectiveness of internal controls pertaining to JACK's ESG disclosures.

Management Level Oversight

On the operational side, the Chief Legal & Risk (CLR) Officer, in collaboration with the ESG Steering Committee, holds a significant role in ESG oversight. The committee is actively shaping JACK's ESG approach, ensuring effective communication with various stakeholders and addressing ESG-related trends and concerns.

The ESG Steering Committee consists of experts from areas such as environment, health, safety, legal, finance, and human resources. Their combined expertise is pivotal in informing our sustainability initiatives. The committee maintains a schedule of regular reports to the company's board of directors, providing insights and recommendations based on their specialized knowledge.

45

Figure 21: Representative example from the ESG-Human subset.

Knowledge / Art

Who were the architects of the temple shown in the figure?

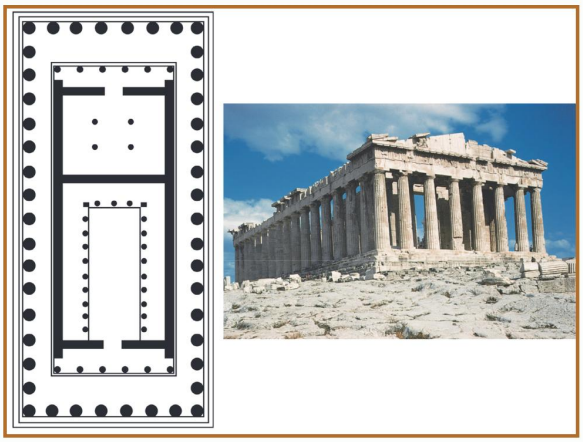


Figure 22: Representative example from the Knowledge / Art subset.

Knowledge / Medicine

Which organ(s) are abnormally large in this abdominal radiograph?

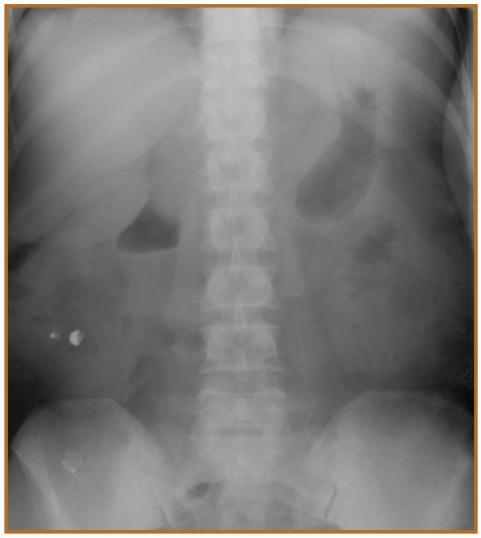


Figure 23: Representative example from the Knowledge / Medicine subset.

Knowledge / Science

The following choices show the atomic orbitals of some neutral atoms. Choose the one that forms an aqueous cation that is colored.

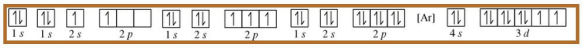


Figure 24: Representative example from the Knowledge / Science subset.

Knowledge / Humanities

A historian would most likely use the figure in Source 1 as support for what assertions about Daoism?



Figure 25: Representative example from the Knowledge / Humanities subset.

Theorem / Mathematics

The radius of the circle above is 4 and $\angle A=45^{\circ}$. What is the area of the shaded section of the circle?

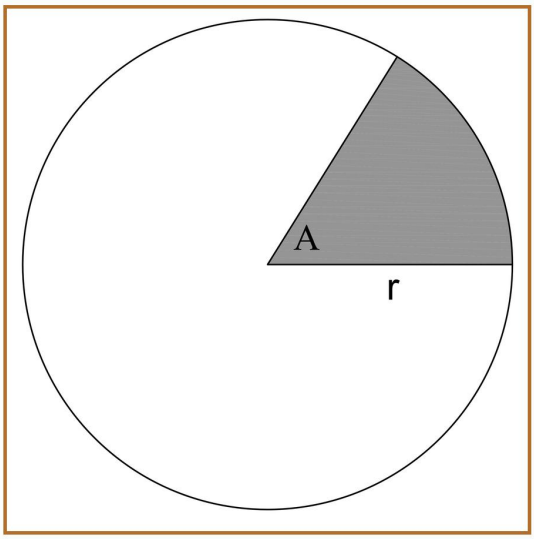


Figure 26: Representative example from the Theorem / Mathematics subset.

Theorem / Physics

Three processes compose a thermodynamic cycle shown in the accompanying pV diagram of an ideal gas. Process 1->2 takes place at constant temperature (300 K). During this process 60 J of heat enters the system. Process 2->3 takes place at constant volume. During this process 40 J of heat leaves the system. Process 3->1 is adiabatic. T_3 is 275 K. What is the change in internal energy of the system during process 3->1?

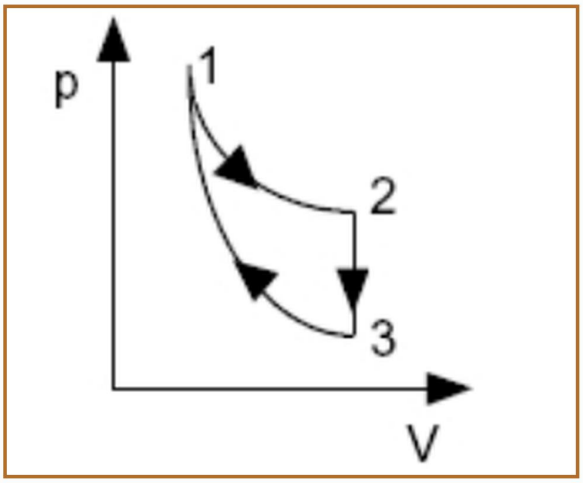


Figure 27: Representative example from the Theorem / Physics subset.

Theorem / Engineering

Derive the equations of motion of the system shown in the figure.

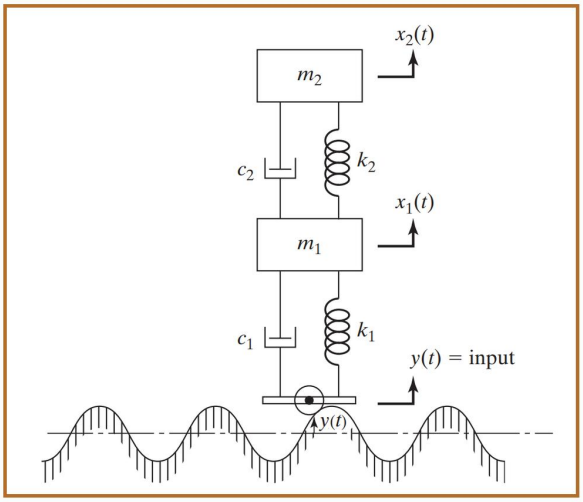


Figure 28: Representative example from the Theorem / Engineering subset.

Theorem / Business

For case B in the following table, find the future value at the end of the deposit period, assuming that interest is compounded continuously at the given nominal annual rate.

Case	Amount of initial deposit	Nominal annual rate, r	Deposit period (years), n
A	$1,000	9%	2
B	600	10	10
C	4,000	8	7
D	2,500	12	4

Figure 29: Representative example from the Theorem / Business subset.

Contradiction / Negation

This figure includes umbrella, backpack, chair but no person.



Figure 30: Representative example from the Contradiction / Negation subset.

Contradiction / Design

We are a student engineering team designing a vehicle for the FSAE competition. Attached is a figure that shows an engineering drawing of the vehicle on the top accompanied by six CAD views of the vehicle on the bottom. The six CAD views each feature a different orientation of our design, so that 3D information about our design can be inferred. In the CAD views, the wheels are highlighted in pink. The CAD views are provided to contextualize the engineering drawing, which has the same orientation as one of the six CAD views. All units displayed in the engineering drawing have units of mm.

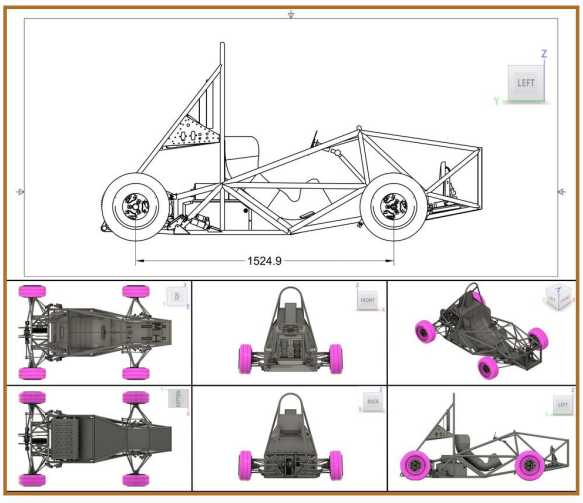


Figure 31: Representative example from the Contradiction / Design subset.

Contradiction / Traffic

Jane was driving the truck to her company in Singapore.



Figure 32: Representative example from the Contradiction / Traffic subset.

C Implementation and Reproducibility Details

This section provides the complete training and evaluation settings summarized in Section 4.1.

C.1 Training Data, Models, and Optimization

Training data and model selection. The two diffusion retrievers use the ColPali training collection[1] (Faysse et al., 2025), comprising 118,195 training pairs and a separate 500-pair evaluation split. Training uses in-batch negatives rather than explicit hard negatives. We use the final checkpoints after five epochs. The AR comparisons are ColLLaVA-1.6–Mistral-7B, ColLLaVA-Llama3, ColQwen2-7B, and ColQwen2.5-7B. All use multi-vector representations, and their archived run settings specify five epochs, a learning rate of 2×10^{-4}, LoRA, and a per-device batch size of eight. Table 2 summarizes the two main diffusion configurations.

Table 2: Verified adaptation settings for the two designated diffusion runs. Base weights remain frozen under LoRA; visual LoRA differs between the models. The cached LaViDa path and standard SDAR path are distinct optimization recipes.

Setting	CoLLaViDa	ColSDAR-VL
Training pairs / evaluation pairs	118,195 / 500	118,195 / 500
Epochs / final updates	5 / 2,310	5 / 9,235
Learning rate / temperature	2×10^{-4} / 0.02	2×10^{-4} / 0.02
Optimizer / schedule	AdamW / linear	AdamW / linear
Warmup updates / seed	100 / 42	100 / 42
LoRA rank / alpha / dropout	64 / 64 / 0.1	64 / 64 / 0.1
Devices / microbatch / accumulation	16 / 4 / 4	16 / 4 / 1
Global update batch	256	64
Contrastive gradient path	Query-side cached	Both sides
Visual feature layer	Penultimate	Final (native)
Visual LoRA	No	Yes
Retrieval dimension	128	128
Instantiation-specific regularizer	Prompt denoising	Query decorrelation
Regularizer weight	0.5	0.1

Optimizer and numerical settings. Both diffusion runs use AdamW with a learning rate of 2×10^{-4}, a linear schedule with 100 warmup updates, temperature $\tau = 0.02$, and seed 42. LoRA rank and scaling parameter are both 64, with dropout 0.1. Backbone weights are loaded in bfloat16, and gradient checkpointing is enabled. These settings apply to the designated main checkpoints; the readout and auxiliary-coefficient configurations are reported separately in Appendix F.

Batching and gradient paths. CoLLaViDa uses 16 devices, microbatch size four, and four accumulation steps, for a global update batch of 256. Its cached contrastive path stops direct document-side gradients. ColSDAR-VL uses 16 devices, microbatch size four, and one accumulation step, for a global batch of 64, with gradients through both encoders. The respective runs finish at 2,310 and 9,235 optimizer updates. The two runs therefore use different optimization recipes.

Auxiliary coefficients. We use $\lambda_{\mathrm{pd}} = 0.5$, $\rho = 0.15$, and $\sigma = 0.1$ for CoLLaViDa's prompt-denoising branch, and $\lambda_{\mathrm{qd}} = 0.1$, $c = 0$ for ColSDAR-VL's query regularizer. The checkpoint identifiers record the requested coefficient on the auxiliary denoising loss: 0.5 for CoLLaViDa and 0.3 for ColSDAR-VL; they also encode the run-specific visual-sequence and query-decorrelation settings. The corruption and padding conventions underlying these objectives are specified below.

[1] https://huggingface.co/datasets/vidore/colpali_train_set

C.2 Visual Processing and Implementation

Visual processing. ColLaViDa uses LaViDa's native any-resolution tiling and the penultimate SigLIP feature layer. Its visual tower is frozen. ColSDAR-VL uses SDAR-VL's native any-resolution configuration and final SigLIP-2 feature layer; its base weights are frozen, but LoRA adapters also cover visual attention and MLP layers. Both produce 128-dimensional retrieval vectors. Native processing does not enforce matched image-token budgets or encoding costs.

Tokenization and normalization. Text queries are augmented with ten processor-defined augmentation tokens. In full mode, we retain non-padding output positions rather than extracting only generated or mask-position states. Retrieval vectors are row-normalized after the affine projection, and padded output rows are zero. At evaluation, document padding is excluded from `MaxSim`. The batched training loss instead operates on zero-padded document matrices: when a candidate has padding, a zero similarity can participate in the maximum. Equation 4 uses that padded implementation of the score during training. Query-length normalization counts nonzero positions, whereas query decorrelation retains padding in its batch-width denominator (Eq. 6).

Auxiliary denoising details. The document prompt consists of an image placeholder followed by the fixed text *Describe the image.* Text corruption excludes the image placeholder and padding. Visual features are computed without an auxiliary-branch gradient. After image expansion, the implementation perturbs the leading prefix of length $\ell_{\text{expanded}} - \ell_{\text{ids}}$ and aligns text logits to the trailing ℓ_{ids} positions; these two lengths refer to the expanded embedding sequence and pre-expansion token-ID sequence, respectively. Consequently, the corruption is defined by this input layout, not by OCR-derived page spans. A layout-preserving implementation and fixed input checks are necessary when porting this objective to a different multimodal processor.

C.3 Evaluation Protocol and Reporting

ViDoRe subsets and metrics. ViDoRe-v1 comprises ten evaluation subsets, including document, infographic, table, scientific, and synthetic document-query collections (Faysse et al., 2025). ViDoRe-v2 covers more challenging document collections (Macé et al., 2025); we evaluate four subsets: ESG reports, biomedical lectures, economics reports, and ESG-Human (the human-labeled ESG subset). We report unweighted subset means of nDCG@5 for each version. This four-subset v2 average excludes the three English variants used in the broader release. Table 3 lists every included subset, and Appendix E reports additional nDCG, Recall, and Precision cutoffs for all six main retrievers.

MRMR query and candidate protocol. MRMR tests Knowledge, Theorem, and Contradiction retrieval (Zhang et al., 2026); Negation, Design, and Traffic belong to the Contradiction category. Our runs use the document screenshots but a modified query and candidate protocol. For mixed-modal queries, the evaluator uses the provided `text` field (falling back to `question` if empty) and omits query images, without caption substitution. For image-only Negation queries, it uses the image through the page encoder. All queries are ranked against their full task corpus, including all 800 documents for Negation instead of its four query-specific candidates. We follow the official metric definitions, using nDCG@10 for ten subtasks and Hit@1 for Negation, and average the eleven subtask scores equally. MRMR is therefore an explicitly modified evaluation, not an official MRMR leaderboard score. Every model in our tables uses this same protocol; comparison with published MRMR numbers would additionally require matched query information and candidate pools.

Aggregation and reproducibility. All scores are multiplied by 100, and differences are computed before rounding. Each table entry is recomputed from per-subset evaluation files, counting duplicate exports only once. In Table 3, *Overall (Benchmark)* averages the three benchmark-level scores equally, while *Overall (Dataset)* averages all 25 subsets directly. These descriptive summaries mix different metrics and do not replace the individual benchmark results. Scores correspond to the designated model configurations. Section 4.5 separately characterizes the measured native implementations, without claiming an architecture-level speedup. Appendix F records readout and auxiliary-coefficient conventions. Comparisons with multiple changing settings are not treated as single-factor ablations.

Figure conventions. In Figures 3 and 33, the left-panel Δ annotations subtract the lower-scoring AR member's benchmark mean from the diffusion model's mean. The right-panel differences instead use the higher-scoring AR member for each MRMR subtask. The fixed-baseline comparisons in Section 4.2 instead select each family's AR model with the highest MRMR mean.

Scope of the comparisons. The two diffusion models use different pretrained visual encoders, different visual adaptation targets, and different gradient and batch configurations. The AR runs are also not matched in language checkpoints, pretraining data, or visual token count. These are capacity-comparable retrieval systems, not matched-pretraining controls: their results compare the evaluated systems without isolating attention direction or pretraining paradigm. Appendix G reports native runtime, peak memory, and raw vector-storage accounting, with the corresponding measurement definitions.

C.4 Source-Code Map

The following paths are relative to the implementation repository. They connect the mathematical definitions and pseudocode to the components intended for the anonymized code release.

Backbones, processors, and readouts. Under `colpali_engine/models/dllm/`, `collavida/modeling_collavida.py` and `colsdarvl/modeling_colsdarvl.py` define the two retrievers. Their `forward` methods implement independent encoding, the 128-dimensional projection, row normalization, and output masking in Eq. 2. The paired `processing_collavida.py` and `processing_colsdarvl.py` modules construct page prompts and query augmentation. The shared `backbone_adapter.py` and `vision_encoder.py` provide hidden-state access and the native visual-processing paths. In `modeling_collavida.py`, `_mean_pool_forward`, `_extract_mask_tokens`, and `_refine_forward`, together with the Full branch of `forward`, correspond to Algorithm 2.

Objectives and training procedure. The `ColbertLoss` class in `colpali_engine/loss/late_interaction_losses.py` implements the `MaxSim`-based contrastive objective, including query-length normalization and temperature scaling (Eq. 4). The LaViDa model's `forward_denoise` method implements prompt corruption and reconstruction, while `IntraDocDecorrelationLoss` in `colpali_engine/loss/regularization_losses.py` implements the mean-similarity hinge in Eq. 6. In `colpali_engine/trainer/contrastive_trainer.py`, `training_step`, `compute_loss`, and `_compute_loss_from_outputs` assemble the cached or symmetric contrastive paths and auxiliary terms described in Algorithm 1. The shared entry point `scripts/configs/dllm/train_collavida_model.py` configures both diffusion families, including data loading, LoRA, optimization, and losses; `scripts/run_train.sh` provides the launch interface.

Evaluation and result accounting. `scripts/run_eval.sh` dispatches benchmark evaluation, and `scripts/eval_mrmr.py` implements MRMR query loading, full-corpus ranking, and the subtask metrics. Its `load_mrmr_data` function defines the query-modality handling in Appendix C.3. The inputs to the paper's numerical summaries are the saved per-subset records under `data/eval/results/` and the timing records under `tests/efficiency_test/`. The manuscript's companion scripts `build_vidore_metric_tables.py`, `build_readout_comparison.py`, and `build_efficiency_analysis.py` check and aggregate these records for the reported tables and figures.

C.5 Training and Retrieval Algorithm

Algorithm 1 details training and retrieval. ENCODE maps a query or page x to $\mathbf{Z}_x \in \mathbb{R}^{\ell_x \times r}$ with ℓ_x retained rows. The recipe $\mathcal{R}$ specifies caching, accumulation, and auxiliary losses; $\mathcal{G}$ groups microbatches into one optimizer update.

Algorithm 1: MM-DIFFEMBED training and retrieval

Input: pairs $\mathcal{S}$; corpus $\mathcal{C}$; backbone F_Θ; projection $(\mathbf{W}, \mathbf{B})$; recipe $\mathcal{R}$; $\tau, \rho, \sigma, \lambda_{\mathrm{pd}}, \lambda_{\mathrm{qd}}, c$
Output: adapted encoder, page index, and ranking π_q for a new query q

Function `Encode(`x`)`**:**
  Build expanded input $\mathbf{E}_x \in \mathbb{R}^{\ell_x \times h}$ with the native processor
  $\mathbf{H}_x \leftarrow F_\Theta(\mathbf{E}_x; \mathcal{A}_{\mathrm{bi}})$ `// Eq. 1; no generation`
  Project and normalize valid rows as $\mathbf{Z}_x$ using Eq. 2
  return $\mathbf{Z}_x$
foreach *update group* $\mathcal{G}$ *sampled from* $\mathcal{S}$ **do**
  Clear accumulated gradients
  if $\mathcal{R}$ *uses cached documents* **then**
    Encode pages across $\mathcal{G}$ without gradients; cache $\mathrm{sg}(\mathbf{D}_i)$
    Gather cached pages across workers `// fixed candidate pool for this update`
  foreach *microbatch* $\{(q_i, d_i^+)\}_{i=1}^{B_{\mathrm{tr}}}$ *in* $\mathcal{G}$ **do**
    $\mathbf{Q}_i \leftarrow \mathtt{Encode}(q_i) \in \mathbb{R}^{m_i \times r}$ with gradients
    if $\mathcal{R}$ *does not use cached documents* **then**
      $\mathbf{D}_i \leftarrow \mathtt{Encode}(d_i^+) \in \mathbb{R}^{n_i \times r}$ with gradients
      Gather page candidates when distributed synchronization is enabled
    Compute s_{ij} with `MaxSim` (Eq. 3) and valid query lengths m_i
    Form $\mathcal{L}_{\mathrm{ret}} = (\mathcal{L}_{q\to d} + \mathcal{L}_{d\to q})/2$ as in Eq. 4
    `// The reverse loss exchanges the token matrices, not only the score indices.`
    $\mathcal{L}_{\mathrm{pd}} \leftarrow 0$; $\mathcal{L}_{\mathrm{qd}} \leftarrow 0$
    if $\mathcal{R}$ *includes prompt denoising* **then**
      Select prompt positions $\mathcal{M}_{d_i}$ independently with probability ρ
      Detach visual features; construct $\widetilde{\mathbf{E}}_{d_i}$ with $C_{\rho,\sigma}$
      Predict masked prompt tokens and compute $\mathcal{L}_{\mathrm{pd}}$ by Eq. 5
    if $\mathcal{R}$ *includes query decorrelation* **then**
      Zero-pad $\mathbf{Q}_i$ to $\overline{\mathbf{Q}}_i \in \mathbb{R}^{\bar{m} \times r}$
      Average off-diagonal similarities, subtract c, and apply ReLU (Eq. 6)
    Form $\mathcal{L}$ by Eq. 7; backpropagate with accumulation scaling
  Update trainable parameters $(\Theta, \mathbf{W}, \mathbf{B})$ with AdamW
`// Evaluation: auxiliary branches are disabled; pages are encoded offline.`
Cache $\mathbf{D}_j \leftarrow \mathtt{Encode}(d_j)$ for each $d_j \in \mathcal{C}$
$\mathbf{Q} \leftarrow \mathtt{Encode}(q)$; compute $s(q, d_j)$ for all pages using Eq. 3
return π_q, *the pages ordered by decreasing* $s(q, d_j)$

C.6 Alternative Readout Procedure

Algorithm 2 spells out the four branches illustrated in Figure 5, following the CoLLaViDa wrapper and processor. We write one example at a time; batching additionally pads input and output rows. Let $\mathbf{X} \in (\mathcal{V} \cup \{\text{IMG}\})^{\ell_{\text{id}}}$ be the processed IDs, where IMG is the image placeholder, and $\mathbf{A} \in \{0,1\}^{\ell_{\text{id}}}$ their validity mask. After visual expansion, $\mathbf{H} \in \mathbb{R}^{\ell_{\text{exp}} \times h}$ and $\mathbf{A}_{\text{exp}} \in \{0,1\}^{\ell_{\text{exp}}}$ denote hidden states and validity. The mode is ω, k is the mask-slot budget, and $T \geq 1$ is the configured refinement-loop length.

Algorithm 2: Four CoLLaViDa readouts and their shared output processing

Input: query or page x; $\omega \in \{\text{Full}, \text{Mean-pool}, \text{Mask-}k, \text{Refine}\}$; k, T; selection rule; F_Θ and optional projection $(\mathbf{W}, \mathbf{B})$
Output: $\mathbf{Z} \in \mathbb{R}^{\ell_{\text{out}} \times r}$, with padded rows zeroed

Function `Finish` $(\mathbf{Y} \in \mathbb{R}^{\ell_{\text{out}} \times h}$, *optional* $\mathbf{O} \in \{0,1\}^{\ell_{\text{out}}})$**:**
  $\mathbf{P} \leftarrow \mathbf{YW} + \mathbf{1B}^\top$ if projected, else $\mathbf{P} \leftarrow \mathbf{Y}$ and $r \leftarrow h$
  Replace nonfinite entries of $\mathbf{P}$ by zero
  $\mathbf{Z}_{b:} \leftarrow \mathbf{P}_{b:} / \max(\|\mathbf{P}_{b:}\|_2, 10^{-12})$ for each row b
  If $\mathbf{O}$ is supplied, multiply rows by its float32 values; **return** $\mathbf{Z}$
Function `Slots` $(\mathbf{H}, \mathbf{X})$**:**
  $\mathcal{J} \leftarrow$ increasing indices where the original $\mathbf{X}$ equals MASK
  if $\mathcal{J} = \varnothing$ **then** $\mathcal{J} \leftarrow \{\max(1, \ell_{\text{exp}} - k + 1), \ldots, \ell_{\text{exp}}\}$
  Keep the first $\min(k, |\mathcal{J}|)$ indices; gather $\mathbf{H}_{\mathcal{J},:}$
  Pad gathered rows to $\mathbf{Y} \in \mathbb{R}^{k \times h}$; set $\mathbf{O} \in \{0,1\}^k$ to one only at copied rows
  return $(\mathbf{Y}, \mathbf{O})$ `// raw ID indices; no expansion-offset correction`
$(\mathbf{X}, \mathbf{A}) \leftarrow$ native preprocessing before mode-specific appending
`// Queries receive 10 augmentation tokens; pages use the fixed retrieval prompt.`
if $\omega \in \{\text{Mask-}k, \text{Refine}\}$ **then**
  Append k MASK IDs after the padded input width; extend $\mathbf{A}$ with ones
$(\mathbf{H}, \mathbf{A}_{\text{exp}}) \leftarrow$ ordinary backbone pass with native visual injection, if applicable
`// This initial pass also runs for Refine; that branch discards its states.`
if $\omega = \text{Full}$ **then return** `Finish` $(\mathbf{H}, \mathbf{A}_{\text{exp}})$
if $\omega = \text{Mean-pool}$ **then**
  $\mathbf{Y} \leftarrow \sum_b A_{\text{exp},b} \mathbf{H}_{b:} / \max(1, \sum_b A_{\text{exp},b})$ in float32; restore hidden-state dtype and shape $1 \times h$
  return `Finish` $(\mathbf{Y})$ `// pool before projection`
if $\omega = \text{Mask-}k$ **then**
  $(\mathbf{Y}, \mathbf{O}) \leftarrow$ `Slots` $(\mathbf{H}, \mathbf{X})$; **return** `Finish` $(\mathbf{Y}, \mathbf{O})$
$\mathcal{J}_{\text{upd}} \leftarrow \{a : X_a = \text{MASK}\}$; $\mathbf{X}^{(0)} \leftarrow \mathbf{X}$
if x *contains a page* **then**
  Recompute and detach visual features $\mathbf{V}_d \in \mathbb{R}^{p_d \times h}$; cache them within refinement
  Build the initial expanded embeddings and mask `// the wrapper also performs this preparation`
for $t = 0, \ldots, T-1$ **do**
  if $t < T - 1$ **then**
    Under no gradient, rebuild inputs from $\mathbf{X}^{(t)}$ and cached visual features; obtain logits $\mathbf{L}^{(t)} \in \mathbb{R}^{\ell_{\text{exp}} \times |\mathcal{V}|}$
    $\widehat{\mathbf{L}}^{(t)} \leftarrow \mathbf{L}^{(t)}_{\ell_{\text{exp}} - \ell_{\text{id}} + 1 : \ell_{\text{exp}}, :}$ `// right-align logits to ID width`
    Draw $\mathbf{G} \in \mathbb{R}^{\ell_{\text{id}} \times |\mathcal{V}|}$ with i.i.d. standard Gumbel entries for Gumbel selection; otherwise $\mathbf{G} \leftarrow \mathbf{0}$
    Copy $\mathbf{X}^{(t+1)} \leftarrow \mathbf{X}^{(t)}$; for $a \in \mathcal{J}_{\text{upd}}$, set $X_a^{(t+1)} \leftarrow \arg\max_{v \in \mathcal{V}} (\widehat{L}_{av}^{(t)} + G_{av})$
  else
    Rebuild inputs from $\mathbf{X}^{(t)}$; obtain final $\mathbf{H}$ with the caller's gradient setting
    $(\mathbf{Y}, \mathbf{O}) \leftarrow$ `Slots` $(\mathbf{H}, \mathbf{X})$ `// select using original, not updated, IDs`
return `Finish` $(\mathbf{Y}, \mathbf{O})$

Input and position conventions. In Finish, $\mathbf{1} \in \mathbb{R}^{\ell_{\text{out}}}$ is the all-ones vector, $\mathbf{P} \in \mathbb{R}^{\ell_{\text{out}} \times r}$ contains projected rows, and $\mathbf{O}$ marks copied rather than padded output rows. Mean-pool omits this final mask multiplication, as in the implementation. In Slots, ℓ_{exp} is the supplied hidden-state matrix's row count. Its no-MASK fallback uses the final k available rows, not a search over only valid positions.

For page inputs, the newly appended masks are the intended readout slots. Text queries already contain mask-valued augmentation tokens, so the implementation selects the first k mask occurrences across the original IDs, not necessarily just the newly appended block. Refinement updates all original mask occurrences, even when only k are retained for readout.

Refinement and alignment. The Gumbel branch is equivalent to the code's hard Gumbel-softmax selection at unit temperature followed by argmax; this temperature is separate from the contrastive temperature τ. Intermediate predictions are discrete and have no gradient. The final pass uses detached visual features but retains the caller's gradient setting for the language and retrieval layers. The code right-aligns intermediate logits but uses unshifted original-ID indices for final mask readout. Algorithm 2 preserves these operations, and Appendix F records the corresponding evaluation configurations. The initial wrapper pass and separate refinement preparation are also retained, so T is not an end-to-end forward-call or latency measurement.

D More Theoretical Analysis

This section analyzes the encoding and scoring interface, not the relative quality of two pretrained checkpoints. All statements concern deterministic evaluation with fixed weights. They supply conditions under which information can be represented or a ranking is stable; none assumes that a diffusion checkpoint has learned a better representation than an AR checkpoint.

D.1 Attention Support and Available Evidence

Let $\mathbf{E} \in \mathbb{R}^{\ell \times h}$ be the expanded input embedding matrix, and let $\mathbf{H}^{(\nu)} \in \mathbb{R}^{\ell \times h}$ denote the hidden states after language layer $\nu \in \{0, \dots, g\}$, with $\mathbf{H}^{(0)} = \mathbf{E}$. Assume that attention is the only operation mixing positions; normalization, feed-forward blocks, and residual updates are position-wise.

Theorem 1 (Prefix invariance under causal attention)**.** *Consider two input matrices $\mathbf{E}, \mathbf{E}' \in \mathbb{R}^{\ell \times h}$ with $\mathbf{E}_{b:} = \mathbf{E}'_{b:}$ for every $b \le a$. A transformer using $\mathcal{A}_{\mathrm{AR}}(b) = \{1, \dots, b\}$ satisfies $\mathbf{H}^{(\nu)}_{b:} = \mathbf{H}'^{(\nu)}_{b:}$ for all $b \le a$ and every layer ν. This invariance is not imposed by full bidirectional attention.*

Proof sketch. The result holds at layer zero by assumption. At the next layer, a position $b \le a$ attends only to states at positions at most b, which are identical by the induction hypothesis; all remaining operations are position-wise. Induction proves the claim. In a bidirectional layer, a suffix state can receive nonzero attention weight at b, so changing that state can change the output. This establishes a permitted dependency, not a guaranteed one. □

The premise concerns *expanded transformer inputs*, not raw page pixels: a vision encoder can already mix information across a page before the language layers. Moreover, a causal model's later tokens can encode suffix evidence. The theorem therefore motivates contextualizing *each retained token*; it does not show that an AR multi-vector representation is information-free or that changing an attention mask preserves pretrained capabilities. The masked-token objectives of LaViDa and SDAR-VL motivate using their pretrained contextualization, while their generation algorithms remain separate from this retrieval analysis (Li et al., 2025; Cheng et al., 2025b).

D.2 Score Stability, Ranking Margins, and Compression

Write $s(\mathbf{Q}, \mathbf{D})$ for the score in Eq. 3. The following bound applies to vectors of norm at most one, including zero padding when padding is treated consistently in both arguments.

Theorem 2 (Lipschitz stability of multi-vector scoring)**.** *Let $\mathbf{Q}, \mathbf{Q}' \in \mathbb{R}^{m \times r}$ and $\mathbf{D}, \mathbf{D}' \in \mathbb{R}^{n \times r}$ have row norms at most one, and let $\epsilon_q = \max_a \|\mathbf{Q}_{a:} - \mathbf{Q}'_{a:}\|_2$ and $\epsilon_d = \max_b \|\mathbf{D}_{b:} - \mathbf{D}'_{b:}\|_2$. Then*

$$|s(\mathbf{Q}, \mathbf{D}) - s(\mathbf{Q}', \mathbf{D}')| \le m(\epsilon_q + \epsilon_d). \tag{10}$$

Proof. For each token pair, add and subtract $\langle \mathbf{Q}'_{a:}, \mathbf{D}_{b:} \rangle$ and apply Cauchy–Schwarz:

$$|\langle \mathbf{Q}_{a:}, \mathbf{D}_{b:} \rangle - \langle \mathbf{Q}'_{a:}, \mathbf{D}'_{b:} \rangle| \le \epsilon_q + \epsilon_d.$$

The inequality $|\max_b u_b - \max_b v_b| \le \max_b |u_b - v_b|$ then bounds each query row's maximum. Summing over the m query rows proves the result. □

Corollary 1 (Pairwise ranking preservation)**.** *Suppose a positive page and a negative page have score gap $\gamma = s(\mathbf{Q}, \mathbf{D}^+) - s(\mathbf{Q}, \mathbf{D}^-) > 0$. If both page perturbations satisfy the same ϵ_d bound and $\gamma > 2m(\epsilon_q + \epsilon_d)$, their order is unchanged after perturbation. For query-length-normalized scores s/m, the corresponding threshold is $2(\epsilon_q + \epsilon_d)$.*

Proof. Apply Theorem 2 to each page; one score can decrease by at most the bound and the other can increase by at most the bound. □

Corollary 2 (Coverage-preserving document compression)**.** *Let $\mathbf{C}_d \in \mathbb{R}^{k \times r}$ consist of k selected rows of $\mathbf{D} \in \mathbb{R}^{n \times r}$. If every original row has a selected row within Euclidean distance δ, then for query rows of norm at most one,*

$$0 \le s(\mathbf{Q}, \mathbf{D}) - s(\mathbf{Q}, \mathbf{C}_d) \le m\delta. \tag{11}$$

Proof sketch. Selection cannot increase any maximum. For the original maximizing row of each query token, the coverage assumption supplies a selected row whose inner product decreases by at most δ. Sum the resulting bounds. □

This result offers a testable quality–storage criterion: a small vector budget is useful when it preserves evidence coverage. Mean pooling and mask-position readout are not row selection and need not satisfy this condition. Their performance therefore cannot be inferred from the corollary; the readout experiments measure that trade-off directly.

D.3 What the Auxiliary Objectives Do and Do Not Imply

Query decorrelation. For the m valid unit rows of a query, expand the squared norm of their sum:

$$\left\| \sum_{a=1}^{m} \mathbf{Q}_{a:} \right\|_2^2 = m + 2 \sum_{a<a'} \langle \mathbf{Q}_{a:}, \mathbf{Q}_{a':} \rangle. \tag{12}$$

This proves the identity in Section 3.6. With zero padding to $\bar{m}$, the numerator remains $\| \sum_a \mathbf{Q}_{a:} \|_2^2 - m$, while the implemented denominator is $\bar{m}(\bar{m}-1)$. Consequently, padding dilutes the penalty. Nor does zero penalty imply a high-rank representation: for even m, equally many copies of a unit vector $\mathbf{U} \in \mathbb{R}^r$ and $-\mathbf{U}$ have zero centroid and rank one, yet incur zero penalty when $c = 0$. The objective discourages a positive average alignment, not all forms of redundancy.

Prompt denoising. For a randomly sampled masked prompt position a, let $p^*(v \mid \widetilde{\mathbf{E}}, a)$ be the data-induced conditional probability of vocabulary token $v \in \mathcal{V}$. Its expected cross-entropy with the model distribution decomposes as

$$\mathbb{E}_{v \sim p^*}[-\log p_\Theta(v \mid \widetilde{\mathbf{E}}, a)] = \mathcal{H}(p^*) + \mathrm{KL}(p^* \| p_\Theta), \tag{13}$$

where $\mathcal{H}$ is Shannon entropy and KL is Kullback–Leibler divergence, both conditional on the displayed input and position. This follows by adding and subtracting $-\log p^*(v \mid \widetilde{\mathbf{E}}, a)$. Minimizing the loss encourages the correct masked-prompt distribution under corruption, but it supplies no direct bound on relevance or ranking error. In particular, reconstructing a fixed retrieval prompt is not reconstructing the page's semantic content. Its contribution must therefore be established empirically, rather than justified as guaranteed reasoning preservation.

D.4 Computational Implications

For g language layers and expanded length ℓ, dense transformer encoding has order $O(g(\ell h^2 + \ell^2 h))$. Full bidirectional attention exposes ℓ^2 position pairs, while causal attention exposes $\ell(\ell+1)/2$; implementations can exploit this difference, so full attention is not automatically cheaper. Projection adds $O(\ell h r)$, and an uncompressed corpus index stores $r \sum_{j=1}^{N} n_j$ scalar values, where n_j is page j's retained vector count. Computing `MaxSim` for m query vectors and n page vectors costs $O(mnr)$ per candidate pair; exact scoring against the full corpus therefore costs $O(mr \sum_j n_j)$ before top-k selection. These expressions exclude the backbone-specific vision encoder and preprocessing costs.

Both our main encoders and the AR multi-vector baselines process observed inputs in one pass and have the same asymptotic encoding order at equal sequence lengths, hidden widths, and layer counts. Any practical advantage must therefore be established through measurements of parallelism, token counts, memory, or indexing, rather than inferred from the absence of autoregressive decoding. Generation-time prefix caching or block-wise decoding speedups (Li et al., 2025; Cheng et al., 2025b) therefore do not transfer automatically. A parallel mask-position retriever can avoid the sequential output-token generation used by a generated-representation baseline (Wang et al., 2026a), but that is a different comparison from our full-token AR baselines. The relevant empirical quantities are expanded token counts, actual forward-call counts, encoding latency, memory, vector storage, and search latency at a specified retrieval-quality target. Appendix G reports native implementation measurements; its output-normalized timings are descriptive ratios, not measurements under equal sequence lengths or kernels.

E Additional Retrieval Results

This section collects the full benchmark results and complementary ViDoRe retrieval-depth metrics for the six main models.

E.1 Benchmark-Level and Subtask Results

Figure 33 shows the Qwen-family benchmark means and MRMR subtask differences using the main-text LLaVA-family convention. Table 3 provides the complete subset-level comparison.

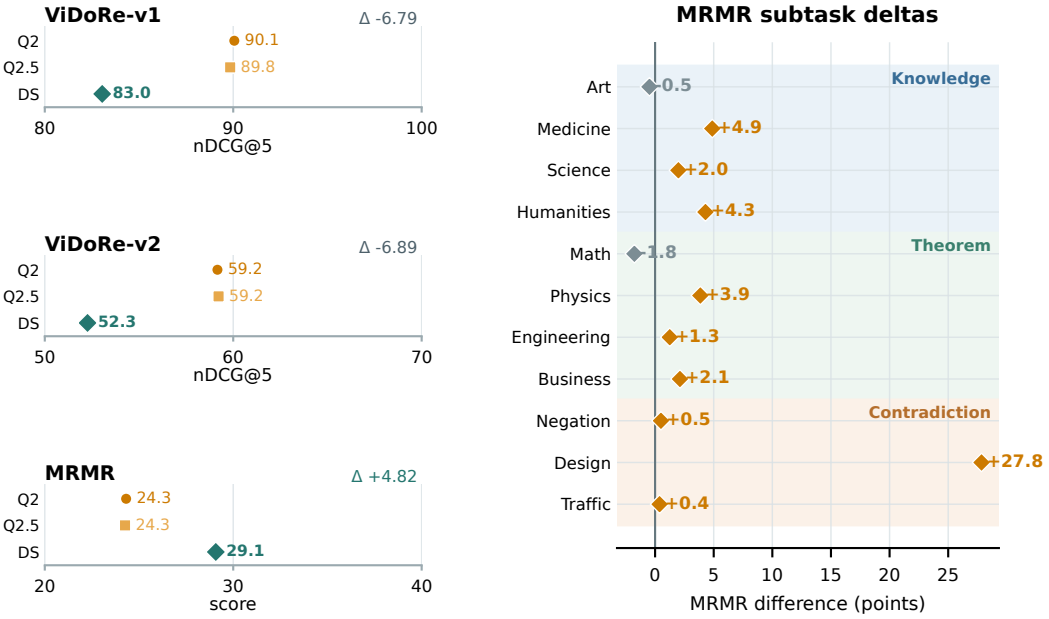


Figure 33: Qwen-family results. Left: benchmark means; the displayed deltas subtract the lower-scoring AR member. Right: ColSDAR-VL's MRMR subtask differences from the strongest Qwen-family AR model, with shaded Knowledge, Theorem, and Contradiction groups.

Table 3 reports all ten ViDoRe-v1 subsets, the four selected ViDoRe-v2 subsets, and all eleven MRMR subtasks. The three English variants of ViDoRe-v2 are excluded from the reported averages. All MRMR scores use the evaluation protocol described in Section 4.1; published leaderboard values are not mixed with these results.

Table 3: Main retrieval results (%). *Mean* averages the subsets within each benchmark. *Overall (Dataset)* averages all 25 subsets; *Overall (Benchmark)* averages the three benchmark means. ViDoRe uses nDCG@5; MRMR follows Section 4.1. Syn-AI/Energy/Gov./Health. denote SyntheticDocQA domains; ESG-Human denotes human-labeled ESG; Know., Thm., and Contr. abbreviate Knowledge, Theorem, and Contradiction. Within each row and family, dark shading with boldface marks the highest score, light shading the second highest, and white the remaining score. Blue denotes the LLaVA family and orange the Qwen family; ranks use unrounded values.

Benchmark	ColLLaVA-1.6–Mistral	ColLLaVA-Llama3	ColLaVi-Da	ColQwen2	ColQwen2.5	ColSDAR-VL
ViDoRe-v1 (nDCG@5)						
ArXivQA	75.96	**80.70**	68.24	88.69	**89.57**	77.81
DocVQA	56.33	**57.19**	54.76	63.21	**65.29**	56.22
InfoVQA	81.51	84.80	**85.31**	**93.69**	93.43	87.16
TabFQuAD	80.11	**80.56**	80.02	92.10	**92.88**	75.94
TAT-DQA	73.86	**76.00**	61.89	**82.24**	81.72	78.46
Shift Project	70.05	**72.19**	65.70	**88.21**	85.85	71.90
Syn-AI	**97.95**	96.79	97.16	98.52	**99.26**	97.58
Syn-Energy	**92.67**	92.43	89.69	96.52	**96.63**	93.14
Syn-Gov.	90.18	91.14	**91.85**	**97.39**	96.58	94.23
Syn-Health.	94.91	94.52	**95.52**	**100.00**	97.16	98.02
Mean	81.35	**82.63**	79.01	**90.06**	89.84	83.05
ViDoRe-v2 (nDCG@5)						
ESG	46.10	**51.35**	47.49	**58.23**	57.75	46.15
Biomedical	44.66	49.31	**49.47**	60.98	**62.79**	51.85
Economics	39.07	43.61	**48.23**	**53.52**	52.64	52.99
ESG-Human	49.72	49.43	**52.18**	**63.92**	63.70	58.08
Mean	44.89	48.43	**49.34**	59.16	**59.22**	52.27
MRMR						
Know.: Art	10.66	12.28	**17.22**	**15.17**	14.40	14.69
Know.: Medicine	16.17	22.84	**27.69**	32.75	31.44	**37.62**
Know.: Science	23.82	34.52	**45.40**	52.19	52.92	**54.90**
Know.: Humanities	18.03	24.31	**29.91**	33.01	34.54	**38.84**
Thm.: Math	6.17	11.56	**12.55**	17.79	**18.01**	16.25
Thm.: Phys.	12.70	15.41	**15.51**	24.99	24.94	**28.85**
Thm.: Eng.	12.63	14.35	**16.02**	21.26	20.94	**22.51**
Thm.: Bus.	28.60	31.36	**35.24**	37.83	35.16	**39.95**
Contr.: Neg.	3.00	1.50	**5.50**	3.50	0.50	**4.00**
Contr.: Design	20.27	21.18	**27.50**	13.51	13.78	**41.62**
Contr.: Traffic	16.57	**17.84**	16.35	15.49	20.22	**20.60**
Mean	15.33	18.83	**22.63**	24.32	24.26	**29.08**
Overall (Dataset)	46.47	49.09	**49.46**	**56.19**	56.08	54.38
Overall (Benchmark)	47.19	49.96	**50.33**	**57.85**	57.77	54.80

E.2 Additional ViDoRe Retrieval Metrics

Tables 4–6 complement the nDCG@5 results in Table 3 with nDCG, Recall, and Precision at additional retrieval depths. Each table uses the same six main checkpoints and separately averages the ten ViDoRe-v1 subsets and the four selected ViDoRe-v2 subsets. The latter comprise ESG, Biomedical, Economics, and ESG-Human; the three English variants are excluded. Subsets receive equal weight regardless of query count. We aggregate the stored per-subset scores before rounding and report percentages, with duplicate exports counted once. Boldface identifies the highest unrounded mean within each model family, including exact ties.

Metric interpretation. The cutoff k is the number of top-ranked documents evaluated. nDCG measures ranking quality with position discounts and normalization by the ideal ranking; Recall measures the fraction of relevant documents retrieved, while Precision measures the relevant fraction of the top-k results. These quantities are complementary and should not be averaged together. In particular, ViDoRe-v1's QA evaluation assigns one relevant document to each query, so Precision@k equals Recall@k divided by k, up to the rounding of the stored scores. A decrease in Precision as k grows therefore does not by itself imply worse retrieval coverage. ViDoRe-v2 permits multiple relevant documents, so Recall and Precision need not follow this relation.

Table 4: nDCG@k (%), averaged equally over ten v1 or four selected v2 subsets.

Model	@1	@3	@10	@20	@50	@100
ViDoRe-v1 (10 subsets)						
ColLLaVA-1.6–Mistral	72.99	79.87	82.43	83.18	83.75	83.95
ColLLaVA-Llama3	74.71	81.25	83.71	84.38	84.92	85.06
ColLaViDa	70.77	77.62	80.34	81.04	81.74	81.98
ColQwen2-7B	84.56	89.20	90.62	90.91	91.13	91.25
ColQwen2.5-7B	83.80	89.08	90.32	90.63	90.93	91.04
ColSDAR-VL	75.15	81.35	84.19	84.65	84.97	85.28
ViDoRe-v2 (4 subsets)						
ColLLaVA-1.6–Mistral	42.23	42.96	47.69	51.19	55.68	58.15
ColLLaVA-Llama3	43.59	46.70	51.34	55.18	59.49	61.54
ColLaViDa	48.26	47.56	51.80	54.79	58.84	61.07
ColQwen2-7B	57.48	57.64	62.00	65.04	68.57	70.23
ColQwen2.5-7B	57.01	57.98	62.18	65.17	68.79	70.51
ColSDAR-VL	50.44	50.06	54.75	57.88	61.91	63.93

Table 5: Recall@k (%), averaged equally over ten v1 or four selected v2 subsets.

Model	@1	@3	@5	@10	@20	@50	@100
ViDoRe-v1 (10 subsets)							
ColLLaVA-1.6–Mistral	72.99	84.61	88.24	91.56	94.54	97.39	98.64
ColLLaVA-Llama3	74.71	85.75	89.12	92.44	95.05	97.74	98.63
ColLaViDa	70.77	82.39	85.76	89.83	92.58	96.11	97.55
ColQwen2-7B	84.56	92.44	94.53	96.25	97.41	98.50	99.24
ColQwen2.5-7B	83.80	92.67	94.49	95.97	97.21	98.67	99.38
ColSDAR-VL	75.15	85.56	89.65	93.15	94.89	96.52	98.38
ViDoRe-v2 (4 subsets)							
ColLLaVA-1.6–Mistral	20.88	36.25	44.48	54.89	66.41	79.77	88.16
ColLLaVA-Llama3	22.30	39.36	47.44	58.30	70.92	83.88	90.68
ColLaViDa	22.54	37.51	47.09	57.67	68.33	80.85	88.86
ColQwen2-7B	28.74	46.29	55.22	67.37	78.51	88.99	94.26
ColQwen2.5-7B	28.55	46.40	55.68	67.74	77.97	89.18	94.94
ColSDAR-VL	24.33	39.39	49.35	60.52	71.84	85.18	92.09

Table 6: Precision@k (%), averaged equally over ten v1 or four selected v2 subsets.

Model	@1	@3	@5	@10	@20	@50	@100
ViDoRe-v1 (10 subsets)							
ColLLaVA-1.6–Mistral	72.99	28.20	17.65	9.16	4.73	1.95	0.99
ColLLaVA-Llama3	74.71	28.58	17.82	9.24	4.75	1.95	0.99
COLLAVIDA	70.77	27.46	17.15	8.98	4.63	1.92	0.98
ColQwen2-7B	84.56	30.81	18.91	9.63	4.87	1.97	0.99
ColQwen2.5-7B	83.80	30.89	18.90	9.60	4.86	1.97	0.99
COLSDAR-VL	75.15	28.52	17.93	9.32	4.74	1.93	0.98
ViDoRe-v2 (4 subsets)							
ColLLaVA-1.6–Mistral	42.23	28.64	23.83	17.27	12.31	7.57	4.83
ColLLaVA-Llama3	43.59	31.96	26.19	19.14	13.64	8.13	5.06
COLLAVIDA	48.26	33.23	27.40	19.72	13.57	8.04	5.01
ColQwen2-7B	57.48	39.70	32.61	23.33	15.68	8.82	5.32
ColQwen2.5-7B	57.01	39.99	32.50	23.49	16.00	8.95	5.40
COLSDAR-VL	50.44	34.51	29.25	21.11	14.51	8.33	5.13

F READOUT AND ADAPTATION CONFIGURATIONS

F.1 READOUT SELECTION

Overall is the arithmetic mean of the ViDoRe-v1, ViDoRe-v2, and MRMR means, which first average their 10, 4, and 11 included subsets, respectively. It equals Overall (Benchmark) in Table 3, not Overall (Dataset). Full uses the designated COLLAVIDA main checkpoint. Each alternative mode uses the configuration with the highest Overall among its recorded LaViDa configurations on these benchmarks. That configuration is held fixed across all four panels of Figure 4; benchmark-specific or subset-specific maxima are not substituted. All models use the LaViDa-L backbone and ColPali training collection, and differences are computed before rounding.

Selected configurations. The four selected configurations use five epochs, a learning rate of 2×10^{-4}, rank-64 LoRA, and per-device microbatch size four. Full follows Table 2, with four accumulation steps, cached retrieval adaptation, and prompt-denoising weight $\lambda_{\mathrm{pd}} = 0.5$. Mean-pool and Mask-$k$ use the earlier retrieval recipe without this cached-denoising combination. Mean-pool retains the 128-dimensional projection, and Mask-k uses $k = 16$. Refine uses cached adaptation with $k = 512$, $T = 2$, and Gumbel token selection. These settings describe the compared retrieval systems, including their readout and adaptation choices.

Position conventions. Algorithm 2 specifies each readout's processing order. For mask-based readouts, the extractor selects positions from the original token IDs and applies those indices to the expanded hidden-state sequence. Refine right-aligns intermediate logits to the token-ID width but retains the original-ID indices for its final readout. The reported comparisons follow these implementation conventions.

F.2 AUXILIARY HYPERPARAMETER CONVENTIONS

Figure 6 compares prompt-denoising weights $\lambda_{\mathrm{pd}} \in \{0, 0.3, 0.5, 0.7\}$ for COLLAVIDA and query-decorrelation weights $\lambda_{\mathrm{qd}} \in \{0, 0.1\}$ for COLSDAR-VL. Each panel reports the change from its corresponding zero-weight configuration in percentage points, without independently rescaling the metrics. ViDoRe-v1, ViDoRe-v2, and MRMR first average their 10, 4, and 11 included subsets; Overall then averages the three benchmark scores equally. The primary coefficients are 0.5 and 0.1, respectively, and the remaining main-model settings are specified in Table 2. All differences are computed before rounding, with duplicate exports counted once.

G Additional Efficiency Results and Protocol

G.1 Recorded Protocol and Measurement Scope

Models, hardware, and inputs. We evaluate the six main checkpoints with merged adapters, their native processors, Full readout, and 128-dimensional retrieval vectors. The records identify one NVIDIA A100-SXM4-80GB GPU, PyTorch 2.5.1+cu121, CUDA 12.1, cuDNN 9.1, Python 3.11.13, and Transformers 5.8.0. BF16 is the runner-configured precision. The diffusion models report eager attention, while the four AR models report FlashAttention-2. Timing therefore reflects each model's native attention implementation, rather than a shared kernel.

All runs identify the same ArxivQA pool of 500 queries and 500 pages. Query batch sizes are $B \in \{1, 8, 32\}$, and page batch sizes are $B \in \{1, 2, 4\}$, where B denotes the number of items in a timed batch. There are ten timed batches per condition: 10/80/320 query instances and 10/20/40 page instances, respectively, rather than 500 timed instances in every condition. The recorded warm-up count is 20, except for query batch 32, which records 16. The shared dataset path and first-five-example metadata agree across models; these are the recorded input-consistency checks. All tested conditions complete without recorded errors or out-of-memory events. The sweep characterizes the specified batch sizes, not maximum capacity.

Reported quantities. Tables 7 and 8 summarize the raw wall-clock arrays. Forward, preprocessing, and total encoding times are aggregated separately; adding their medians would not reproduce a median total. Model-loading time is recorded separately and is not included in these summaries. Allocated and reserved memory are the maxima of the ten recorded observations, converted to GiB. They include the resident model and are not estimates of incremental activation memory. Both main diffusion models record one forward call per query and per page, consistent with the Full interface; training-only regularization does not introduce iterative inference. The reported units are preprocessing, whole-forward encoding, and the scorer workloads defined below, rather than end-to-end request-to-top-k latency or a vision/backbone/projection decomposition.

Table 7: Native query-encoding records. Forward, Total, and Prep. are separate median wall times (ms per batch). Memory columns give maximum allocated/reserved GiB across ten timed batches per row. Total is the recorded encoding time, not full retrieval latency.

Model	B	Forward	Total	Prep.	Alloc.	Reserv.
ColLLaVA-1.6–Mistral	1	30.8	32.8	1.9	13.86	13.88
	8	59.3	62.5	2.9	13.92	13.93
	32	131.1	136.0	4.6	14.10	14.15
ColLLaVA-Llama3	1	31.7	33.9	2.1	14.60	14.62
	8	55.9	59.1	2.9	14.65	14.67
	32	127.9	132.7	4.7	14.83	14.87
ColLaViDa	1	40.8	43.1	0.5	15.81	15.83
	8	57.6	59.2	1.4	16.05	16.08
	32	187.0	190.5	3.4	16.84	16.94
ColQwen2-7B	1	29.3	33.2	3.9	14.46	14.47
	8	59.7	65.5	5.6	14.61	14.64
	32	144.5	153.5	9.0	15.14	15.16
ColQwen2.5-7B	1	28.4	35.3	3.9	14.46	14.47
	8	56.6	62.2	5.5	14.61	14.64
	32	140.5	149.8	9.3	15.14	15.16
ColSDAR-VL	1	57.5	58.5	0.9	16.10	16.12
	8	319.3	323.2	3.8	16.10	16.12
	32	1213.9	1228.1	13.5	16.11	16.13

G.2 Absolute and Output-Normalized Encoding Cost

At batch one, ColLaViDa and ColSDAR-VL have median total query-encoding times of 43.1 and 58.5 ms, compared with forward-only times of 40.8 and 57.5 ms. All four AR baselines have lower query-forward medians. The batch-size records also differ substantially: at batch 32, forward times are 187.0 ms for ColLaViDa and 1,213.9 ms for ColSDAR-VL, while the AR baselines range from 127.9 to 144.5 ms. Single-pass encoding and batched throughput are distinct properties. The batch-size comparison uses ten timed batches per setting, so the number of encoded inputs grows with batch size.

There is also an implementation-specific batching consideration. The current SDAR adapter uses a batched language-model call for unpadded inputs, but handles a padded two-dimensional attention mask by removing padding, encoding each example separately, and repadding the hidden states. The query records at batch 8 and 32 have nonzero padding fractions. This path can limit batch parallelism without introducing iterative generation for any individual query. This implementation path is relevant to interpreting native batching behavior; its contribution to the measured runtime is not separately quantified.

Table 8: Native page-encoding records. Forward, Total, and Prep. are separate median wall times (ms per batch). Memory columns give maximum allocated/reserved GiB across ten timed batches per row. Total is the recorded encoding time, not full retrieval latency.

Model	B	Forward	Total	Prep.	Alloc.	Reserv.
ColLLaVA-1.6–Mistral	1	220.4	305.4	95.9	14.25	14.28
	2	422.2	495.5	76.8	14.63	14.67
	4	804.5	938.6	148.9	15.40	15.47
ColLLaVA-Llama3	1	221.3	282.7	63.8	14.98	15.01
	2	413.5	565.2	128.6	15.37	15.40
	4	807.6	979.4	103.6	16.13	16.20
ColLaViDa	1	765.1	928.1	119.5	21.75	21.86
	2	1705.7	1997.9	219.7	27.91	28.31
	4	3510.8	3930.8	461.1	39.98	40.75
ColQwen2-7B	1	123.3	217.1	89.8	14.71	14.74
	2	245.6	386.0	119.8	14.98	15.01
	4	418.8	577.4	148.6	15.51	15.57
ColQwen2.5-7B	1	121.7	224.3	85.1	14.71	14.74
	2	241.1	331.5	117.9	14.98	15.01
	4	427.3	569.6	137.9	15.51	15.57
ColSDAR-VL	1	594.8	745.5	131.9	18.44	18.56
	2	1208.6	1436.1	263.1	19.09	19.21
	4	2580.7	3133.8	570.4	22.11	22.31

Figure 34 preserves every batch-one page observation. Both diffusion models incur higher absolute page-forward costs than their family baselines. At this batch size, peak allocated memory is 21.75 GiB for ColLaViDa and 18.44 GiB for ColSDAR-VL, compared with 14.25/14.98 GiB for the LLaVA AR baselines and 14.71 GiB for either Qwen baseline. The larger page batches fit in the tested device, but do not reverse the diffusion models' higher absolute page-encoding cost.

For an output-normalized description, let $\mathcal{J}$ index the ten timed pages at batch one, χ_j be page j's recorded forward time in milliseconds, and n_j its retained vector count. The page representation is $\mathbf{D}_j \in \mathbb{R}^{n_j \times r}$ with $r = 128$, as in the main formulation. We compute

$$\kappa = \mathrm{median}_{j \in \mathcal{J}} \left(\frac{10^3 \chi_j}{n_j} \right), \qquad |\mathcal{J}| = 10, \tag{14}$$

in milliseconds per 1,000 retained vectors. Pairing time and vector count within each observation is essential: a ratio of their separate medians is a different estimator. Figure 34 shows the individual ratios, their median, linearly interpolated quartiles, and full range. These summaries describe page-level dispersion, with quartiles and full ranges rather than confidence intervals.

For ColSDAR-VL, $\kappa = 126.1$, versus 165.1 for ColQwen2 and 161.0 for ColQwen2.5. The corresponding LLaVA values are 93.3 for the Mistral baseline, 96.6 for ColLLaVA-Llama3, and 257.1 for ColLaViDa. Thus, ColSDAR-VL has lower output-normalized cost than its Qwen baselines, but not than every evaluated system; ColLaViDa has no normalized-cost advantage in this comparison. Moreover, retained output count is not a matched input-compute budget. The ratio includes native vision and language processing, and does not measure equal-resolution encoders producing equally useful vectors. The diffusion records' nominal expanded-token field reflects short prompt counts rather than the complete visual expansion, so it is not used as a visual-input budget. Accordingly, κ characterizes the evaluated systems at their native settings, rather than an isolated backbone cost.

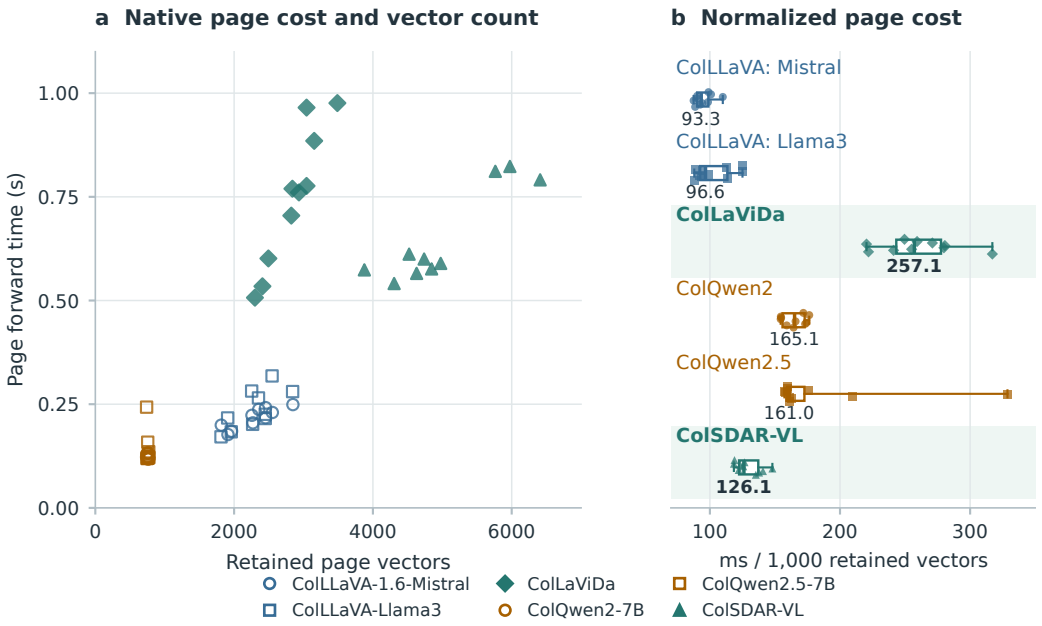


Figure 34: Native page-encoding measurements for all six systems, with ten batch-one pages per model. **(a)** Retained vector count versus absolute forward time. **(b)** Per-page normalized time from Eq. 14; labels give medians, boxes quartiles, whiskers full ranges, and points individual pages. No observation is removed. These ratios do not isolate architecture costs.

G.3 Vector Storage and Recorded Scoring Workloads

For N indexed pages with n_j vectors per page, vector dimension r, and β bytes per scalar, raw vector storage is

$$M_{\text{raw}} = \beta r \sum_{j=1}^{N} n_j, \qquad N = 500, \quad r = 128, \quad \beta \in \{2, 4\}. \tag{15}$$

Table 9 applies this accounting identity to all 500 pages. This population differs from the ten pages used in the batch-one encoding summaries. FP32 totals are 690.02 MiB for ColLaViDa and 1,187.91 MiB for ColSDAR-VL, versus 549.30/548.32 MiB for the LLaVA AR baselines and 180.24 MiB for either Qwen baseline. FP16 accounting halves these values; it does not establish the accuracy or speed of an FP16 retrieval index. These totals exclude padding, metadata, and search-index overheads, and are not measurements of serialized files or GPU residency.

The scorer records contain five repeats of a workload with 100 queries, 500 pages, and query batch 32. Median workload times are 683.7 ms for ColLaViDa, 546.9/546.7 ms for its AR baselines, 1,244.3 ms for ColSDAR-VL, and 160.7/160.6 ms for the Qwen baselines. These values describe the recorded scorer workloads. Dividing a workload time by 100 yields amortized batched scoring

Table 9: Storage accounting and recorded scorer workloads for 500 pages. Vectors is the mean retained count per page; FP32/FP16 are raw unpadded MiB at 128 dimensions, not measured index files. Score is the median ms for all 100 queries over five repeats, with query batch 32. Scores describe the recorded scorer workloads.

Model	Vectors	FP32	FP16	Score
ColLLaVA-1.6–Mistral	2249.9	549.30	274.65	546.9
ColLLaVA-Llama3	2245.9	548.32	274.16	546.7
COLLAVIDA	2826.3	690.02	345.01	683.7
ColQwen2-7B	738.2	180.24	90.12	160.7
ColQwen2.5-7B	738.2	180.24	90.12	160.6
COLSDAR-VL	4865.7	1187.91	593.96	1244.3

cost, not single-query latency. In particular, adding that quantity to batch-one encoding time would not give a measured end-to-end retrieval request. We report encoding and scorer costs separately; these workloads are not presented as serving or top-k latency measurements.

Quality–cost interpretation. The measurements establish a native quality–cost trade-off: the two main models improve their family's MRMR score without iterative inference, while requiring more page-encoding time and vector storage. The favorable output-normalized result for COLSDAR-VL does not remove those absolute costs. As discussed in Appendix D.4, LaViDa prefix caching and SDAR block-wise generation speedups do not automatically transfer to Full embedding extraction. The reported measurements concern retrieval inference at native token budgets: absolute encoding time, output-normalized cost, memory, and storage describe complementary aspects of deployment.

H LIMITATIONS AND OUTLOOK

Evidence-preserving scalability. Full-token readout preserves independently matchable evidence, but its representation size grows with the expanded page sequence. Scaling to large collections therefore requires controlling index storage and search cost as well as encoding latency (Appendix G). A long-term direction is to jointly design retrieval representations and indexes: learn content-adaptive region aggregation, train quantization-aware embeddings, and distill full-token rankings into budget-adaptive retrievers. The central challenge is to reduce redundant vectors without discarding small, decisive evidence such as a negation or a table entry. Progress should be assessed through retrieval quality, index size, and end-to-end search latency together, rather than vector count alone.

Retrieval-oriented diffusion pretraining. MM-DIFFEMBED adapts existing multimodal diffusion backbones rather than pretraining their visual and language components jointly for retrieval. The backbone-dependent balance between ViDoRe and MRMR performance motivates studying how pretraining can support both semantic matching and reasoning-intensive evidence selection. One direction is to combine masked modeling with region–text correspondence and page-level contrastive alignment, so that contextualization learns which local distinctions matter for ranking. Developing this approach would require suitable large-scale pretraining corpora, alignment supervision, and coordinated optimization of the visual frontend and language backbone. A central goal is better transfer across retrieval tasks while retaining independent, single-pass encoding.

From page relevance to connected evidence. The present formulation ranks individual pages independently. Although each page vector can incorporate contextual information, `MaxSim` does not explicitly model whether several retrieved pages jointly complete an evidence chain. Extending the framework to document-level and multi-hop retrieval therefore calls for representations of relations across pages, not merely longer inputs. A feasible direction is a hierarchical index that retains independently encoded page vectors while adding document summaries and cross-page links for subsequent evidence selection. Developing this extension would require annotated cross-page evidence chains and evaluation of evidence completeness alongside page-relevance metrics.

I AI USE STATEMENT

Claude Opus 4.6 was used for language polishing and grammar checking. OpenAI Codex assisted with manuscript drafting and translation, literature organization, development of theoretical arguments and proof sketches, experiment-planning suggestions, interpretation of recorded results, and LaTeX and visualization code. Reported numerical results are drawn from the saved experimental records, rather than generated by these assistants. The authors are responsible for the accuracy, originality, citations, and conclusions of the paper, including all AI-assisted content.

J ETHICS STATEMENT

This study develops retrieval models using existing research datasets and pretrained backbones; it does not recruit human participants or collect new personal records. The public document pages and queries remain subject to their original licenses, attribution requirements, and access conditions. Anonymized code will reference dataset sources rather than redistribute materials without permission. Benchmark documents and model representations may retain social biases, sensitive information, or copyrighted content. Retrieval also makes such material easier to locate, so deployment should preserve document-level access controls and review dataset provenance. Errors on medical, traffic, or technical queries can surface misleading evidence; the models retrieve pages rather than verify their correctness and are not substitutes for professional judgment. The experiments use an A100-80GB cluster; the inference measurements characterize computational cost, not a carbon-emissions assessment.

K REPRODUCIBILITY STATEMENT

We will provide anonymized code with configuration files, training and evaluation entry points, and result-aggregation scripts. Section 3 defines the encoders and objectives; Appendix C documents data,

optimization, visual processing, and the modified MRMR protocol. Appendix C.4 maps the core files to the equations and Algorithms 1 and 2. Appendix D provides assumptions and proofs, Appendix E supplies detailed retrieval metrics, Appendix F specifies readout and coefficient conventions, and Appendix G records timing, memory, and storage definitions. The release will preserve the reported configurations and aggregation rules while removing identifying paths and metadata.