

DataComp-VLM: Improved Open Datasets for Vision-Language Models

Matteo Farina^{*1,2} Vishaal Udandarao^{*2,3} Thao Nguyen^{*15} Selim Kuzucu^{†,5} Maximilian Böther^{‡,7}
 Andreas Hochlehner^{†,2} Adhiraj Ghosh^{†,2} Marianna Nezhurina^{†,8,9} Karsten Roth^{†,10}
 Joschka Struber², Yuhui Zhang⁴, Sebastian Dziadzio², Elaine Sui⁴, Soumya Jahagirdar²,
 Dhruba Ghosh⁴, Hasan Hammoud¹¹, Thomas De Min¹, Simone Caldarella¹, Jehanzeb Mirza¹²,
 Sedrick Keh¹³, Mehdi Cherti^{8,9}, Hilde Kuehne², Bernt Schiele^{5,6}, Serena Yeung-Levy⁴,
 Muhammad Ferjad Naeem⁶, Federico Tombari⁶, Ana Klimovic⁷, Elisa Ricci^{1,14}, Matthias Bethge²,
 Sewoong Oh¹⁵, Ameya Prabhu², Alessio Tonioni⁶, Jenia Jitsev^{8,9}, Massimiliano Mancini¹
 Ludwig Schmidt^{‡,4} Nikhil Parthasarathy^{‡,10}

*Project leads †Core contributors ‡Equal supervision

🔗 Code: <https://github.com/mlfoundations/dcvlm>
 🏠 Website: <https://www.datacomp.ai/dcvlm>

Abstract

Building performant Vision-Language Models (VLMs) requires carefully curating large-scale training datasets, yet the community lacks systematic benchmarks for evaluating such curation strategies. We introduce DataComp for VLMs (DCVLM), a benchmark for controlled data-centric experiments to improve VLM training. As part of DCVLM, we collect 160 datasets spanning four data types—image-caption pairs, multimodal interleaved documents, text-only, and instruction-tuning data—into a corpus of 6T multimodal tokens. DCVLM allows participants to test curation strategies (filtering, mixing, formatting, sampling) across 1B–8B models and 6.25B–200B token budgets. Models are then evaluated on a carefully selected suite of up to 52 downstream benchmarks across 9 domains. We conduct extensive experiments on DCVLM and find that data mixing, *not* filtering, is key to a high-quality training dataset: instruction-heavy mixtures *scale better* than caption-heavy ones, with gains widening at larger scales. The resulting dataset, DCVLM-BASELINE, enables training an 8B VLM to 63.6% accuracy on our 33-task core suite with 200B training tokens. Compared to FINEVISION, the state-of-the-art open VLM training dataset, this represents an improvement of +5.4pp. DCVLM and all accompanying artifacts will be made publicly available [here](#).

1 Introduction

The performance of foundation models is fundamentally shaped by the composition and quality of their pretraining¹ data [67, 73, 79, 83, 158, 224, 232, 233, 251, 285]. This has led to a rise of systematic studies of pretraining data curation, including DataComp [73] for contrastive vision-language models, DCLM [158], Nemotron-CC [267], and FineWeb [233] for language models, and OlmoASR [222] in the speech domain. The core design principle of these works is to fix model

¹University of Trento, ²Tübingen AI Center, University of Tübingen, ³University of Cambridge, ⁴Stanford University, ⁵Max Planck Institute for Informatics, ⁶Google, ⁷ETH Zürich, ⁸LAION, ⁹Juelich Supercomputing Center (JSC), Research Center Juelich (FZJ), ¹⁰Google DeepMind, ¹¹KAUST, ¹²MIT, ¹³Toyota Research Institute, ¹⁴Fondazione Bruno Kessler (FBK), ¹⁵University of Washington

¹The VLM literature is highly fragmented in its terminology, with stages variously called “pretraining,” “alignment,” or “instruction tuning” depending on the data type used. We refer to VLM “pretraining” as in [41, 300, 365], i.e., the first or only multimodal training stage of a VLM, starting from an independently pretrained vision encoder and language model, bridged by a randomly initialized connector [175]. When a VLM is trained in multiple stages, “pretraining” denotes the first stage; otherwise, it denotes the sole training cycle.

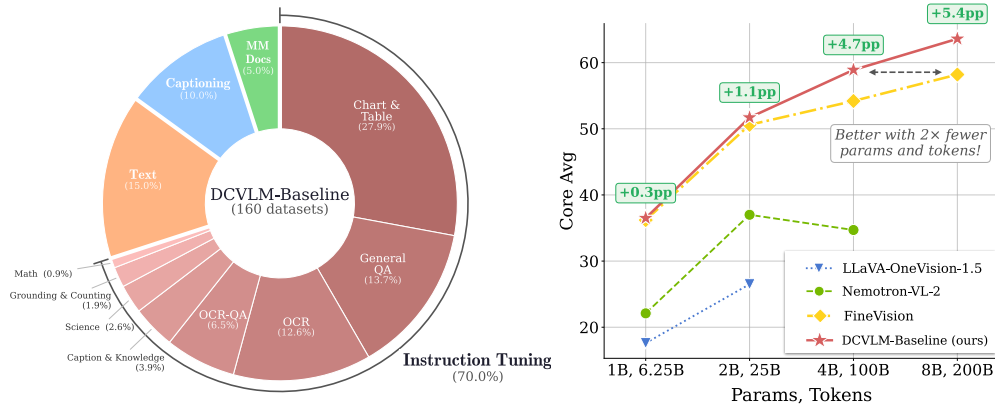


Figure 1: **DCVLM-BASELINE outperforms open VLM training datasets.** DCVLM-BASELINE (left) combines 160 sources as 10% image-caption pairs, 5% multimodal documents, 15% text-only, and 70% multimodal instruction-tuning data. On our 33-evaluation Core set (right), it outperforms existing datasets [12, 58, 310] across all scales. Notably, a 4B model trained on DCVLM-BASELINE for 100B tokens beats an 8B model trained on FINEVISION for 200B tokens, a 4× compute reduction.

architecture and pretraining procedure while varying *only* the data, enabling isolated measurement of data-centric interventions. However, progress in *autoregressive* vision-language models (VLMs) has mostly focused on novel architectures [3, 19, 21, 57, 58, 103, 275, 276, 313, 320, 325], training recipes [12, 44, 87, 159, 175–177, 189, 203, 213, 265, 279, 300, 365], or evaluation protocols [64, 78, 118, 350], treating data as a second-class citizen. The data curation strategies behind their success (which datasets to include, how to filter them, what ratios to mix them in) remain poorly understood and largely irreproducible [12, 44, 117, 213, 276, 279, 300, 365]. Our goal is precisely to fill this gap and enable *open* data curation research for the latest class of modern autoregressive VLMs.

Several factors make VLM data curation more challenging compared to other domains. **First**, unlike early text or vision-language models that often train directly on raw web crawls (e.g., CommonCrawl), modern VLMs are typically trained by aggregating existing datasets from a wide variety of data types—web-crawled image-caption pairs, interleaved multimodal documents, text-only corpora, and multimodal instruction-tuning data—that differ in quality and downstream utility. Because these datasets have already undergone varying degrees of upstream curation, what actually drives quality under this aggregation-based regime—filtering, mixing ratios, or something else—remains an open question. Indeed, existing models largely sidestep it, drawing on a single data type [175] or, at most, an ad-hoc subset [8, 17]. **Second**, existing open training datasets [12, 57, 58, 279, 310, 361] operate at the scale of millions of samples, far below the trillions of tokens used by state-of-the-art (SoTA) models [16, 275, 277, 300]. This limits the scope of curation experiments that can be conducted. **Third**, the interaction between data types, model scale, and training budget creates a design space that is too large for exhaustive experimentation. **Fourth**, VLM evaluation lacks standardization [118]: different papers use different benchmark suites, making fair comparisons across datasets difficult.

To mitigate these challenges and enable controlled comparisons, we introduce DataComp for VLMs (DCVLM), the first benchmark designed to systematically study data curation strategies within a realistic VLM-practitioner’s paradigm. DCVLM provides the following:

1. A **standardized data pool** of 160 existing datasets spanning four data types: image-caption pairs, multimodal documents, text-only data, and (multimodal) instruction-tuning data. Our pool contains 6T multimodal tokens, enabling a diverse range of data-centric experiments.
2. A **principled scaling ladder** spanning 1B–8B model parameters and 6.25B–200B training tokens. This enables researchers to test curation strategies across a wide range of compute scales.
3. A **comprehensive evaluation protocol** with 52 downstream benchmarks organized across 9 domains, split into validation, core, and extended tiers, filtered for stability and reliability.

Using our benchmark, we conduct more than 1,000 experiments yielding multiple findings, including:

Mixing, not filtering, is the dominant lever. Recent VLM technical reports often apply additional downstream quality filters (e.g., CLIP-score or image quality) on top of existing public datasets [277,

335, 345]. Yet, through controlled experiments with common quality filters, we find that such downstream filtering provides diminishing, and sometimes negative, returns (Sec. 4.1). We trace this to the modern data landscape: unlike models trained directly on raw web crawls (e.g., CommonCrawl), today’s VLM datasets have already undergone moderate to significant upstream curation. While curating VLM training data directly from raw pools remains an important direction for future work, our results show that applying additional filters to already-curated data is largely ineffective. In contrast, optimizing mixture ratios, which specifically interpolate instruction-tuning and image-caption proportions, yields significant, scale-dependent gains: instruction-heavy mixtures scale better than caption-heavy ones, and this gap widens with model size and token budget (Sec. 4.2).

Pretraining decisions reliably transfer after supervised fine-tuning and across backbones. We show that pretraining performance predicts post-SFT performance with near-perfect fidelity (Pearson $r = 0.99$ across 54 SFT runs), and that our findings are robust to the choice of LLM initialization, i.e., initializing from Qwen2.5-Base or Qwen2.5-Instruct [240] produces similar data rankings. This validates the use of pretraining-only metrics for data curation research with DCVLM (Sec. 4.3).

Our controlled experiments yield DCVLM-BASELINE, a new state-of-the-art open VLM training dataset (Fig. 1). At the x-large scale (8B model, 200B tokens), a DCVLM-BASELINE-trained model achieves 63.6% on our core set, outperforming FINEVISION [310], the previous best open dataset, by +5.4 pp (Sec. 5). We release the full data pool, evaluation suite, model checkpoints at four scales, and all experimental infrastructure to serve as a reproducible testbed for future research.

2 Related Work

Vision-Language Pretraining. Modern VLMs adopt a modular architecture consisting of a pretrained vision encoder, a language model backbone, and a connector [12, 16, 17, 151, 175, 255, 300, 365]. Originally, “pretraining” involved training the connector on large-scale data, predominantly image-caption pairs [12]. In contrast, recent SoTA models train all parameters [300, 365] and incorporate diverse data types. However, the precise mixture ratios, filtering criteria, and formatting choices largely remain proprietary and poorly documented across leading VLMs, motivating our benchmark.

Benchmarking Data Curation. DataComp [73] and DataPerf [212] established the paradigm of fixing model architecture and training procedure while *varying only data*. DCLM [158] extended this paradigm to language models, demonstrating that a fasttext classifier trained on high-quality samples can substantially boost performance. FineWeb [233] and its educational-quality variant, FineWeb-Edu, showed similar gains through filtering. Generally, quality-based filtering has shown strong results for text [158, 233] and image-text pairs [73, 301]. Common approaches include CLIP-score filtering [101], image quality assessment [201, 312], text quality classifiers [58, 267], and multimodal quality estimators [301]. Beyond filtering, prior works have also explored data mixing approaches such as domain weighting [227, 317], mixture optimization [18, 37, 59, 121, 179, 330], and temperature-sampling [45, 57]. Despite recent released datasets (e.g., FINEVISION [310], CAULDRON [142]), there exists no systematic study on filtering and mixing strategies in the VLM setting. Our work fills this gap by providing the first *scale-aware* study of data curation for VLMs.

3 The DCVLM Benchmark

DCVLM provides a controlled framework (Fig. 2) for constructing VLM training sets. We fix the model and training recipe, and participants propose ways to filter and mix data from our pool. We next describe the pool (Sec. 3.1), training recipe (Sec. 3.2), scales (Sec. 3.3), and evaluation (Sec. 3.4).

3.1 Data Pool Construction

Our data pool aggregates 160 publicly available datasets organized into four data types. For the full list of source datasets, pool composition, visualizations, and sample and token counts, see Sec. E.

① **Image-caption pairs** form the largest component, with great variability in their constituent datasets. At one end, sources like DataComp-1B [73] and ReLAION-2B [251] provide abundant, CLIP-score-filtered image-alt-text pairs from web crawls. At the other, datasets like the synthetic ShareGPT-4o [51] and the human-annotated Pixmo-Cap [57] offer fewer yet higher quality samples.

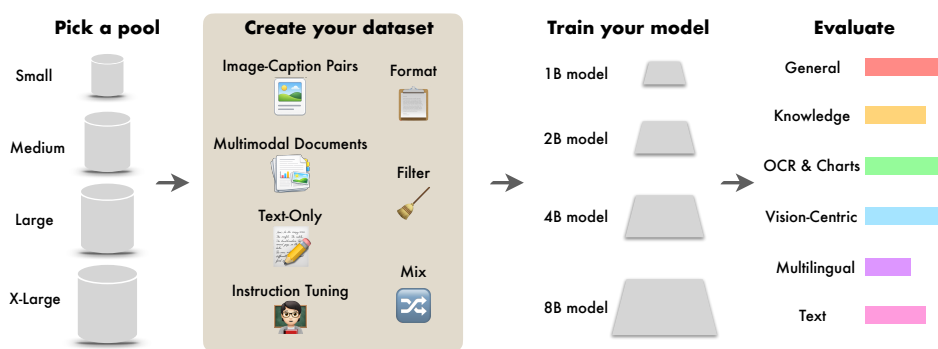


Figure 2: **DCVLM allows researchers to construct effective multimodal datasets.** Participants can choose one of four scales (small, medium, large, and x-large) according to their compute availability. We provide tools to format, filter, and mix the data pool so that participants can create their own datasets. The resulting datasets are then used to train an autoregressive VLM using a fixed training recipe. Models are comprehensively evaluated across a broad spectrum of capabilities.

② **Multimodal interleaved documents** consist of web-crawled interleaved image-text sequences as they appear on websites, PDF documents, and academic papers. Sources include MINT-1T-HTML [14], MINT-1T-PDF [14], WanJuan [94], OmniCorpus [161], and Multimodal-Textbook [355]. These are the least curated sources in our pool. Most are scraped directly from the web with minimal URL-based and heuristic filtering, and thus tend to have lower quality scores on average.

③ **Text-only data** preserve the language model’s capabilities during multimodal training, following recent VLMs [16, 365]. Examples include FLAN [186], SlimOrca [165], and Dolly [48], alongside image-free science and knowledge sources such as Numina-Math-1.5 [156] and xCoder80k [305].

④ **Multimodal instruction-tuning data** comprise single or multi-turn instruction-tuning datasets, typically with human-written or model-generated question-answer pairs grounded in one or multiple images. We manually categorize these into eight capabilities following [310]: knowledge, chart & table understanding, general-QA, grounding & counting, math, naive OCR, OCR-QA, and science. For a complete breakdown of the capability distribution of instruction-tuning data, see Sec. E.1.

Our DCVLM pool contains 6T multimodal tokens (measured using the InternVL-2.5 [41] tokenizer). It is highly heterogeneous in source quality, data types, instruction-tuning capabilities (e.g., grounding, OCR, chart and table understanding, captioning), visual and textual domains (e.g., natural, synthetic, tabular images), and languages (over 20 including English and Chinese, see Sec. E.5). This heterogeneity is *deliberate*: it lets participants study curation recipes in a realistic setup with several confounders to control for. To avoid train-test overlap, we decontaminate our entire pool against our **Extended** eval suite of 52 benchmarks (Sec. 3.4): multimodal samples are filtered with a ResNet-50 SSCD embedding model [235] (cosine-sim > 0.75 to any test image) and text-only samples with MinHash [25] Jaccard similarity (> 0.55). The exact details of decontamination are in Sec. G.

3.2 Model Architecture and Training Recipe

To ensure DCVLM employs a state-of-the-art training recipe, we use an architecture that mimics InternVL3 models [365]: an InternViT-300M vision encoder [41], a 2-layer MLP projector, and a Qwen2.5-Base language model [240] (we show that our central findings transfer to Instruct backbones as well in Sec. 4.3). We adopt *AnyRes* [177] tiling, where images are dynamically split into 448×448 tiles, each encoded into 256 visual tokens after pixel shuffling [41, 257]. We use the AdamW [188] optimizer, a linear 3% warmup, and a cosine decay with peak learning rate of 2×10^{-5} , identified after an initial sweep to ensure optimal hyperparameters. For more details, refer to Secs. C and D.

3.3 Competition Scales and Design Principles

A key principle of DCVLM is to evaluate data curation strategies *across scales*, because findings at small scales may not transfer to larger ones [81, 217, 221, 278]. To simultaneously (i) approach the scale of foundation models like InternVL-3 [365] and (ii) ensure accessibility for researchers with fewer resources, we define four scales: small, medium, large, and x-large. Model sizes and token budgets are illustrated in Tab. 1. We design the small, medium, and large scales such that a step

Table 1: **DCVLM scales.** Each scale specifies model size (N), number of training tokens (D), and token size of the original pool to be used for curation (‘Pool’). We also present the vision encoder and language model that we initialize training runs from, along with compute estimates (‘H100 hrs’).

Scale	N	D	Vision init.	LLM init.	H100 hrs	Pool
small	1B	6.25B	InternViT-300M	Qwen2.5-0.5B	80	187.5B
medium	2B	25B	InternViT-300M	Qwen2.5-1.5B	640	750B
large	4B	100B	InternViT-300M	Qwen2.5-3B	5,120	3T
x-large	8B	200B	InternViT-300M	Qwen2.5-7B	20,480	6T

corresponds to an $8\times$ compute increase: models become $2\times$ larger and tokens increase by $4\times$. At the `x-large` scale, our entire pool of 6T tokens is the candidate for dataset construction. We design all scales to fix pool-to-training token ratio at $30\times$, i.e, the pool always contains $30\times$ more tokens than the training budget. The primary reason for keeping this ratio constant is to enable participants to experiment with aggressive filtering at all scales while hitting a constant number of data repetitions.

3.4 Evaluation Protocol

Participants in DCVLM can evaluate models on up to 52 benchmarks. To get reliable signal, we start from a candidate set of 65 benchmarks, which we categorize across 9 domains based on the majority consensus of prior work [41, 213, 279, 297, 365]: General Understanding, Knowledge-Centric, OCR & Charts, Vision-Centric, Multilingual, Text-Only, Safety, Hallucination, and Reasoning benchmarks. We then filter them for (i) *stability*, removing those with high seed variance [197, 286], and (ii) *monotonicity*, removing those that do not improve from `small` to `medium` scales [97, 233]. We organize benchmarks into three nested tiers, each a superset of the previous: a **Validation set**, used for rapid iteration (13 benchmarks), a **Core set**, the primary tier used for main results (33 benchmarks), and an **Extended set** (52 benchmarks), including all benchmarks for comprehensive analysis. Safety, Hallucination, and Reasoning are deferred to the extended tier, as they are typically targeted by (and thus most relevant for) post-training methods. Unless otherwise specified, we report the average accuracy across all benchmarks in a given tier. For full details of benchmark selection, see Sec. F.

4 Towards a Strong Baseline on DCVLM

We now present a suite of controlled experiments showing how to obtain a strong baseline dataset on DCVLM, along two primary axes: data filtering (Sec. 4.1) and data mixing (Sec. 4.2). For additional axes (including data formatting, synthetic captions, and temperature sampling), refer to Sec. H. We also run control experiments validating the generality of our results (Sec. 4.3). Unless specified, all filtering experiments use a base mixture of 75% image-caption, 18% text-only, 4% multimodal documents, and 3% instruction-tuning data, derived by length-proportional sampling across the pool. In this section, we always report results on our 33-task Core evaluation suite.

4.1 Data Filtering

Quality-based filtering has been central to pretraining strong language [158, 227, 228, 233] and CLIP models [65, 68, 73, 286], hence a natural question is whether these gains transfer to VLMs as well. We answer this question in the *negative* by testing **more than 60 filter configurations** at both `small` and `medium` scales (for an exhaustive report, refer to Sec. H.1). To illustrate our findings, here we report and discuss `medium` scale results for filters shown to be successful in prior work (in Sec. H.1, we describe several other variants across scales, yielding the same conclusions):

- **CLIP-score.** We experiment with filtering image-caption pairs according to three different CLIP models: OpenAI’s CLIP ViT-L/14 [241], DFN-CLIP [68], and SigLIP-2-B/16@384 [282].
- **Text quality classifiers.** We experiment with filtering samples according to the quality of their constituent text snippet(s), as judged by three classifiers: DCLM’s fasttext classifier [158], as well as NVIDIA’s Nemotron and Mixtral educational-quality classifiers [267].
- **Multimodal filters.** We additionally experiment with (i) filtering with two UniFilter models [301] (Qwen2.5-1.5B and Qwen3-0.6B), and (ii) filters grounded in perplexity [13]: text-only perplexity

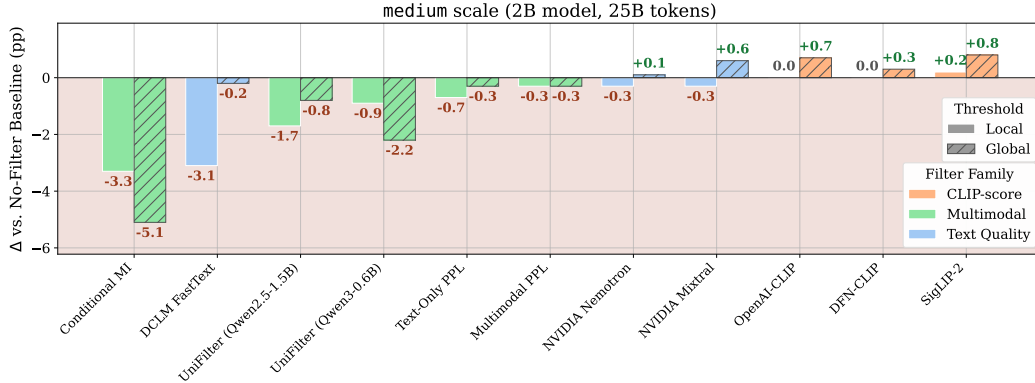


Figure 3: **Filtering rarely helps, but changing the data composition does move performance substantially.** Established data filtering techniques do *not* significantly outperform a no-filter baseline. This observation holds consistently at both the small (Fig. 17) and medium scales of our benchmark. At the same time, inducing a different data mixture via global filtering (hatched bars) leads to significant performance variations compared to locally filtered datasets (solid bars).

(computed on text tokens by *excluding* image tokens), multimodal perplexity (computed on text tokens by *including* image tokens), and Conditional Mutual Information [146], which measures their difference (i.e., the reduction in perplexity with and without image tokens).

Importantly, we study two different filtering paradigms to isolate the impact of data filtering and that of implicit data mixing: ① **Local filtering**, which computes filtering percentile thresholds independently within each source dataset. This preserves the global mixture by construction: every dataset loses the same sample fraction, and ② **Global filtering**, which computes a single filtering threshold across the entire pool of samples to which the filter can be applied. Because different data sources have systematically different score distributions, a global cut implicitly *reshapes the data mixture*. Following prior evidence that smaller models benefit from more aggressive filtering [9, 217], we retain the top-10% of samples at the small scale, and the top-40% at the medium scale.

Fig. 3 illustrates the results. We make two key observations: (i) regardless of whether the mixture is held fixed, no quality filter we tested produces a robust and significant improvement over a no-filter baseline; and (ii) local and global filtering yield notably different results. We expand on each below.

Filtering rarely helps, but why? The best filtering outcome is given by SigLIP-2 when globally filtering image-text pairs (rightmost bar in Fig. 3), yet this result is defined by a marginal +0.8pp improvement, far below the gains one would a priori expect from quality-based filtering [65, 73, 158, 251, 267, 286]. Other filters either leave the baseline mostly unchanged or actively hurt performance. This observation holds across both small and medium scales (see Fig. 17 for the small figure).

This failure is surprising, especially in light of strong results from prior works. We hypothesize this is because *there is no significant noise to remove from our base pool*: unlike raw Common Crawl (used in DCLM [158]) or raw web-crawled image-text pairs (used in DataComp-CLIP [73]), existing VLM training sets aggregate datasets that have already undergone a level of upstream filtering (e.g. CLIP-score filtering) by their original creators, and our pool follows this data collection process. To validate our hypothesis, we create three sub-pools from our original pool, varying the effective percentage (25%, 65%, and 100%) of “pre-filtered” data samples in the mixture (see Sec. H.3 for more details). From each of these sub-pools, we create three more training sets by applying further CLIP-score filtering to the image-caption data. For each pair of datasets, we then train small scale models and measure the performance gain due to downstream

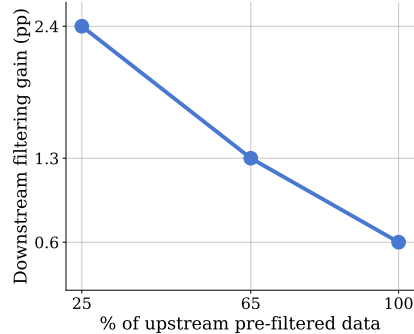


Figure 4: Upstream filtering leads to diminishing returns from additional (i.e., “downstream”) filtering.

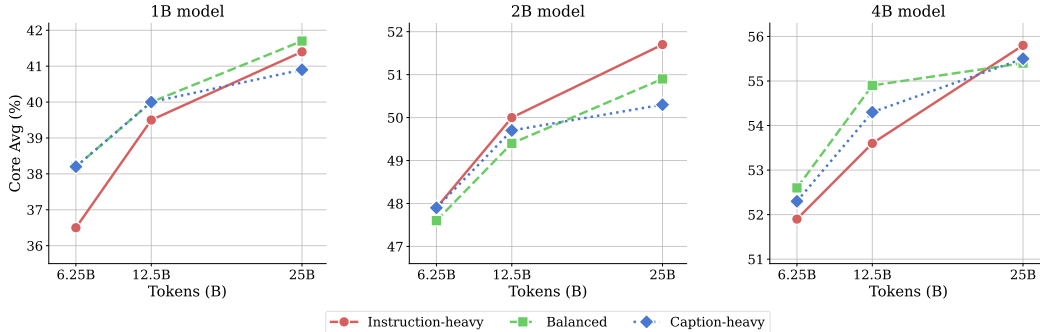


Figure 5: **Instruction-heavy mixtures scale better with compute.** For the 1B model (*left*), the Instruction-heavy mix (red) starts as the worst mixture with 6.25B training tokens, but recovers quickly up to becoming the second-best with 25B training tokens. For the 2B model (*middle*), all mixtures have comparable performance with 6.25B tokens, but performance gains consistently grow in favor of the Instruction-heavy mixture as training tokens grow. For the 4B model (*right*), yet again, the Instruction-heavy mix starts as the worst with 6.25B tokens, and becomes the best at 25B tokens.

filtering (Fig. 4). For 25% pre-filtering (i.e., when the sub-pool is dominated by unfiltered data), the gain is significant (+2.4pp). However, this decreases the more the sub-pool is pre-filtered (dropping to +1.3pp at 65% and +0.6pp at 100%). In other words, *additional filtering on top of already-curated data operates in a regime of diminishing returns.*

Interaction between filtering and implicit mixing. The second takeaway from Fig. 3 is *local and global filtering produce very different results.* The inconsistent trends suggests that global filtering is not a reliable strategy. However, given significant performance fluctuations between local and global filtering, we hypothesize the underlying mixture distribution is the lever that dictates performance.

4.2 Data Mixing

Having established that filtering over the base mixture provides negligible gains on our pool, we turn to data mixing, i.e., the allocation of training samples across data types, as our primary curation lever.

Setup. We optimize the mix along an important axis based on prior work [42, 123, 175, 213, 259, 279, 347]: the ratio of *image-caption pairs* to *instruction-tuning data*. Text-only samples and multimodal documents are fixed at 15% and 5%, respectively, as supporting components. Here, we study three ratios along the image-caption \leftrightarrow instruction-tuning axis: (i) a **Caption-heavy** mixture with 65% image-caption pairs and 15% instruction-tuning data; a (ii) **Balanced** mixture with 40% image-caption and 40% instruction; and (iii) an **Instruction-heavy** mixture with 10% image-caption and 70% instruction-tuning data (see Sec. 1 for finer sweeps across scales). Each mixture is evaluated across a scaling grid of 3 model sizes (1B, 2B, 4B) \times 3 token budgets (6.25B, 12.5B, 25B).

Data mixing cannot be scale agnostic. Fig. 5 reveals a striking interaction between data mixture and compute scale: as both model size and token budget increase, the Instruction-heavy mix exhibits a markedly steeper scaling slope. It starts as the *worst* mixture at 1B \times 6.25B (small scale) but becomes the *best* at 2B \times 25B (medium scale), and remains so at 4B \times 25B. This crossover pattern has an important practical implication: *mixture rankings established at small scale do not transfer reliably to larger scales.* In our setting, optimizing the data mix at the small scale (1B \times 6.25B) would select the Caption-heavy mix and miss the Instruction-heavy configuration that ultimately performs best. This underscores the need for *scale-aware data curation* that validates mixture choices across multiple points on the scaling ladder, rather than at a single small-scale proxy [74, 81, 217, 218, 221, 223, 259].

Table 2: Instruction-heavy mixes are robust to moderate data repetitions.

Configuration	Core Avg
Instruction-heavy, unique	51.7
Instruction-heavy, $\sim 2\times$	50.2
Instruction-heavy, $\sim 4\times$	49.8
Instruction-heavy, $\sim 8\times$	48.6
<i>Other mixes (unique data)</i>	
Balanced	50.9
Caption-heavy	50.3
Base mix	48.8

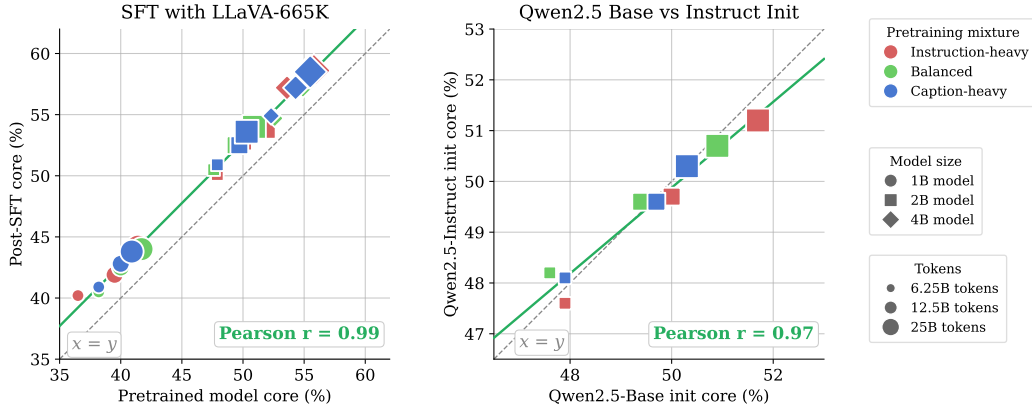


Figure 6: **Control experiments.** (Left) Pretraining performance predicts post-SFT performance with near-perfect fidelity. (Right) Data mixture rankings are preserved when switching the LM backbone from Qwen2.5-Base to Qwen2.5-Instruct, verifying robustness of our results to choice of backbone.

Repeatability of instruction-tuning data. Given our previous finding that Instruction-heavy mixes scale better, a natural concern about *scalability* arises: instruction-tuning datasets are typically orders of magnitude smaller in size than web-crawled image-caption pairs. A 70% allocation might require extreme data repetitions to fill the token budget, a known cause of performance degradation [28, 69, 100, 174, 219]. We test this effect by holding all non-instruction data sources fixed and randomly subsampling instruction-tuning data to induce up to $2\times$, $4\times$, and $8\times$ repetitions at the medium scale of our benchmark. From Tab. 2, we find that performance degrades gracefully: each doubling of repetition factor costs roughly 0.5–1.0% in performance. Notably, the Instruction-heavy mix with $2\times$ repetitions (50.2%) still matches the Caption-heavy mix with fully unique data (50.3%), and at $4\times$ repetitions it remains above the base mix (49.8% vs 48.8%). The mix ultimately degrades at $\sim 8\times$ repetitions. This result has a practical takeaway: *the benefits of a good mixture outweigh the costs of moderate data repetition*. Our results corroborate similar findings from the language domain regarding the benefits of including instruction-like data during pretraining [7, 10, 15, 70, 154, 307, 341].

4.3 Control Experiments

We now verify the generality of our findings before scaling up. Specifically, we ask: (i) Does the effectiveness of pretraining data curation hold after supervised fine-tuning (SFT)?, and (ii) Are our findings tied to the LM backbone (Qwen2.5-Base) used for initialization? We provide answers next.

Pretraining results transfer reliably post-SFT. A common concern with pretraining-only evaluations is the worry that SFT will overwrite differences induced by pretraining data choices [134]. In particular, given the findings in Sec. 4.2, it is natural to hypothesize that SFT (which also uses instruction-tuning data by definition) may interfere with or diminish the effect of using an Instruction-heavy pretraining mixture. We study this by SFT-ing all 27 pretrained checkpoints from our previous scaling grid (3 mixes \times 3 model sizes \times 3 token budgets) using two different SFT datasets: LLaVA-665K [175] and Mammoth-VL-12M [89], for a total of 54 SFT runs. We set the total SFT tokens to $0.29\times$ the pretraining tokens by estimating InternVL3’s [365] SFT-to-pretraining token ratio. Fig. 6 (left) shows the results with LLaVA-665K (refer to Sec. L for identical results with Mammoth-VL-12M). We observe that pretraining and post-SFT scores are near-perfectly correlated (Pearson $r=0.99$; Spearman $\rho=0.99$), and *the pretraining ordering is preserved across all runs*.

Our findings are robust to LM initialization. So far, we’ve used Qwen2.5-Base as the language model backbone. To verify that our findings are not specific to this particular choice, we repeat the full 2B-model sweep (3 mixes \times 3 token budgets) with Qwen2.5-Instruct-2B as the LM. This allows us to verify whether instruction-heavy mixes are better even when the LM has been “unimodally” instruction-tuned already. As shown in Fig. 6 (right), it produces nearly identical mixture rankings to Qwen2.5-Base (Pearson $r=0.97$), especially at larger token budgets (denoted by larger markers). These results provide some evidence of generality of our results—particularly, the advantage of instruction-heavy mixes after training at scale may be agnostic to the LM initialization choice.

Table 3: **DCVLM results across scales.** We compare our DCVLM-BASELINE against the best open pretraining datasets (LLaVA-ONEVISION-1.5, NEMOTRON-VL-2, FINEVISION) on the core evaluation suite, across all four scales. We also report pretrained InternVL models for reference. Benchmark categories: **Gen** = General Understanding – **Know** = Knowledge-Centric – **OCR** = OCR & Charts – **Vision** = Vision-Centric – **MTL** = Multilingual – **Text** = Text-Only Understanding.

Method	Model	Tokens	Gen	Know	OCR	Vision	MTL	Text	Core Avg
small scale									
LLaVA-OneVision-1.5	1B	6.25B	22.4	34.8	8.2	27.8	13.5	6.9	17.6
Nemotron-VL-2	1B	6.25B	20.0	39.7	7.9	33.5	16.1	20.7	22.1
FineVision	1B	6.25B	40.1	45.6	35.0	41.0	28.2	28.9	36.2
DCVLM-Baseline (ours)	1B	6.25B	40.5	43.6	33.0	39.1	25.4	34.7	36.5
medium scale									
LLaVA-OneVision-1.5	2B	25B	33.3	43.0	21.0	30.4	21.5	16.0	26.5
Nemotron-VL-2	2B	25B	48.6	54.6	19.9	41.1	36.7	28.6	37.0
FineVision	2B	25B	55.3	62.6	51.9	45.8	40.6	46.3	50.6
DCVLM-Baseline (ours)	2B	25B	62.3	60.5	45.8	47.3	44.2	47.8	51.7
large scale									
Nemotron-VL-2	4B	100B	31.5	53.8	23.6	38.6	27.5	36.4	34.7
FineVision	4B	100B	59.0	70.7	58.9	39.1	45.1	51.2	54.2
DCVLM-Baseline (ours)	4B	100B	68.4	67.6	54.1	57.2	50.9	53.8	58.9
x-large scale									
FineVision	8B	200B	63.5	72.8	57.5	49.6	48.4	55.7	58.2
DCVLM-Baseline (ours)	8B	200B	73.0	73.0	53.4	63.5	56.1	61.1	63.6
<i>open-weight, closed-data reference models</i>									
InternVL-2.5-8B	8B	~98B	68.2	70.7	52.2	52.3	45.6	63.3	60.0
InternVL-3-8B	8B	~200B	78.8	81.1	64.1	65.1	60.7	61.4	68.5
InternVL-3.5-8B	8B	~250B	77.2	80.2	63.6	63.7	59.7	63.1	68.1

5 Scaling Up Our Findings

Building on our findings, we propose DCVLM-BASELINE—a simple data recipe that forgoes filtering and instead focuses on carefully tuned data mixtures. Accordingly, we use the Instruction-heavy mix: 10% Image-Caption data, 5% Multimodal Documents, 15% Text-Only data, and 70% Instruction-Tuning data (see the full mix in Fig. 1), which was found to be optimal for medium and large scales (Sec. 4.2). For simplicity, we use this as well for the small scale—however, we reiterate the *scale-aware* nature of data curation and the fact that the “optimal” mixture for the small scale was in fact the Caption-heavy one (Fig. 5). For each data type, we fill the token budget by drawing samples from its constituent sources via simple length-proportional sampling.

We compare DCVLM-BASELINE to three open VLM pretraining datasets: ① LLaVA-OneVision-1.5-Midtraining-85M, used to pretrain the LLaVA-ONEVISION-1.5 family [12]; ② the public data released with NEMOTRON-VL-2 [58]; and ③ FINEVISION [310], the prior largest effort to unify existing sources into a single open dataset. As upper-bound reference points, we also report results from the pretrained InternVL 8B models (INTERNVL-2.5, INTERNVL-3, INTERNVL-3.5).

We train models on DCVLM-BASELINE and the FINEVISION baseline at all compute scales of our benchmark (small, medium, large, and x-large). For the other baselines (LLaVA-ONEVISION-1.5 and NEMOTRON-VL-2), we observe that performance is quite poor at smaller scales and hence did not use those datasets for training at the larger scales. We report all results in Tab. 3. These results, across all scales, confirm that **DCVLM-BASELINE outperforms open pretraining dataset for VLMs**. Specifically, compared with FINEVISION (the previous best open pretraining dataset), we observe consistent gains that *increase with scale*: DCVLM-BASELINE achieves progressive gains of +0.3pp (small), +1.1pp (medium), +4.7pp (large), and +5.4pp (x-large) on our 33-task Core evaluations. Remarkably, a 4B model trained for 100B tokens on DCVLM-BASELINE (our large scale) *outperforms* an 8B model trained for 200B tokens on FINEVISION (x-large).

52-task Extended results. These trends are further confirmed by our 52-task Extended evaluation suite (Sec. M), where a DCVLM-Baseline-trained model at the x-large scale, scores 60.5% vs. 56.6% for the corresponding x-large scale FINEVISION-trained model (an absolute improvement

of +3.9pp). In fact, with a score of 56.0, the DCVLM-Baseline-trained model at the large scale nearly achieves the same performance as the FineVision x-large model.

6 Conclusion

We introduced DCVLM, a systematic benchmark for studying data curation strategies for VLM pretraining. Through extensive experimentation across a principled scaling ladder, we established two central findings: (i) individual quality filters provide negligible benefits when the source pools are pre-filtered, which is typical for VLMs, and (ii) data mixture optimization (specifically, instruction-heavy mixtures) is the most effective curation lever, providing gains that scale reliably with model size and compute; at our largest scale (8B model, 200B tokens), DCVLM-BASELINE outperforms FINEVISION by +5.4pp on our comprehensive 33-benchmark core set. We release the full data pool (160 datasets), evaluation suite (52 benchmarks in total), model checkpoints at 4 scales, and all experimental infrastructure to serve as a reproducible testbed for future data research.

Acknowledgements

The authors would like to thank (in no particular order): Jeffrey Li, Etash Guha, Alex Fang, Pratyush Maini, Hritik Bansal, Moreno D’Inca, Songlong Xing, Olivier Henaff, Matthew Leavitt, Siddharth Joshi, Wieland Brendel, Samuel Albanie, Francesco Tonini, and Evgenia Rusak, for thoughtful feedback and comments throughout the project.

VU, AH, AG, SD and JS thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS). VU, SK and SD also thank the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program for support. AH acknowledges funding by the Federal Ministry of Research, Technology and Space (BMFTR), FKZ: 16IS24079A. SD acknowledges support by the Tübingen AI Center. AP acknowledges funding by the Federal Ministry of Research, Technology and Space (BMFTR), FKZ: 16IS24085B. VU was supported by a Google PhD Fellowship in Machine Intelligence. This work was supported by the German Research Foundation (DFG): SFB 1233, Robust Vision: Inference Principles and Neural Mechanisms, TP4, project number: 276693517 and the UKRI grant: Turing AI Fellowship EP/W002981/1. MB is a member of the Machine Learning Cluster of Excellence, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC number 2064/1 – Project number 390727645. The authors gratefully acknowledge LAION and the Gauss Centre for Supercomputing e.V. for funding this work by providing computing time on the JUWELS Booster at Jülich Supercomputing Centre (JSC). AG, MM, ER, JJ and HK receive funding from the European Union’s Horizon Europe research and innovation program under ELLIOT - Grant Agreement No 101214398. MN and JJ acknowledge co-funding by EU from Digital Europe Programme under grant no. 101195233 (openEuroLLM), co-funding from EU under Digital Europe Programme under grant no. 101198470 (LLMs4EU) and from EuroHPC Joint Undertaking programme under grant no. 101182737 (MINERVA), funding by the German Federal Ministry of Research, Technology and Space (BMFTR) under grant no. 01IS24085C (OPENHAFM), under the grant 01IS22094B (WestAI – AI Service Center West), and under the grant 16HPC117K (MINERVA). SJ is funded by the European Research Council (ERC) under the Starting Grant GraViLa (101117556). MF acknowledges travel support from ELIAS (GA no 101120237). MB acknowledge financial support by the Federal Ministry of Education and Research (BMBF), FKZ: 011524085B and Open Philanthropy Foundation funded by the Good Ventures Foundation. The authors acknowledge the CINECA award under the ISCRA initiative for the availability of high-performance computing resources and support, and the projects EU Horizon projects ELIAS (No. 101120237) and ELLIOT (No. 101214398). The authors also acknowledge the Leonardo supercomputing hours awarded by through the project EHPC-AIF-2025SC03-174. TN and SO acknowledge NSF grants 2505865, 2229876, 2229876, and 2502281. The entire team wishes to acknowledge the Gauss Centre for Supercomputing e.V. (GCS) for funding this work by providing computing time through the John von Neumann Institute for Computing (NIC) on the supercomputer JUWELS Booster and JUPITER at Jülich Supercomputing Centre (JSC), storage resources on JUST granted and operated by JSC and supported by Helmholtz Data Federation (HDF), computing time granted by the JARA and JSC on the supercomputer JURECA at JSC, computing time granted on prototype JEDI via JUREAP (JUPITER Early Access Program) grant at JSC and computing time granted via Gauss AI Competition (reformo) on JUPITER through GCS and German

Federal Ministry of Research, Technology and Space (BMFTR). LAION further acknowledges public storage grant by HuggingFace that allows us to provide convenient access to the output of the open-source research to broad community via HF repository. Further thanks go for support provided by supercomputing facilities and their teams, especially to Bjoern Hagemeyer and Mathis Bode from Juelich Supercomputer Center (JSC, Germany). SK is supported by the CS at Max Planck Doctoral Program, VIA Center and Saarland Informatics Campus. The authors acknowledge the GCP Credit Award Program by Google with award GCP444206605 for supporting the project with computational credits on GCP. MBö is supported by the Swiss National Science Foundation (project number 200021_204620).

References

- [1] A. Abbas, K. Tirumala, D. Simig, S. Ganguli, and A. S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023. Cited on page 45.
- [2] A. Abbas, E. Rusak, K. Tirumala, W. Brendel, K. Chaudhuri, and A. S. Morcos. Effective pruning of web-scale datasets based on complexity of concept clusters. *arXiv preprint arXiv:2401.04578*, 2024. Cited on page 45.
- [3] A. Abouelenin, A. Ashfaq, A. Atkinson, H. Awadalla, N. Bach, J. Bao, A. Benhaim, M. Cai, V. Chaudhary, C. Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025. Cited on page 2.
- [4] M. Acharya, K. Kafle, and C. Kanan. TallyQA: Answering complex counting questions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019. Cited on pages 53 and 54.
- [5] L. Agnolucci, L. Galteri, M. Bertini, and A. Del Bimbo. Arniqa: Learning distortion manifold for image quality assessment. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 189–198, 2024. Cited on pages 45 and 75.
- [6] J. Ainslie, J. Lee-Thorp, M. De Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4895–4901, 2023. Cited on page 48.
- [7] S. N. Akter, S. Prabhumoye, E. Nyberg, M. Patwary, M. Shoeybi, Y. Choi, and B. Catanzaro. Front-loading reasoning: The synergy between pretraining and post-training data. *arXiv preprint arXiv:2510.03264*, 2025. Cited on page 8.
- [8] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:23716–23736, 2022. Cited on pages 2 and 45.
- [9] L. B. Allal, A. Lozhkov, E. Bakouch, G. M. Blázquez, G. Penedo, L. Tunstall, A. Marafioti, H. Kydlíček, A. P. Lajarín, V. Srivastav, et al. Smollm2: When smol goes big—data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025. Cited on pages 6 and 46.
- [10] Z. Allen-Zhu and Y. Li. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*, 2023. Cited on page 8.
- [11] A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL <https://aclanthology.org/N19-1245/>. Cited on pages 53 and 54.
- [12] X. An, Y. Xie, K. Yang, W. Zhang, X. Zhao, Z. Cheng, Y. Wang, S. Xu, C. Chen, D. Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025. Cited on pages 2, 3, 9, and 45.

- [13] Z. Ankner, C. Blakeney, K. Sreenivasan, M. Marion, M. L. Leavitt, and M. Paul. Perplexed by perplexity: Perplexity-based data pruning with small reference models. *arXiv preprint arXiv:2405.20541*, 2024.
Cited on page 5.
- [14] A. Awadalla, L. Xue, O. Lo, M. Shu, H. Lee, E. Guha, M. Jordan, S. Shen, M. Awadalla, S. Savarese, et al. Mint-1t: Scaling open-source multimodal data by 10x: A multimodal dataset with one trillion tokens. *Advances in Neural Information Processing Systems (NeurIPS)*, 37: 36805–36828, 2024.
Cited on pages 4, 53, 54, 55, and 76.
- [15] C. Baek, R. P. Monti, D. Schwab, A. Abbas, R. Adiga, C. Blakeney, M. Böther, P. Burstein, A. G. Carranza, A. Deng, et al. The finetuner’s fallacy: When to pretrain with your finetuning data. *arXiv preprint arXiv:2603.16177*, 2026.
Cited on page 8.
- [16] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.
Cited on pages 2, 3, and 4.
- [17] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report, 2025. URL <https://arxiv.org/abs/2502.13923>.
Cited on pages 2, 3, and 62.
- [18] D. Berasi, M. Farina, M. Mancini, and E. Ricci. Linear model merging unlocks simple and scalable multimodal data mixture optimization. *arXiv preprint arXiv:2602.04937*, 2026.
Cited on pages 3 and 45.
- [19] A. Bevli, S. Chaybouti, Y. Dahou, H. Hacid, N. D. Huynh, P. H. L. Khac, S. Narayan, W. R. Para, and A. Singh. Falcon perception. *arXiv preprint arXiv:2603.27365*, 2026.
Cited on page 2.
- [20] L. Beyer. On the speed of ViTs and CNNs. <http://lb.eyer.be/a/vit-cnn-speed.html>, 2024.
Cited on page 68.
- [21] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.
Cited on pages 2, 45, and 46.
- [22] A. F. Biten, R. Tito, A. Mafla, L. Gomez, M. Rusinol, E. Valveny, C. Jawahar, and D. Karatzas. Scene text visual question answering. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4291–4301, 2019.
Cited on pages 53 and 54.
- [23] S. Bordt, S. Srinivas, V. Boreiko, and U. Von Luxburg. How much can we forget about data contamination? *arXiv preprint arXiv:2410.03249*, 2024.
Cited on page 46.
- [24] T. Breuel and WebDataset Contributors. WebDataset: A high-performance Python-based I/O system for large (and small) deep learning problems, with strong support for PyTorch. <https://github.com/webdataset/webdataset>, 2020.
Cited on page 85.
- [25] A. Z. Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997.
Cited on pages 4, 46, and 69.

- [26] S. Cahyawijaya, H. Lovenia, J. R. A. Moniz, T. H. Wong, M. R. Farhansyah, T. T. Maung, F. Hudi, D. Anugraha, M. R. S. Habibi, M. R. Qorib, et al. Crowdsourced, crawled, or generated? creating sea-vl, a multicultural vision-language dataset for southeast asia. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18685–18717, 2025.
Cited on pages 53 and 54.
- [27] J. Cao and J. Xiao. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *International Conference on Computational Linguistics (COLING)*, 2022.
Cited on pages 53 and 54.
- [28] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, and C. Zhang. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*, 2022.
Cited on page 8.
- [29] J. Carter. TextOCR-GPT4V: A re-captioning of TextOCR with GPT-4V, 2024. Hugging Face dataset card, <https://huggingface.co/datasets/jimmycarter/textocr-gpt4v>.
Cited on pages 53 and 54.
- [30] S. Chang, D. Palzer, J. Li, E. Fosler-Lussier, and N. Xiao. MapQA: A dataset for question answering on choropleth maps. *arXiv preprint arXiv:2211.08545*, 2022.
Cited on pages 53 and 54.
- [31] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3558–3568, 2021.
Cited on page 45.
- [32] G. H. Chen, S. Chen, R. Zhang, J. Chen, X. Wu, Z. Zhang, Z. Chen, J. Li, X. Wan, and B. Wang. ALLaVA: Harnessing GPT4V-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*, 2024.
Cited on pages 53 and 54.
- [33] J. Chen, T. Li, J. Qin, P. Lu, L. Lin, C. Chen, and X. Liang. UniGeo: Unifying geometry logical reasoning via reformulating mathematical expression. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.
Cited on pages 53 and 54.
- [34] L. Chen, J. Li, X. Dong, P. Zhang, C. He, J. Wang, F. Zhao, and D. Lin. ShareGPT4V: Improving large multi-modal models with better captions. In *European Conference on Computer Vision (ECCV)*, pages 370–387. Springer, 2024.
Cited on pages 53, 54, and 55.
- [35] L. Chen, J. Li, X. Dong, P. Zhang, Y. Zang, Z. Chen, H. Duan, J. Wang, Y. Qiao, D. Lin, et al. Are we on the right way for evaluating large vision-language models? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
Cited on page 66.
- [36] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
Cited on page 66.
- [37] M. F. Chen, T. Murray, D. Heineman, M. Jordan, H. Hajishirzi, C. Ré, L. Soldaini, and K. Lo. Olmix: A framework for data mixing throughout lm development. *arXiv preprint arXiv:2602.12237*, 2026.
Cited on pages 3, 45, and 93.
- [38] W. Chen, M. Yin, M. Ku, P. Lu, Y. Wan, X. Ma, J. Xu, X. Wang, and T. Xia. Theoremqa: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7901, 2023.
Cited on page 66.

- [39] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.
Cited on pages 53, 54, and 66.
- [40] Y. Chen, S. Qian, H. Tang, X. Lai, Z. Liu, S. Han, and J. Jia. LongLoRA: Efficient fine-tuning of long-context large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL <https://openreview.net/forum?id=6PmJoRfdaK>.
Cited on pages 53 and 54.
- [41] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.
Cited on pages 1, 4, 5, 47, 50, 52, 54, 62, and 86.
- [42] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
Cited on pages 7, 45, 47, 53, and 54.
- [43] C. K. Chng, Y. Liu, Y. Sun, C. C. Ng, C. Luo, Z. Ni, C. Fang, S. Zhang, J. Han, E. Ding, et al. ICDAR2019 robust reading challenge on arbitrary-shaped text - RRC-ArT. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.
Cited on pages 53 and 54.
- [44] J. H. Cho, A. Madotto, E. Mavroudi, T. Afouras, T. Nagarajan, M. Maaz, Y. Song, T. Ma, S. Hu, S. Jain, et al. PerceptionLM: Open-access data and models for detailed visual understanding. *arXiv preprint arXiv:2504.13180*, 2025.
Cited on pages 2 and 86.
- [45] C. Clark, J. Zhang, Z. Ma, J. S. Park, M. Salehi, R. Tripathi, S. Lee, Z. Ren, C. D. Kim, Y. Yang, et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. *arXiv preprint arXiv:2601.10611*, 2026.
Cited on pages 3, 45, and 77.
- [46] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
Cited on pages 53, 54, and 62.
- [47] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
Cited on page 66.
- [48] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, and R. Xin. Free Dolly: Introducing the world’s first truly open instruction-tuned LLM, 2023. Databricks Blog <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>.
Cited on pages 4, 53, and 54.
- [49] O. Contributors. Opencompass: A universal evaluation platform for foundation models. <https://github.com/open-compass/opencompass>, 2023.
Cited on page 62.
- [50] M. R. Costa-Jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.
Cited on page 59.
- [51] E. Cui, Y. He, Z. Ma, Z. Chen, H. Tian, W. Wang, K. Li, Y. Wang, W. Wang, X. Zhu, L. Lu, T. Lu, Y. Wang, L. Wang, Y. Qiao, and J. Dai. ShareGPT-4o: Comprehensive multimodal annotations with gpt-4o, 2024. URL <https://sharegpt4o.github.io/>.
Cited on page 3.

- [52] G. Cui, L. Yuan, N. Ding, G. Yao, B. He, W. Zhu, Y. Ni, G. Xie, R. Xie, Y. Lin, Z. Liu, and M. Sun. UltraFeedback: Boosting language models with scaled AI feedback. *International Conference on Machine Learning (ICML)*, 2024.
Cited on pages 53 and 54.
- [53] D. Dai, Y. Li, Y. Liu, M. Jia, Z. YuanHui, and G. Wang. 15M multimodal facial image-text dataset. *arXiv preprint arXiv:2407.08515*, 2024.
Cited on pages 53 and 54.
- [54] T. Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.
Cited on page 47.
- [55] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual dialog. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
Cited on pages 53 and 54.
- [56] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, et al. Large scale distributed deep networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 25, 2012.
Cited on page 50.
- [57] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 91–104, 2025.
Cited on pages 2, 3, 45, 53, 54, 55, 77, and 86.
- [58] A. S. Deshmukh, K. Chumachenko, T. Rintamaki, M. Le, T. Poon, D. M. Taheri, I. Karmanov, G. Liu, J. Seppanen, G. Chen, et al. Nvidia nemotron nano v2 vl. *arXiv preprint arXiv:2511.03929*, 2025.
Cited on pages 2, 3, 9, 45, and 86.
- [59] S. Diao, Y. Yang, Y. Fu, X. Dong, D. Su, M. Kliegl, Z. Chen, P. Belcak, Y. Suhara, H. Yin, et al. Nemotron-climb: Clustering-based iterative data mixture bootstrapping for language model pre-training. *arXiv preprint arXiv:2504.13161*, 2025.
Cited on pages 3, 45, and 77.
- [60] N. Ding, Y. Chen, B. Xu, Y. Qin, S. Hu, Z. Liu, M. Sun, and B. Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In H. Bouamor, J. Pino, and K. Bali, editors, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3029–3051, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.183. URL <https://aclanthology.org/2023.emnlp-main.183/>.
Cited on pages 53 and 54.
- [61] J. Dodge, M. Sap, A. Marasović, W. Agnew, G. Ilharco, D. Groeneveld, M. Mitchell, and M. Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 1286–1305, 2021.
Cited on page 94.
- [62] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
Cited on page 47.
- [63] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. The faiss library. *IEEE Transactions on Big Data*, 2025.
Cited on page 68.
- [64] H. Duan, J. Yang, Y. Qiao, X. Fang, L. Chen, Y. Liu, X. Dong, Y. Zang, P. Zhang, J. Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *ACM International Conference on Multimedia*, pages 11198–11201, 2024.
Cited on pages 2 and 62.

- [65] T. Evans, N. Parthasarathy, H. Merzić, and O. J. Henaff. Data curation via joint example selection further accelerates multimodal learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:141240–141260, 2024.
Cited on pages 5 and 6.
- [66] L. Fan, D. Krishnan, P. Isola, D. Katabi, and Y. Tian. Improving clip training with language rewrites. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:35544–35575, 2023.
Cited on page 45.
- [67] A. Fang, G. Ilharco, M. Wortsman, Y. Wan, V. Shankar, A. Dave, and L. Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip). In *International Conference on Machine Learning (ICML)*, pages 6216–6234. PMLR, 2022.
Cited on page 1.
- [68] A. Fang, A. M. Jose, A. Jain, L. Schmidt, A. Toshev, and V. Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023.
Cited on pages 5 and 45.
- [69] A. Fang, H. Pouransari, M. Jordan, A. Toshev, V. Shankar, L. Schmidt, and T. Gunter. Datasets, documents, and repetitions: The practicalities of unequal data quality. *arXiv preprint arXiv:2503.07879*, 2025.
Cited on page 8.
- [70] L. Feng, G. R. Ghosal, J. M. Springer, Z. Zhong, and A. Raghunathan. Early data exposure improves robustness to subsequent fine-tuning. *arXiv preprint arXiv:2605.12705*, 2026.
Cited on page 8.
- [71] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, J. Yang, X. Zheng, K. Li, X. Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.
Cited on pages 65 and 66.
- [72] X. Fu, Y. Hu, B. Li, Y. Feng, H. Wang, X. Lin, D. Roth, N. A. Smith, W.-C. Ma, and R. Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024.
Cited on page 66.
- [73] S. Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortsman, D. Ghosh, J. Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:27092–27112, 2023.
Cited on pages 1, 3, 5, 6, 45, 53, 54, 55, 59, 78, 79, and 89.
- [74] L. Gao. An empirical exploration in quality filtering of text data. *arXiv preprint arXiv:2109.00698*, 2021.
Cited on page 7.
- [75] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.
Cited on page 46.
- [76] P. Gervais, A. Fadeeva, and A. Maksai. Mathwriting: A dataset for handwritten mathematical expression recognition. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 5459–5469, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737436. URL <https://doi.org/10.1145/3711896.3737436>.
Cited on pages 53 and 54.
- [77] D. Ghosal, V. T. Y. Han, C. Y. Ken, and S. Poria. Are language models puzzle prodigies? Algorithmic puzzles unveil serious challenges in multimodal reasoning. *arXiv preprint arXiv:2403.03864*, 2024.
Cited on pages 53 and 54.

- [78] A. Ghosh, S. Dziadzio, A. Prabhu, V. Udandarao, S. Albanie, and M. Bethge. Onebench to test them all: Sample-level benchmarking over open-ended capabilities. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32445–32481, 2025.
Cited on page 2.
- [79] A. Ghosh, V. Udandarao, T. Nguyen, M. Farina, M. Cherti, J. Jitsev, S. Oh, E. Ricci, L. Schmidt, and M. Bethge. Concept-aware batch sampling improves language-image pretraining. *arXiv preprint arXiv:2511.20643*, 2025.
Cited on pages 1, 45, 78, and 79.
- [80] Glaive AI. Glaive-Code-Assistant, 2023. <https://huggingface.co/datasets/glaiveai/glaive-code-assistant>.
Cited on pages 53 and 54.
- [81] S. Goyal, P. Maini, Z. C. Lipton, A. Raghunathan, and J. Z. Kolter. Scaling laws for data filtering—data curation cannot be compute agnostic. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22702–22711, 2024.
Cited on pages 4, 7, 46, and 82.
- [82] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
Cited on pages 53 and 54.
- [83] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
Cited on page 1.
- [84] T. Gu, Z. Zhou, K. Huang, D. Liang, Y. Wang, H. Zhao, Y. Yao, X. Qiao, K. Wang, Y. Yang, et al. Mllmguard: A multi-dimensional safety evaluation suite for multimodal large language models. *Advances in Neural Information Processing Systems*, 37:7256–7295, 2024.
Cited on page 66.
- [85] T. Guan, F. Liu, X. Wu, R. Xian, Z. Li, X. Liu, X. Wang, L. Chen, F. Huang, Y. Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14375–14385, 2024.
Cited on page 66.
- [86] E. Guha, R. Marten, S. Keh, N. Raoof, G. Smyrnis, H. Bansal, M. Nezhurina, J. Mercat, T. Vu, Z. Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.
Cited on page 65.
- [87] D. Guo, F. Wu, F. Zhu, F. Leng, G. Shi, H. Chen, H. Fan, J. Wang, J. Jiang, J. Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.
Cited on page 2.
- [88] H. Guo, X. Qin, J. Liu, J. Han, J. Liu, and E. Ding. EATEN: Entity-aware attention for single shot visual text extraction. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.
Cited on pages 53 and 54.
- [89] J. Guo, T. Zheng, Y. Li, Y. Bai, B. Li, Y. Wang, K. Zhu, G. Neubig, W. Chen, and X. Yue. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13869–13920, 2025.
Cited on pages 8 and 87.
- [90] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
Cited on pages 53 and 54.

- [91] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.
Cited on page 66.
- [92] Y. Han, C. Zhang, X. Chen, X. Yang, Z. Wang, G. Yu, B. Fu, and H. Zhang. ChartLlama: A multimodal LLM for chart understanding and generation. *arXiv preprint arXiv:2311.16483*, 2023.
Cited on pages 53 and 54.
- [93] L. Hanu and Unitary team. Detoxify. Github. <https://github.com/unitaryai/detoxify>, 2020.
Cited on page 89.
- [94] C. He, Z. Jin, C. Xu, J. Qiu, B. Wang, W. Li, H. Yan, J. Wang, and D. Lin. Wanjuan: A comprehensive multimodal dataset for advancing english and chinese large models. *arXiv preprint arXiv:2308.10755*, 2023.
Cited on pages 4, 53, 54, and 55.
- [95] M. He, Y. Liu, Z. Yang, S. Zhang, C. Luo, F. Gao, Q. Zheng, Y. Wang, X. Zhang, and L. Jin. ICPR 2018 contest on robust reading for multi-type web images (MTWI). In *International Conference on Pattern Recognition (ICPR)*, 2018.
Cited on pages 53 and 54.
- [96] X. He, Y. Zhang, L. Mou, E. Xing, and P. Xie. PathVQA: 30000+ questions for medical visual question answering. *arXiv preprint arXiv:2003.10286*, 2020.
Cited on pages 53 and 54.
- [97] D. Heineman, V. Hofmann, I. Magnusson, Y. Gu, N. A. Smith, H. Hajishirzi, K. Lo, and J. Dodge. Signal and noise: A framework for reducing uncertainty in language model evaluation. *arXiv preprint arXiv:2508.13144*, 2025.
Cited on page 5.
- [98] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.
Cited on page 66.
- [99] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
Cited on page 66.
- [100] D. Hernandez, T. Brown, T. Conerly, N. DasSarma, D. Drain, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, T. Henighan, T. Hume, et al. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.
Cited on page 8.
- [101] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7514–7528, 2021.
Cited on pages 3 and 45.
- [102] R. Hong, W. Agnew, T. Kohno, and J. Morgenstern. Who’s in and who’s out? a case study of multimodal clip-filtering in datacomp. In *Proceedings of the 4th ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–17, 2024.
Cited on page 94.
- [103] W. Hong, W. Yu, X. Gu, G. Wang, G. Gan, H. Tang, J. Cheng, J. Qi, J. Ji, L. Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.
Cited on page 2.

- [104] O. Honovich, T. Scialom, O. Levy, and T. Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.806. URL <https://aclanthology.org/2023.acl-long.806/>.
Cited on pages 53 and 54.
- [105] A. Hu, H. Xu, J. Ye, M. Yan, L. Zhang, B. Zhang, J. Zhang, Q. Jin, F. Huang, and J. Zhou. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3096–3120, 2024.
Cited on pages 53 and 54.
- [106] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.
Cited on page 49.
- [107] Y. Huang, Y. Bai, Z. Zhu, J. Zhang, J. Zhang, T. Su, J. Liu, C. Lv, Y. Zhang, Y. Fu, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in neural information processing systems*, 36:62991–63010, 2023.
Cited on page 66.
- [108] Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, and C. Jawahar. ICDAR2019 competition on scanned receipt OCR and information extraction. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.
Cited on pages 53 and 54.
- [109] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6700–6709, 2019.
Cited on pages 53, 54, and 66.
- [110] B. Ionescu, H. Müller, et al. Overview of the ImageCLEF 2024: Multimedia retrieval in medical applications. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, 2024.
Cited on pages 53 and 54.
- [111] H. Jhamtani and T. Berg-Kirkpatrick. Learning to describe differences between pairs of similar images. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
Cited on pages 53 and 54.
- [112] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, pages 4904–4916. PMLR, 2021.
Cited on page 45.
- [113] Y. Jia, J. Li, X. Yue, B. Li, P. Nie, K. Zou, and W. Chen. VisualWebInstruct: Scaling up multimodal instruction data through web search. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1373–1393, Suzhou, China, Nov. 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.72. URL <https://aclanthology.org/2025.emnlp-main.72/>.
Cited on pages 53 and 54.
- [114] D. Jiang, X. He, H. Zeng, C. Wei, M. Ku, Q. Liu, and W. Chen. Mantis: Interleaved multi-image instruction tuning. *arXiv preprint arXiv:2405.01483*, 2024.
Cited on page 66.
- [115] M. Jiang, K. Z. Liu, M. Zhong, R. Schaeffer, S. Ouyang, J. Han, and S. Koyejo. Investigating data contamination for pre-training language models. *arXiv preprint arXiv:2401.06059*, 2024.
Cited on page 46.

- [116] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.
Cited on page 66.
- [117] S. Joshi, H. Yin, R. Adiga, H. Mongstad, A. Deng, A. Carranza, A. Fang, A. Abbas, A. Suri, B. Larsen, et al. 20/20 vision language models: A prescription for better vlms through data curation alone. *arXiv preprint arXiv:2605.11405*, 2026.
Cited on page 2.
- [118] S. Joshi, H. Yin, R. Adiga, R. Monti, A. Carranza, A. Fang, A. Deng, A. Abbas, B. Larsen, C. Blakeney, et al. Datbench: Discriminative, faithful, and efficient vlm evaluations. *arXiv preprint arXiv:2601.02316*, 2026.
Cited on page 2.
- [119] K. Kafle, B. Price, S. Cohen, and C. Kanan. DVQA: Understanding data visualizations via question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5648–5656, 2018.
Cited on pages 53 and 54.
- [120] S. E. Kahou, V. Michalski, A. Atkinson, Á. Kádár, A. Trischler, and Y. Bengio. FigureQA: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2017.
Cited on pages 53 and 54.
- [121] F. Kang, Y. Sun, B. Wen, S. Chen, D. Song, R. Mahmood, and R. Jia. Autoscale: Scale-aware data mixing for pre-training llms. *arXiv preprint arXiv:2407.20177*, 2024.
Cited on pages 3 and 45.
- [122] S. Kantharaj, R. T. Leong, X. Lin, A. Masry, M. Thakkar, E. Hoque, and S. Joty. Chart-to-text: A large-scale benchmark for chart summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4005–4023, 2022.
Cited on pages 53 and 54.
- [123] S. Karamcheti, S. Nair, A. Balakrishna, P. Liang, T. Kollar, and D. Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.
Cited on page 7.
- [124] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015.
Cited on page 66.
- [125] M. Kazemi, H. Alvari, A. Anand, J. Wu, X. Chen, and R. Soricut. GeomVerse: A systematic evaluation of large models for geometric reasoning. *arXiv preprint arXiv:2312.12241*, 2023.
Cited on pages 53 and 54.
- [126] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
Cited on pages 53, 54, and 66.
- [127] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision (ECCV)*, pages 235–251. Springer, 2016.
Cited on pages 53, 54, and 66.
- [128] A. Kembhavi, M. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi. Are you smarter than a sixth grader? Textbook question answering for multimodal machine comprehension. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
Cited on pages 53 and 54.

- [129] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
Cited on pages 53 and 54.
- [130] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, and S. Park. OCR-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, 2022.
Cited on pages 53 and 54.
- [131] W. Kim, S. Chun, T. Kim, D. Han, and S. Yun. Hype: Hyperbolic entailment filtering for underspecified images and texts. In *European Conference on Computer Vision (ECCV)*, pages 247–265. Springer, 2024.
Cited on page 45.
- [132] knowrohit07 and Knowledge Tech Team. Know-Saraswati-CoT: Chain-of-thought Sanskrit/Hindi reasoning dataset. <https://huggingface.co/datasets/knowrohit07/know-saraswati-cot>, 2024. Hugging Face dataset card.
Cited on pages 53 and 54.
- [133] J. Kuang, W. Hua, D. Liang, M. Yang, D. Jiang, B. Ren, and X. Bai. Visual information extraction in the wild: practical dataset and end-to-end solution. *International Conference on Document Analysis and Recognition (ICDAR)*, 2023.
Cited on pages 53 and 54.
- [134] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.
Cited on page 8.
- [135] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, et al. The Open Images dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision (IJCV)*, 128(7):1956–1981, 2020.
Cited on pages 53 and 54.
- [136] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.
Cited on page 66.
- [137] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 785–794, 2017.
Cited on page 66.
- [138] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.
Cited on page 68.
- [139] J. J. Lau, S. Gayen, A. Ben Abacha, and D. Demner-Fushman. A dataset of clinically generated visual questions and answers about radiology images. *Scientific Data*, 5(1):1–10, 2018.
Cited on pages 53 and 54.
- [140] H. Laurençon, L. Saulnier, L. Tronchon, S. Bekman, A. Singh, A. Lozhkov, T. Wang, S. Karamcheti, A. Rush, D. Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:71683–71702, 2023.
Cited on pages 45 and 76.
- [141] H. Laurençon, L. Tronchon, M. Cord, and V. Sanh. What matters when building vision-language models? *Advances in Neural Information Processing Systems (NeurIPS)*, 37: 87874–87907, 2024.
Cited on page 45.

- [142] H. Laurençon, L. Tronchon, M. Cord, and V. Sanh. What matters when building vision-language models? *Advances in Neural Information Processing Systems (NeurIPS)*, 37: 87874–87907, 2024.
Cited on page 3.
- [143] H. Laurençon, A. Marafioti, V. Sanh, and L. Tronchon. Building and better understanding vision-language models: insights and future directions, 2024. URL <https://arxiv.org/abs/2408.12637>.
Cited on pages 53 and 54.
- [144] H. Laurençon, L. Tronchon, and V. Sanh. Unlocking the conversion of web screenshots into html code with the websight dataset, 2024. URL <https://arxiv.org/abs/2403.09029>.
Cited on pages 53 and 54.
- [145] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, 2022.
Cited on page 46.
- [146] S. Lee and S. Hwang. Selective training for large vision language models via visual information gain. *arXiv preprint arXiv:2602.17186*, 2026.
Cited on page 6.
- [147] P. Lerner, O. Ferret, C. Guinaudeau, H. Le Borgne, R. Besançon, J. G. Moreno, and J. Lovon-Melgarejo. ViQuAE, a dataset for knowledge-based visual question answering about named entities. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2022.
Cited on pages 53 and 54.
- [148] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.
Cited on page 66.
- [149] B. Li, Y. Ge, Y. Chen, Y. Ge, R. Zhang, and Y. Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*, 2024.
Cited on page 66.
- [150] B. Li, Z. Lin, W. Peng, J. d. D. Nyandwi, D. Jiang, Z. Ma, S. Khanuja, R. Krishna, G. Neubig, and D. Ramanan. Naturalbench: Evaluating vision-language models on natural adversarial samples. *Advances in Neural Information Processing Systems*, 37:17044–17068, 2024.
Cited on page 66.
- [151] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.
Cited on pages 3 and 87.
- [152] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao. LLaVA-Med: Training a large language-and-vision assistant for biomedicine in one day. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.
Cited on pages 53 and 54.
- [153] H. Li, Y. Zhang, F. Koto, Y. Yang, H. Zhao, Y. Gong, N. Duan, and T. Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, 2024.
Cited on page 66.
- [154] H. Li, Y. Chen, S. Miao, Q. Dong, J. Chen, Y. Hu, J. Chen, M. Qin, Y. Wu, Y. Zhou, et al. Legalone: a family of foundation models for reliable legal reasoning. *arXiv preprint arXiv:2602.00642*, 2026.
Cited on page 8.

- [155] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, pages 19730–19742. PMLR, 2023.
Cited on page 45.
- [156] J. LI, E. Beeching, L. Tunstall, B. Lipkin, R. Soletskyi, S. Huang, K. Rasul, L. Yu, A. Q. Jiang, Z. Shen, Z. Qin, B. Dong, L. Zhou, Y. Fleureau, G. Lample, and S. Polu. NuminaMath 1.5: Second iteration of NuminaMath, 2024. Hugging Face dataset card <https://huggingface.co/datasets/AI-MO/NuminaMath-1.5>.
Cited on pages 4, 53, and 54.
- [157] J. LI, E. Beeching, L. Tunstall, et al. NuminaMath-TIR: Tool-integrated reasoning math dataset, 2024. Hugging Face dataset card <https://huggingface.co/datasets/AI-MO/NuminaMath-TIR>.
Cited on pages 53 and 54.
- [158] J. Li, A. Fang, G. Smyrnis, M. Ivgi, M. Jordan, S. Gadre, H. Bansal, E. Guha, S. Keh, K. Arora, et al. Datacomp-1m: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:14200–14282, 2024.
Cited on pages 1, 3, 5, 6, 45, 46, and 79.
- [159] J. Li, J. Chen, Y. Qu, S. Xu, Z. Lin, J. Zhu, B. Xu, W. Tan, P. Fu, J. Ju, et al. Xiaomi mimo-vl-miloco technical report. *arXiv preprint arXiv:2512.17436*, 2025.
Cited on page 2.
- [160] L. Li, Y. Wang, R. Xu, P. Wang, X. Feng, L. Kong, and Q. Liu. Multimodal ArXiv: A dataset for improving scientific comprehension of large vision-language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.
Cited on pages 53 and 54.
- [161] Q. Li, Z. Chen, W. Wang, W. Wang, S. Ye, Z. Jin, G. Chen, Y. He, Z. Gao, E. Cui, et al. Omnicorpus: A unified multimodal corpus of 10 billion-level images interleaved with text. *arXiv preprint arXiv:2406.08418*, 2024.
Cited on pages 4, 53, 54, 55, and 76.
- [162] S. Li and N. Tajbakhsh. SciGraphQA: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *arXiv preprint arXiv:2308.03349*, 2023.
Cited on pages 53 and 54.
- [163] X. Li, H. Tu, M. Hui, Z. Wang, B. Zhao, J. Xiao, S. Ren, J. Mei, Q. Liu, H. Zheng, et al. What if we recaption billions of web images with llama-3? *arXiv preprint arXiv:2406.08478*, 2024.
Cited on pages 45 and 78.
- [164] Y. Li, Y. Du, K. Zhou, J. Wang, X. Zhao, and J.-R. Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305, 2023.
Cited on pages 65 and 66.
- [165] W. Lian, G. Wang, B. Goodson, E. Pentland, A. Cook, C. Vong, and "Teknium". Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification, 2023. URL <https://huggingface.co/Open-Orca/SlimOrca>.
Cited on pages 4, 53, and 54.
- [166] H. Lin, V. Hosu, and D. Saupe. Kadid-10k: A large-scale artificially distorted iqa database. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3. IEEE, 2019.
Cited on page 75.
- [167] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.
Cited on page 66.

- [168] A. D. Lindström and S. S. Abraham. CLEVR-Math: A dataset for compositional language, visual and mathematical reasoning. In *International Workshop on Neural-Symbolic Learning and Reasoning (NeSy)*, 2022.
Cited on pages 53 and 54.
- [169] B. Liu, L.-M. Zhan, L. Xu, L. Ma, Y. Yang, and X.-M. Wu. SLAKE: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2021.
Cited on pages 53 and 54.
- [170] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang. CASIA online and offline Chinese handwriting databases. *International Conference on Document Analysis and Recognition (ICDAR)*, 2011.
Cited on pages 53 and 54.
- [171] F. Liu, G. Emerson, and N. Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics (TACL)*, 11:635–651, 2023.
Cited on pages 53, 54, and 66.
- [172] F. Liu, K. Lin, L. Li, J. Wang, Y. Yacoob, and L. Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. *International Conference on Learning Representations (ICLR)*, 2024.
Cited on pages 53 and 54.
- [173] F. Liu, X. Wang, W. Yao, J. Chen, K. Song, S. Cho, Y. Yacoob, and D. Yu. Mmc: Advancing multimodal chart understanding with large-scale instruction tuning. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1287–1310, 2024.
Cited on pages 53 and 54.
- [174] F. Liu, W. Zhou, B. Liu, P. Guo, Z. Wang, B. Zhang, Y. Zhang, Y. Yu, X. Zhou, and T. Wang. Infolaw: Information scaling laws for large language models with quality-weighted mixture data and repetition. *arXiv preprint arXiv:2605.02364*, 2026.
Cited on page 8.
- [175] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:34892–34916, 2023.
Cited on pages 1, 2, 3, 7, 8, 45, 47, and 86.
- [176] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306, 2024.
Cited on pages 53 and 54.
- [177] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee. Llawanext: Improved reasoning, ocr, and world knowledge, 2024.
Cited on pages 2, 4, and 45.
- [178] J. Liu, T. Ou, Y. Song, Y. Qu, W. Lam, C. Xiong, W. Chen, G. Neubig, and X. Yue. Harnessing webpage uis for text-rich visual understanding, 2024. URL <https://arxiv.org/abs/2410.13824>.
Cited on pages 53 and 54.
- [179] Q. Liu, X. Zheng, N. Muennighoff, G. Zeng, L. Dou, T. Pang, J. Jiang, and M. Lin. Regmix: Data mixture as regression for language model pre-training, 2024.
Cited on pages 3, 45, and 93.
- [180] X. Liu, Y. Zhu, J. Gu, Y. Lan, C. Yang, and Y. Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *European Conference on Computer Vision*, pages 386–403. Springer, 2024.
Cited on page 66.

- [181] Y. Liu, Y. Cao, Z. Gao, W. Wang, Z. Chen, W. Wang, H. Tian, L. Lu, X. Zhu, T. Lu, et al. Mminstruct: A high-quality multi-modal instruction tuning dataset with extensive diversity. *Science China Information Sciences*, 67(12):220103, 2024.
Cited on pages 53 and 54.
- [182] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision (ECCV)*, pages 216–233. Springer, 2024.
Cited on page 66.
- [183] Y. Liu, Z. Li, M. Huang, B. Yang, W. Yu, C. Li, X.-C. Yin, C.-L. Liu, L. Jin, and X. Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 2024.
Cited on page 66.
- [184] Z. Liu, T. Chu, Y. Zang, X. Dong, P. Zhang, Z. Yang, Y. Duan, D. Lin, Y. Wang, and J. Wang. MMDU: A multi-turn multi-image dialog understanding benchmark and instruction-tuning dataset for LVLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
Cited on pages 53 and 54.
- [185] longmaodata. Chinese OCR dataset. <https://huggingface.co/datasets/longmaodata/Chinese-OCR>, 2024. Hugging Face dataset card.
Cited on pages 53 and 54.
- [186] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, and A. Roberts. The flan collection: Designing data and methods for effective instruction tuning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/longpre23a.html>.
Cited on pages 4, 53, and 54.
- [187] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
Cited on page 50.
- [188] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
Cited on pages 4 and 50.
- [189] H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, J. Sun, T. Ren, Z. Li, H. Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.
Cited on page 2.
- [190] P. Lu, R. Gong, S. Jiang, L. Qiu, S. Huang, X. Liang, and S.-C. Zhu. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.
Cited on pages 53 and 54.
- [191] P. Lu, L. Qiu, J. Chen, T. Xia, Y. Zhao, W. Zhang, Z. Yu, X. Liang, and S.-C. Zhu. IconQA: A new benchmark for abstract diagram understanding and visual language reasoning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.
Cited on pages 53 and 54.
- [192] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:2507–2521, 2022.
Cited on pages 53, 54, and 66.

- [193] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.
Cited on page 66.
- [194] P. Lu, L. Qiu, K.-W. Chang, Y. N. Wu, S.-C. Zhu, T. Rajpurohit, P. Clark, and A. Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *International Conference on Learning Representations (ICLR)*, 2023.
Cited on pages 53 and 54.
- [195] Z. Luo, C. Xu, P. Zhao, Q. Sun, X. Geng, W. Hu, C. Tao, J. Ma, Q. Lin, and D. Jiang. WizardCoder: Empowering code large language models with Evol-Instruct. *International Conference on Learning Representations (ICLR)*, 2024.
Cited on pages 53 and 54.
- [196] W. Ma, H. Chen, G. Zhang, Y.-C. Chou, J. Chen, C. de Melo, and A. Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6924–6934, 2025.
Cited on page 66.
- [197] L. Madaan, A. K. Singh, R. Schaeffer, A. Poulton, S. Koyejo, P. Stenetorp, S. Narang, and D. Hupkes. Quantifying variance in evaluation benchmarks. *arXiv preprint arXiv:2406.10229*, 2024.
Cited on page 5.
- [198] I. Magar and R. Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, 2022.
Cited on page 46.
- [199] A. Mahmoud, M. Elhoushi, A. Abbas, Y. Yang, N. Ardalani, H. Leather, and A. S. Morcos. Sieve: Multimodal dataset pruning using image captioning models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22423–22432, 2024.
Cited on page 45.
- [200] P. Maini, S. Goyal, Z. C. Lipton, J. Z. Kolter, and A. Raghunathan. T-mars: Improving visual representations by circumventing text feature learning. *arXiv preprint arXiv:2307.03132*, 2023.
Cited on page 45.
- [201] C. Mao, C.-W. Xie, C. Zhong, H. Deng, J. Zhao, J. Xiao, J. Xing, J. Zhang, J. Zhou, J. Zhang, et al. Wan-image: Pushing the boundaries of generative visual intelligence. *arXiv preprint arXiv:2604.19858*, 2026.
Cited on page 3.
- [202] H. Mao, M. Cheung, and J. She. Deepart: Learning joint representations of visual arts. In *ACM International Conference on Multimedia*, pages 1183–1191, 2017.
Cited on pages 53 and 54.
- [203] A. Marafioti, O. Zohar, M. Farré, M. Noyan, E. Bakouch, P. Cuenca, C. Zakka, L. B. Allal, A. Lozhkov, N. Tazi, et al. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025.
Cited on page 2.
- [204] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
Cited on pages 53, 54, and 66.
- [205] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition (IJ DAR)*, 5:39–46, 2002.
Cited on pages 53 and 54.

- [206] A. Masry, X. L. Do, J. Q. Tan, S. Joty, and E. Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, 2022.
Cited on pages 53, 54, and 66.
- [207] A. Masry, P. Kavehzadeh, D. X. Long, E. Hoque, and S. Joty. UniChart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
Cited on pages 53 and 54.
- [208] A. Masry, M. Thakkar, A. Bajaj, A. Kartha, E. Hoque, and S. Joty. Chartgemma: Visual instruction-tuning for chart reasoning in the wild. *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 625–643, 2025.
Cited on pages 53 and 54.
- [209] M. Mathew, D. Karatzas, and C. Jawahar. Docvqa: A dataset for vqa on document images. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.
Cited on pages 53, 54, and 66.
- [210] M. Mathew, V. Bagal, R. Tito, D. Karatzas, E. Valveny, and C. Jawahar. Infographicvqa. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1697–1706, 2022.
Cited on pages 53, 54, and 66.
- [211] Maywell. KOpen-Hermes-25: Korean translation of OpenHermes-2.5, 2024. Hugging Face dataset card https://huggingface.co/datasets/maywell/ko_Ultrafeedback_binarized.
Cited on pages 53 and 54.
- [212] M. Mazumder, C. Banbury, X. Yao, B. Karlaš, W. Gaviria Rojas, S. Diamos, G. Diamos, L. He, A. Parrish, H. R. Kirk, et al. Dataperf: Benchmarks for data-centric ai development. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:5320–5347, 2023.
Cited on pages 3 and 45.
- [213] B. McKinzie, Z. Gan, J.-P. Fauconnier, S. Dodge, B. Zhang, P. Dufter, D. Shah, X. Du, F. Peng, A. Belyi, et al. Mm1: methods, analysis and insights from multimodal llm pre-training. In *European Conference on Computer Vision (ECCV)*, pages 304–323. Springer, 2024.
Cited on pages 2, 5, 7, 45, 47, 62, and 86.
- [214] N. Methani, P. Ganguly, M. M. Khapra, and P. Kumar. Plotqa: Reasoning over scientific plots. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1527–1536, 2020.
Cited on pages 53 and 54.
- [215] A. Mishra, S. Shekhar, A. K. Singh, and A. Chakraborty. OCR-VQA: Visual question answering by reading text in images. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.
Cited on pages 53, 54, and 66.
- [216] A. Mitra, H. Khanpour, C. Rosset, and A. Awadallah. Orca-math: Unlocking the potential of slms in grade school math, 2024. URL <https://arxiv.org/abs/2402.14830>.
Cited on pages 53 and 54.
- [217] D. Mizrahi, A. B. L. Larsen, J. Allardice, S. Petryk, Y. Gorokhov, J. Li, A. Fang, J. Gardner, T. Gunter, and A. Dehghan. Language models improve when pretraining data matches target tasks. *arXiv preprint arXiv:2507.12466*, 2025.
Cited on pages 4, 6, 7, 46, and 82.
- [218] C. Mohri, J. Duchi, and T. Hashimoto. A bitter lesson for data filtering. *arXiv preprint arXiv:2605.19407*, 2026.
Cited on page 7.

- [219] N. Muennighoff, A. Rush, B. Barak, T. Le Scao, N. Tazi, A. Piktus, S. Pyysalo, T. Wolf, and C. A. Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:50358–50376, 2023.
Cited on page 8.
- [220] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision (ECCV)*, pages 792–807. Springer, 2016.
Cited on pages 66 and 67.
- [221] M. Nezhurina, T. Porian, G. Pucceti, T. Kerssies, R. Beaumont, M. Cherti, and J. Jitsev. Scaling laws for robust comparison of open foundation language-vision models and datasets. *arXiv preprint arXiv:2506.04598*, 2025.
Cited on pages 4 and 7.
- [222] H. Ngo, M. Deitke, M. Bartelds, S. Pratt, J. Gardner, M. Jordan, and L. Schmidt. Olmoasr: Open models and data for training robust speech recognition models. *arXiv preprint arXiv:2508.20869*, 2025.
Cited on page 1.
- [223] H. Nguyen, V. May, H. Raj, M. Nezhurina, Y. Wang, Y. Luo, M. C. Vu, T. Nakamura, K. Tsui, V. K. Nguyen, et al. Mixturevitae: Open web-scale pretraining dataset with high quality instruction and reasoning data built from permissive-first text sources. *arXiv preprint arXiv:2509.25531*, 2025.
Cited on page 7.
- [224] T. Nguyen, G. Ilharco, M. Wortsman, S. Oh, and L. Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:21455–21469, 2022.
Cited on page 1.
- [225] T. Nguyen, S. Y. Gadre, G. Ilharco, S. Oh, and L. Schmidt. Improving multimodal datasets with image captioning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:22047–22069, 2023.
Cited on page 45.
- [226] T. Nguyen, M. Wallingford, S. Santy, W.-C. Ma, S. Oh, L. Schmidt, P. W. Koh, and R. Krishna. Multilingual diversity improves vision-language representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:91430–91459, 2024.
Cited on pages 45 and 59.
- [227] T. OLMO, P. Walsh, L. Soldaini, D. Groeneveld, K. Lo, S. Arora, A. Bhagia, Y. Gu, S. Huang, M. Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.
Cited on pages 3, 5, 45, and 77.
- [228] T. Olmo, A. Ettinger, A. Bertsch, B. Kuehl, D. Graham, D. Heineman, D. Groeneveld, F. Brahman, F. Timbers, H. Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.
Cited on page 5.
- [229] OpenAI. Chat markup language (ChatML). <https://github.com/openai/openai-python/blob/main/chatml.md>, 2022. Accessed: 29 April 2026.
Cited on page 75.
- [230] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
Cited on pages 45 and 46.
- [231] S. Parashar, Z. Lin, T. Liu, X. Dong, Y. Li, D. Ramanan, J. Caverlee, and S. Kong. The neglected tails in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12988–12997, 2024.
Cited on page 60.

- [232] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023.
Cited on page 1.
- [233] G. Penedo, H. Kydlíček, A. Lozhkov, M. Mitchell, C. Raffel, L. Von Werra, T. Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:30811–30849, 2024.
Cited on pages 1, 3, 5, 45, 46, and 77.
- [234] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.
Cited on pages 53, 54, and 55.
- [235] E. Pizzi, S. D. Roy, S. N. Ravindra, P. Goyal, and M. Douze. A self-supervised descriptor for image copy detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14532–14542, 2022.
Cited on pages 4 and 68.
- [236] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, et al. Color image database tid2013: Peculiarities and preliminary results. In *European workshop on visual information processing (EUVIP)*, pages 106–111. IEEE, 2013.
Cited on page 75.
- [237] A. Pouget, L. Beyer, E. Bugliarello, X. Wang, A. P. Steiner, X. Zhai, and I. Alabdulmohsin. No filter: Cultural and socioeconomic diversity in contrastive vision-language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:106474–106496, 2024.
Cited on page 59.
- [238] S. Pramanick, R. Chellappa, and S. Venugopalan. SPIQA: A dataset for multimodal question answering on scientific papers. *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.
Cited on pages 53 and 54.
- [239] R. Qiao, Q. Tan, G. Dong, M. MinhuiWu, C. Sun, X. Song, J. Wang, Z. Gongque, S. Lei, Y. Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20023–20070, 2025.
Cited on page 66.
- [240] Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
Cited on pages 3, 4, 47, and 48.
- [241] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PmLR, 2021.
Cited on pages 5 and 77.
- [242] N. Rajani, L. Tunstall, E. Beeching, N. Lambert, A. M. Rush, and T. Wolf. No Robots, 2023. https://huggingface.co/datasets/HuggingFaceH4/no_robots.
Cited on pages 53 and 54.
- [243] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pages 1–16. IEEE, 2020.
Cited on page 50.

- [244] RayBernard. leetcode. <https://huggingface.co/datasets/RayBernard/leetcode>, 2024. Hugging Face dataset card.
Cited on pages 53 and 54.
- [245] J. Rodriguez, X. Jian, S. S. Panigrahi, T. Zhang, A. Feizi, A. Puri, A. Kalkunte Suresh, F. Savard, A. Masry, S. Nayak, R. Awal, M. Massoud, A. Abaskohi, Z. Li, S. Wang, P.-A. Noël, M. L. Richter, S. Vadicchino, S. Agarwal, S. Biswas, S. Shanian, Y. Zhang, N. Bolger, K. MacDonald, S. Fauvel, S. Tejaswi, S. Sunkara, J. Monteiro, K. D. Dvijotham, T. Scholak, N. Chapados, S. Kharaghani, S. Hughes, M. T. Özsu, S. Reddy, M. Pedersoli, Y. Bengio, C. Pal, I. Laradji, S. Gella, P. Taslakian, D. Vazquez, and S. Rajeswar. BigDocs: An open dataset for training multimodal models on document and code tasks. In *International Conference on Learning Representations (ICLR)*, 2025.
Cited on pages 53 and 54.
- [246] K. Roth, V. Udandarao, S. Dziadzio, A. Prabhu, M. Cherti, O. Vinyals, O. Hénaff, S. Albanie, M. Bethge, and Z. Akata. A practitioner’s guide to continual multimodal pretraining. *arXiv preprint arXiv:2408.14471*, 2024.
Cited on page 50.
- [247] O. Sainz, I. García-Ferrero, A. Jacovi, J. A. Campos, Y. Elazar, E. Agirre, Y. Goldberg, W.-L. Chen, J. Chim, L. Choshen, et al. Data contamination report from the 2024 conda shared task. In *Proceedings of the 1st Workshop on Data Contamination (CONDA)*, pages 41–56, 2024.
Cited on page 46.
- [248] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.
Cited on page 66.
- [249] R. Saxena, P. Minervini, and F. Keller. PosterSum: A multimodal benchmark for scientific poster summarization. In K. Inui, S. Sakti, H. Wang, D. F. Wong, P. Bhattacharyya, B. Banerjee, A. Ekbal, T. Chakraborty, and D. P. Singh, editors, *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, Dec. 2025.
Cited on pages 53 and 54.
- [250] R. Schaeffer, J. Kazdan, B. Abbasi, K. Z. Liu, B. Miranda, A. Ahmed, F. Berez, A. Puri, S. Biderman, N. Mireshghallah, et al. Quantifying the effect of test set contamination on generative evaluations. *arXiv preprint arXiv:2601.04301*, 2026.
Cited on page 46.
- [251] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:25278–25294, 2022.
Cited on pages 1, 3, 6, 53, 54, 55, and 59.
- [252] D. Schwenk, A. Khandelwal, C. Clark, K. Marino, and R. Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision (ECCV)*, 2022.
Cited on pages 53 and 54.
- [253] S. Shah, A. Mishra, N. Yadati, and P. P. Talukdar. KVQA: Knowledge-aware visual question answering. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.
Cited on pages 53 and 54.
- [254] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun. Objects365: A large-scale, high-quality dataset for object detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
Cited on pages 53 and 54.
- [255] H. Shapourian, K. Hejazi, O. M. Sule, and B. Millidge. Zaya1-v1-8b technical report. *arXiv preprint arXiv:2605.08560*, 2026.
Cited on page 3.

- [256] N. Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.
Cited on page 48.
- [257] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016.
Cited on page 4.
- [258] R. Shinoda, K. Saito, S. Tanaka, T. Hirasawa, and Y. Ushiku. SBS Figures: Pre-training figure QA from stage-by-stage synthesized images. *arXiv preprint arXiv:2412.17606*, 2024.
Cited on pages 53 and 54.
- [259] M. Shukor, L. Bethune, D. Busbridge, D. Grangier, E. Fini, A. El-Nouby, and P. Ablin. Scaling laws for optimal data mixtures. *arXiv preprint arXiv:2507.09404*, 2025.
Cited on page 7.
- [260] O. Sidorov, R. Hu, M. Rohrbach, and A. Singh. TextCaps: A dataset for image captioning with reading comprehension. In *European Conference on Computer Vision (ECCV)*, pages 742–758. Springer, 2020.
Cited on pages 53 and 54.
- [261] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards vqa models that can read. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8317–8326, 2019.
Cited on pages 53, 54, and 66.
- [262] A. Singh, G. Pang, M. Toh, J. Huang, W. Galuba, and T. Hassner. TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
Cited on pages 53 and 54.
- [263] e. a. Singh. Persian synthetic OCR dataset (*ParSynth-OCR-200K*), 2021. Hugging Face dataset card.
Cited on pages 53 and 54.
- [264] B. Sorscher, R. Geirhos, S. Shekhar, S. Ganguli, and A. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:19523–19536, 2022.
Cited on page 45.
- [265] A. Steiner, A. S. Pinto, M. Tschannen, D. Keysers, X. Wang, Y. Bitton, A. Gritsenko, M. Minderer, A. Sherbondy, S. Long, et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.
Cited on page 2.
- [266] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
Cited on pages 53 and 54.
- [267] D. Su, K. Kong, Y. Lin, J. Jennings, B. Norick, M. Kliegl, M. Patwary, M. Shoeybi, and B. Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2459–2475, 2025.
Cited on pages 1, 3, 5, 6, and 45.
- [268] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
Cited on page 48.
- [269] H.-L. Sun, D.-W. Zhou, Y. Li, S. Lu, C. Yi, Q.-G. Chen, Z. Xu, W. Luo, K. Zhang, D.-C. Zhan, et al. Parrot: Multilingual visual instruction tuning. *arXiv preprint arXiv:2406.02539*, 2024.
Cited on page 66.

- [270] T. Sun, X. Zhang, Z. He, P. Li, Q. Cheng, X. Liu, H. Yan, Y. Shao, Q. Tang, S. Zhang, et al. MOSS: An open conversational large language model. *Machine Intelligence Research*, 2024. Cited on pages 53 and 54.
- [271] Y. Sun, Z. Ni, C.-K. Chng, Y. Liu, C. Luo, C. C. Ng, J. Han, E. Ding, J. Liu, D. Karatzas, et al. ICDAR2019 competition on large-scale street view text with partial labeling - RRC-LSVT. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019. Cited on pages 53 and 54.
- [272] R. Tanaka, K. Nishida, and S. Yoshida. VisualMRC: Machine reading comprehension on document images. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. Cited on pages 53 and 54.
- [273] B. J. Tang, A. Boggust, and A. Satyanarayan. VisText: A benchmark for semantically rich chart captioning. *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. Cited on pages 53 and 54.
- [274] J. Tang, Q. Liu, Y. Ye, J. Lu, S. Wei, A.-L. Wang, C. Lin, H. Feng, Z. Zhao, Y. Wang, et al. Mtvqa: Benchmarking multilingual text-centric visual question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7748–7763, 2025. Cited on page 66.
- [275] C. Team, Z. Yue, Z. Lin, Y. Song, W. Wang, S. Ren, S. Gu, S. Li, P. Li, L. Zhao, L. Li, K. Bao, H. Tian, H. Zhang, G. Wang, D. Zhu, Cici, C. He, B. Ye, B. Shen, Z. Zhang, Z. Jiang, Z. Zheng, Z. Song, Z. Luo, Y. Yu, Y. Wang, Y. Tian, Y. Tu, Y. Yan, Y. Huang, X. Wang, X. Xu, X. Song, X. Zhang, X. Yong, X. Zhang, X. Deng, W. Yang, W. Ma, W. Lv, W. Zhuang, W. Liu, S. Deng, S. Liu, S. Chen, S. Yu, S. Liu, S. Wang, R. Ma, Q. Wang, P. Wang, N. Chen, M. Zhu, K. Zhou, K. Zhou, K. Fang, J. Shi, J. Dong, J. Xiao, J. Xu, H. Liu, H. Xu, H. Qu, H. Zhao, H. Lv, G. Wang, D. Zhang, D. Zhang, C. Ma, C. Liu, C. Cai, and B. Xia. MIMO-VL technical report, 2025. URL <https://arxiv.org/abs/2506.03569>. Cited on pages 2 and 86.
- [276] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, L. Rouillard, T. Mesnard, G. Cideron, J. Bastien Grill, S. Ramos, E. Yvinec, M. Casbon, E. Pot, I. Penchev, G. Liu, F. Visin, K. Kenealy, L. Beyer, X. Zhai, A. Tsitsulin, R. Busa-Fekete, A. Feng, N. Sachdeva, B. Coleman, Y. Gao, B. Mustafa, I. Barr, E. Parisotto, D. Tian, M. Eyal, C. Cherry, J.-T. Peter, D. Sinopalnikov, S. Bhupatiraju, R. Agarwal, M. Kazemi, D. Malkin, R. Kumar, D. Vilar, I. Brusilovsky, J. Luo, A. Steiner, A. Friesen, A. Sharma, A. Sharma, A. M. Gilady, A. Goedeckemeyer, A. Saade, A. Feng, A. Kolesnikov, A. Bendebury, A. Abdagic, A. Vadi, A. György, A. S. Pinto, A. Das, A. Bapna, A. Miech, A. Yang, A. Paterson, A. Shenoy, A. Chakrabarti, B. Piot, B. Wu, B. Shahriari, B. Petrini, C. Chen, C. L. Lan, C. A. Choquette-Choo, C. Carey, C. Brick, D. Deutsch, D. Eisenbud, D. Cattle, D. Cheng, D. Paparas, D. S. Sreepathihalli, D. Reid, D. Tran, D. Zelle, E. Noland, E. Huizenga, E. Kharitonov, F. Liu, G. Amirkhanyan, G. Cameron, H. Hashemi, H. Klimczak-Plucińska, H. Singh, H. Mehta, H. T. Lehri, H. Hazimeh, I. Ballantyne, I. Szpektor, I. Nardini, J. Pouget-Abadie, J. Chan, J. Stanton, J. Wieting, J. Lai, J. Orbay, J. Fernandez, J. Newlan, J. yeong Ji, J. Singh, K. Black, K. Yu, K. Hui, K. Vodrahalli, K. Greff, L. Qiu, M. Valentine, M. Coelho, M. Ritter, M. Hoffman, M. Watson, M. Chaturvedi, M. Moynihan, M. Ma, N. Babar, N. Noy, N. Byrd, N. Roy, N. Momchev, N. Chauhan, N. Sachdeva, O. Bunyan, P. Botarda, P. Caron, P. K. Rubenstein, P. Culliton, P. Schmid, P. G. Sessa, P. Xu, P. Stanczyk, P. Tafti, R. Shivanna, R. Wu, R. Pan, R. Rokni, R. Willoughby, R. Vallu, R. Mullins, S. Jerome, S. Smoot, S. Girgin, S. Iqbal, S. Reddy, S. Sheth, S. Pöder, S. Bhatnagar, S. R. Panyam, S. Eiger, S. Zhang, T. Liu, T. Yacovone, T. Liechty, U. Kalra, U. Evcı, V. Misra, V. Roseberry, V. Feinberg, V. Kolesnikov, W. Han, W. Kwon, X. Chen, Y. Chow, Y. Zhu, Z. Wei, Z. Egyed, V. Cotruta, M. Giang, P. Kirk, A. Rao, K. Black, N. Babar, J. Lo, E. Moreira, L. G. Martins, O. Sanseviero, L. Gonzalez, Z. Gleicher, T. Warkentin, V. Mirrokni, E. Senter, E. Collins, J. Barral, Z. Ghahramani, R. Hadsell, Y. Matias, D. Sculley, S. Petrov, N. Fiedel, N. Shazeer, O. Vinyals, J. Dean, D. Hassabis, K. Kavukcuoglu, C. Farabet, E. Buchatskaya, J.-B. Alayrac, R. Anil, Dmitry, Lepikhin, S. Borgeaud, O. Bachem, A. Joulin, A. Andreev, C. Hardin, R. Dadashi, and L. Hussenot. Gemma 3 technical report, 2025. URL

<https://arxiv.org/abs/2503.19786>.

Cited on page 2.

- [277] K. Team, A. Du, B. Yin, B. Xing, B. Qu, B. Wang, C. Chen, C. Zhang, C. Du, C. Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.
Cited on page 2.
- [278] T. M. A. Team. Mai-thinking-1: Building a hill-climbing machine. Technical report, Microsoft AI, 2026. URL <https://microsoft.ai/pdf/mai-thinking-1.pdf>.
Cited on pages 4 and 82.
- [279] S. Tong, E. Brown, P. Wu, S. Woo, M. Middepogu, S. C. Akula, J. Yang, S. Yang, A. Iyer, X. Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:87310–87356, 2024.
Cited on pages 2, 5, 7, 45, 53, 54, 62, and 66.
- [280] S. Tong, Z. Liu, Y. Zhai, Y. Ma, Y. LeCun, and S. Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9568–9578, 2024.
Cited on page 66.
- [281] T. H. Trinh and Q. V. Le. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*, 2018.
Cited on page 46.
- [282] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.
Cited on page 5.
- [283] Y. Tuo, W. Xiang, J.-Y. He, Y. Geng, and X. Xie. Anytext: Multilingual visual text generation and editing. In *International Conference on Learning Representations (ICLR)*, 2024. URL <https://openreview.net/forum?id=ezBH9WE9s2>.
Cited on pages 53 and 54.
- [284] V. Udandarao, A. Prabhu, A. Ghosh, Y. Sharma, P. H. Torr, A. Bibi, S. Albanie, and M. Bethge. No "zero-shot" without exponential data: Pretraining concept frequency determines multimodal model performance. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:61735–61792, 2024.
Cited on page 60.
- [285] V. Udandarao, Z. Lu, X. Chang, Y. Wang, V. Z. Yao, A. M. Jose, F. Faghri, J. Gardner, and C.-C. Chiu. Data-centric lessons to improve speech-language pretraining. *arXiv preprint arXiv:2510.20860*, 2025.
Cited on page 1.
- [286] V. Udandarao, N. Parthasarathy, M. F. Naeem, T. Evans, S. Albanie, F. Tombari, Y. Xian, A. Tonioni, and O. J. Hénaff. Active data curation effectively distills large-scale multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14422–14437, 2025.
Cited on pages 5 and 6.
- [287] D. Ustalov, N. Pavlichenko, S. Koshelev, D. Likhobaba, and A. Smirnova. Toloka visual question answering benchmark. *arXiv preprint arXiv:2309.16511*, 2023.
Cited on pages 53, 54, and 66.
- [288] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The iNaturalist species classification and detection dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
Cited on pages 53 and 54.

- [289] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie. COCO-Text: Dataset and benchmark for text detection and recognition in natural images. In *arXiv preprint arXiv:1601.07140*, 2016.
Cited on pages 53 and 54.
- [290] H. V. Vo, V. Khalidov, T. Darceet, T. Moutakanni, N. Smetanin, M. Szafraniec, H. Touvron, C. Couprie, M. Oquab, A. Joulin, et al. Automatic data curation for self-supervised learning: A clustering-based approach. *arXiv preprint arXiv:2405.15613*, 2024.
Cited on page 45.
- [291] B. Wang, G. Li, X. Zhou, Z. Chen, T. Grossman, and Y. Li. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 498–510, 2021.
Cited on pages 53 and 54.
- [292] F. Wang, X. Fu, J. Y. Huang, Z. Li, Q. Liu, X. Liu, M. D. Ma, N. Xu, W. Zhou, K. Zhang, et al. Muirbench: A comprehensive benchmark for robust multi-image understanding. *arXiv preprint arXiv:2406.09411*, 2024.
Cited on page 66.
- [293] J. Wang, L. Meng, Z. Weng, B. He, Z. Wu, and Y.-G. Jiang. To see is to believe: Prompting gpt-4v for better visual instruction tuning, 2023. URL <https://arxiv.org/abs/2311.07574>.
Cited on pages 53 and 54.
- [294] J. Wang, Y. Wang, G. Xu, J. Zhang, Y. Gu, H. Jia, J. Wang, H. Xu, M. Yan, J. Zhang, et al. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023.
Cited on page 66.
- [295] J. Wang, P. Zhang, T. Chu, Y. Cao, Y. Zhou, T. Wu, B. Wang, C. He, and D. Lin. V3Det: Vast vocabulary visual detection dataset. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
Cited on pages 53 and 54.
- [296] K. Wang, J. Pan, W. Shi, Z. Lu, H. Ren, A. Zhou, M. Zhan, and H. Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.
Cited on page 66.
- [297] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.
Cited on pages 5, 62, 78, and 79.
- [298] W. Wang, Y. Ren, H. Luo, T. Li, C. Yan, Z. Chen, W. Wang, Q. Li, L. Lu, X. Zhu, et al. The all-seeing project V2: Towards general relation comprehension of the open world. In *European Conference on Computer Vision (ECCV)*, 2024.
Cited on pages 53, 54, and 66.
- [299] W. Wang, M. Shi, Q. Li, W. Wang, Z. Huang, L. Xing, Z. Chen, H. Li, X. Zhu, Z. Cao, et al. The all-seeing project: Towards panoptic visual recognition and understanding of the open world. In *International Conference on Learning Representations (ICLR)*, 2024.
Cited on pages 53 and 54.
- [300] W. Wang, Z. Gao, L. Gu, H. Pu, L. Cui, X. Wei, Z. Liu, L. Jing, S. Ye, J. Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.
Cited on pages 1, 2, 3, and 45.
- [301] W. Wang, R. Lin, S. Li, C. Lockard, R. Sarkhel, S. Lokegaonkar, J. Shang, X. Yan, N. Zalmout, and X. Li. Train a unified multimodal data quality classifier with synthetic data. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1972–1986, 2025.
Cited on pages 3, 5, and 45.

- [302] X. Wang, Y. Liu, C. Shen, C. C. Ng, C. Luo, L. Jin, C. S. Chan, A. van den Hengel, and L. Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
Cited on pages 53 and 54.
- [303] Y. Wang, Y. Chen, W. Yan, A. Fang, W. Zhou, K. Jamieson, and S. S. Du. Cliploss and norm-based data selection methods for multimodal contrastive learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:15028–15069, 2024.
Cited on page 45.
- [304] Y. Wang, Y. Chen, W. Yan, K. Jamieson, and S. S. Du. Variance alignment score: A simple but tough-to-beat data selection method for multimodal contrastive learning. *arXiv preprint arXiv:2402.02055*, 2024.
Cited on page 45.
- [305] Y. Wang, K. He, D. Fu, Z. GongQue, H. Xu, Y. Chen, Z. Wang, Y. Fu, G. Dong, M. Diao, J. Wang, M. Zhang, X. Cai, and W. Xu. How do your code LLMs perform? empowering code instruction tuning with really good data. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14027–14043, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.777. URL <https://aclanthology.org/2024.emnlp-main.777/>.
Cited on pages 4, 53, and 54.
- [306] Z. Wang, M. Xia, L. He, H. Chen, Y. Liu, R. Zhu, K. Liang, X. Wu, H. Liu, S. Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems (NeurIPS)*, 37, 2024.
Cited on page 66.
- [307] R. Washbourne, R. Iyer, T. Figliolia, H. Zheng, R. Lorig-Roach, S. Yang, P. Yuvraj, Q. Anthony, Y. Tokpanov, X. Yang, et al. Zaya1-8b technical report. *arXiv preprint arXiv:2605.05365*, 2026.
Cited on page 8.
- [308] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations (ICLR)*, 2022.
Cited on pages 53 and 54.
- [309] F. Wandler. RenderedText: A synthetic dataset of rendered text images, 2023. Hugging Face dataset <https://huggingface.co/datasets/wandlerc/RenderedText>.
Cited on pages 53 and 54.
- [310] L. Wiedmann, O. Zohar, A. Mahla, X. Wang, R. Li, T. Frere, L. von Werra, A. R. Gosthipaty, and A. Marafioti. Finevision: Open data is all you need. *arXiv preprint arXiv:2510.17269*, 2025.
Cited on pages 2, 3, 4, 9, 45, 46, 52, and 68.
- [311] M. Wortsman, P. J. Liu, L. Xiao, K. Everett, A. Alemi, B. Adlam, J. D. Co-Reyes, I. Gur, A. Kumar, R. Novak, et al. Small-scale proxies for large-scale transformer training instabilities. *arXiv preprint arXiv:2309.14322*, 2023.
Cited on page 50.
- [312] C. Wu, J. Li, J. Zhou, J. Lin, K. Gao, K. Yan, S.-m. Yin, S. Bai, X. Xu, Y. Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.
Cited on page 3.
- [313] Z. Wu, X. Chen, Z. Pan, X. Liu, W. Liu, D. Dai, H. Gao, Y. Ma, C. Wu, B. Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.
Cited on page 2.

- [314] xAI. RealWorldQA. <https://huggingface.co/datasets/xai-org/RealworldQA>, 2024. Dataset hosted on Hugging Face.
Cited on page 66.
- [315] R. Xia, B. Zhang, H. Ye, X. Yan, Q. Liu, H. Zhou, Z. Chen, M. Dou, B. Shi, J. Yan, and Y. Qiao. StructChart: Perception, structuring, reasoning for visual chart understanding. *arXiv preprint arXiv:2309.11268*, 2023.
Cited on pages 53 and 54.
- [316] R. Xia, H. Ye, X. Yan, Q. Liu, H. Zhou, Z. Chen, B. Shi, J. Yan, and B. Zhang. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *IEEE Transactions on Image Processing*, 2025.
Cited on pages 53 and 54.
- [317] S. M. Xie, H. Pham, X. Dong, N. Du, H. Liu, Y. Lu, P. S. Liang, Q. V. Le, T. Ma, and A. W. Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:69798–69818, 2023.
Cited on pages 3, 45, and 93.
- [318] C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao, Q. Lin, and D. Jiang. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations (ICLR)*, 2024.
Cited on pages 53 and 54.
- [319] H. Xu, S. Xie, X. E. Tan, P.-Y. Huang, R. Howes, V. Sharma, S.-W. Li, G. Ghosh, L. Zettlemoyer, and C. Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023.
Cited on page 45.
- [320] J. Xu, Z. Guo, H. Hu, Y. Chu, X. Wang, J. He, Y. Wang, X. Shi, T. He, X. Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.
Cited on page 2.
- [321] L. Xu, H. Hu, X. Zhang, L. Li, C. Cao, Y. Li, Y. Xu, K. Sun, D. Yu, C. Yu, et al. Clue: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th international conference on computational linguistics*, pages 4762–4772, 2020.
Cited on page 66.
- [322] Z. Xu, F. Jiang, L. Niu, Y. Deng, R. Poovendran, Y. Choi, and B. Y. Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing, 2024. URL <https://arxiv.org/abs/2406.08464>.
Cited on pages 53 and 54.
- [323] A. Yan, Z. Yang, J. Wu, W. Zhu, J. Yang, L. Li, K. Lin, J. Wang, J. McAuley, J. Gao, and L. Wang. List items one by one: A new data source and learning paradigm for multimodal llms, 2025. URL <https://arxiv.org/abs/2404.16375>.
Cited on pages 53 and 54.
- [324] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Yang, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, X. Liu, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Guo, and Z. Fan. Qwen2 technical report, 2024. URL <https://arxiv.org/abs/2407.10671>.
Cited on pages 47 and 48.
- [325] B. Yang, B. Wen, B. Ding, C. Liu, C. Chu, C. Song, C. Rao, C. Yi, D. Li, D. Zang, et al. Kwai keye-vl 1.5 technical report. *arXiv preprint arXiv:2509.01563*, 2025.
Cited on page 2.
- [326] J. Yang. Firefly: Chinese conversational large language models, 2023. <https://github.com/yangjianxin1/Firefly>.
Cited on pages 53 and 54.

- [327] J. Yang. Longqlora: Efficient and effective method to extend context length of large language models, 2023. URL <https://arxiv.org/abs/2311.04879>.
Cited on pages 53 and 54.
- [328] Y. Yang, A. Patel, M. Deitke, T. Gupta, L. Weihs, A. Head, M. Yatskar, C. Callison-Burch, R. Krishna, A. Kembhavi, et al. Scaling text-rich image understanding via code-guided synthetic multimodal data generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17486–17505, 2025.
Cited on pages 53 and 54.
- [329] J. Ye, A. Hu, H. Xu, Q. Ye, M. Yan, G. Xu, C. Li, J. Tian, Q. Qian, J. Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2841–2858, 2023.
Cited on page 45.
- [330] J. Ye, P. Liu, T. Sun, J. Zhan, Y. Zhou, and X. Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*, 2024.
Cited on pages 3 and 45.
- [331] K. Ying, F. Meng, J. Wang, Z. Li, H. Lin, Y. Yang, H. Zhang, W. Zhang, Y. Lin, S. Liu, J. Lei, Q. Lu, R. Chen, P. Xu, R. Zhang, H. Zhang, P. Gao, Y. Wang, Y. Qiao, P. Luo, K. Zhang, and W. Shao. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi. *International Conference on Machine Learning (ICML)*, 2024.
Cited on page 66.
- [332] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *European Conference on Computer Vision (ECCV)*, 2016.
Cited on pages 53, 54, 66, and 67.
- [333] L. Yu, W. Jiang, H. Shi, J. YU, Z. Liu, Y. Zhang, J. Kwok, Z. Li, A. Weller, and W. Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL <https://openreview.net/forum?id=N8N0hgNDRt>.
Cited on pages 53 and 54.
- [334] Q. Yu, Q. Sun, X. Zhang, Y. Cui, F. Zhang, Y. Cao, X. Wang, and J. Liu. Capsfusion: Rethinking image-text data at scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14022–14032, 2024.
Cited on pages 45 and 78.
- [335] T. Yu, Z. Wang, C. Wang, F. Huang, W. Ma, Z. He, T. Cai, W. Chen, Y. Huang, R. Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11704–11715, 2026.
Cited on page 3.
- [336] T.-L. Yuan, Z. Zhu, K. Xu, C.-J. Li, T.-J. Mu, and S.-M. Hu. A large Chinese text dataset in the wild. *Journal of Computer Science and Technology*, 34:509–521, 2019.
Cited on pages 53 and 54.
- [337] Y. Yuan, X. Liu, W. Dikubab, H. Liu, Z. Ji, Z. Wu, and X. Bai. Syntax-aware network for handwritten mathematical expression recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
Cited on pages 53 and 54.
- [338] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
Cited on page 66.

- [339] yuyijiong. Long-Instruction-with-Paraphrasing. <https://huggingface.co/datasets/yuyijiong/Long-Instruction-with-Paraphrasing>, 2024. Hugging Face dataset card. Cited on pages 53 and 54.
- [340] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800, 2019. Cited on page 66.
- [341] A. Zeng, X. Lv, Q. Zheng, Z. Hou, B. Chen, C. Xie, C. Wang, D. Yin, H. Zeng, J. Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025. Cited on page 8.
- [342] W. Zeng, D. Kurniawan, R. Mullins, Y. Liu, T. Saha, D. Ike-Njoku, J. Gu, Y. Song, C. Xu, J. Zhou, et al. Shieldgemma 2: Robust and tractable image content moderation. *arXiv preprint arXiv:2504.01081*, 2025. Cited on page 89.
- [343] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer. Scaling vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, 2022. Cited on page 46.
- [344] B. Zhang and R. Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. Cited on page 48.
- [345] B. Zhang, L. Ke, R. Yang, Q. Gao, T. Qu, R. Chen, D. Yu, et al. Penguin-vl: Exploring the efficiency limits of vlm with llm-based vision encoders. *arXiv preprint arXiv:2603.06569*, 2026. Cited on page 3.
- [346] B.-W. Zhang, Y. Yan, L. Li, and G. Liu. Infinity $\langle \text{math} \rangle$ a scalable instruction tuning dataset in programmatic mathematical reasoning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, page 5405–5409. ACM, Oct. 2024. doi: 10.1145/3627673.3679122. URL <http://dx.doi.org/10.1145/3627673.3679122>. Cited on pages 53 and 54.
- [347] H. Zhang, M. Gao, Z. Gan, P. Dufter, N. Wenzel, F. Huang, D. Shah, X. Du, B. Zhang, Y. Li, et al. Mm1. 5: Methods, analysis & insights from multimodal llm fine-tuning. *arXiv preprint arXiv:2409.20566*, 2024. Cited on pages 7 and 45.
- [348] J. Zhang, L. Xue, L. Song, J. Wang, W. Huang, M. Shu, A. Yan, Z. Ma, J. C. Niebles, S. Savarese, C. Xiong, Z. Chen, R. Krishna, and R. Xu. Provision: Programmatically scaling vision-centric instruction data for multimodal language models, 2024. URL <https://arxiv.org/abs/2412.07012>. Cited on pages 53 and 54.
- [349] J. Zhang, Y. Bai, X. Lv, W. Gu, D. Liu, M. Zou, S. Cao, L. Hou, Y. Dong, L. Feng, and J. Li. LongCite: Enabling LLMs to generate fine-grained citations in long-context QA. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5098–5122, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.264. URL <https://aclanthology.org/2025.findings-acl.264/>. Cited on pages 53 and 54.
- [350] K. Zhang, B. Li, P. Zhang, F. Pu, J. A. Cahyono, K. Hu, S. Liu, Y. Zhang, J. Yang, C. Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 881–916, 2025. Cited on page 2.

- [351] Q. Zhang, A. Garg, J. Foerster, N. Chatterji, K. Malik, and M. Lewis. An empirical study on noisy data and llm pretraining loss divergence. *arXiv preprint arXiv:2602.02400*, 2026.
Cited on page 50.
- [352] R. Zhang, Y. Zhou, Q. Jiang, Q. Song, N. Li, K. Zhou, L. Wang, D. Wang, M. Liao, M. Yang, X. Bai, B. Shi, D. Karatzas, S. Lu, and C. V. Jawahar. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1577–1581, 2019. doi: 10.1109/ICDAR.2019.00253.
Cited on pages 53 and 54.
- [353] R. Zhang, D. Jiang, Y. Zhang, H. Lin, Z. Guo, P. Qiu, A. Zhou, P. Lu, K.-W. Chang, Y. Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024.
Cited on page 66.
- [354] R. Zhang, X. Wei, D. Jiang, Z. Guo, Y. Zhang, C. Tong, J. Liu, A. Zhou, S. Zhang, P. Gao, and H. Li. MAVIS: Mathematical visual instruction tuning with an automatic data engine. In *International Conference on Learning Representations (ICLR)*, 2025. URL <https://openreview.net/forum?id=MnJzJ2gvuf>.
Cited on pages 53 and 54.
- [355] W. Zhang, H. Zhang, X. Li, J. Sun, Y. Shen, W. Lu, D. Zhao, Y. Zhuang, and L. Bing. 2.5 years in class: A multimodal textbook for vision-language pretraining. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4647–4658, 2025.
Cited on pages 4, 53, 54, and 55.
- [356] X. Zhang, C. Li, Y. Zong, Z. Ying, L. He, and X. Qiu. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*, 2023.
Cited on page 66.
- [357] X. Zhang, C. Wu, Z. Zhao, W. Lin, Y. Zhang, Y. Wang, and W. Xie. PMC-VQA: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*, 2023.
Cited on pages 53 and 54.
- [358] Y. Zhang, R. Zhang, J. Gu, Y. Zhou, N. Lipka, D. Yang, and T. Sun. LLaVAR: Enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint arXiv:2306.17107*, 2023.
Cited on pages 53 and 54.
- [359] Y. Zhang, Y. Su, Y. Liu, X. Wang, J. Burgess, E. Sui, C. Wang, J. Aklilu, A. Lozano, A. Wei, L. Schmidt, and S. Yeung-Levy. Automated generation of challenging multiple-choice questions for vision language model evaluation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
Cited on page 66.
- [360] Y.-F. Zhang, H. Zhang, H. Tian, C. Fu, S. Zhang, J. Wu, F. Li, K. Wang, Q. Wen, Z. Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024.
Cited on page 66.
- [361] B. Zhao, B. Wu, M. He, and T. Huang. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087*, 2023.
Cited on pages 2, 53, and 54.
- [362] T. Zheng, G. Zhang, T. Shen, X. Liu, B. Y. Lin, J. Fu, W. Chen, and X. Yue. OpenCodeInterpreter: Integrating code generation with execution and refinement. *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12834–12859, Aug. 2024. doi: 10.18653/v1/2024.findings-acl.762. URL <https://aclanthology.org/2024.findings-acl.762/>.
Cited on pages 53 and 54.

- [363] X. Zheng, D. Burdick, L. Popa, X. Zhong, and N. X. R. Wang. Global table extractor (GTE): A framework for joint table identification and cell structure recognition using visual context. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 697–706, 2021.
Cited on pages 53 and 54.
- [364] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, et al. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
Cited on pages 53 and 54.
- [365] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.
Cited on pages 1, 2, 3, 4, 5, 8, 45, 47, 50, 62, and 86.
- [366] W. Zhu, J. Hessel, A. Awadalla, S. Y. Gadre, J. Dodge, A. Fang, Y. Yu, L. Schmidt, W. Y. Wang, and Y. Choi. Multimodal c4: An open, billion-scale corpus of images interleaved with text. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:8958–8974, 2023.
Cited on page 76.
- [367] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7W: Grounded question answering in images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
Cited on pages 53 and 54.
- [368] Y. Zong, O. Bohdal, T. Yu, Y. Yang, and T. Hospedales. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. *arXiv preprint arXiv:2402.02207*, 2024.
Cited on pages 65 and 66.
- [369] C. Zou, X. Guo, R. Yang, J. Zhang, B. Hu, and H. Zhang. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv preprint arXiv:2411.00836*, 2024.
Cited on page 66.

Appendix

A Contributions	44
B Extended Related Work	45
C Model Architecture Details	47
C.1 Vision Encoder	47
C.2 Projector	47
C.3 Language Model Backbones	47
C.4 End-to-end Parameter Accounting	49
D Training and hyperparameter details	50
D.1 Learning Rate Selection	50
E DCVLM Pool Details	52
E.1 Pool Composition	52
E.2 Per-Dataset Variation	52
E.3 Data Sources and Licensing	55
E.4 Sample visualizations	58
E.5 Multilingual Nature	59
F Evaluation Suite Details	62
F.1 Issues with Grounding Benchmarks	66
G Train-Test Decontamination	68
G.1 Image-Based Decontamination	68
G.2 Text-Based Decontamination	68
G.3 Overall removal rates	71
H Additional Experiments and Details	74
H.1 Filtering rarely helps	74
H.2 Data formatting	75
H.3 Diminishing returns of downstream filtering	76
H.4 Temperature-scaled sampling	77
H.5 Synthetic recaptioning	78
H.6 Online Filtering	79
I Fine-grained sweep of mixture optimization experiments	82
J Packing Implementation Details	83

J.1	Dual constraints	83
J.2	Streaming best-fit bin-packing	83
J.3	Preventing cross-sample contamination	84
J.4	Stateful and resumable packing	84
K	Annotation Infrastructure	85
K.1	Pipeline and Actor Model	85
K.2	Backpressure and Fault Tolerance	85
L	Additional SFT Results	86
L.1	SFT Compute Budget and Learning-Rate Selection	86
L.2	Robustness to the SFT Dataset	87
M	Extended Suite Evaluations	88
N	Safety Preprocessing: Harmful, Toxic and Biased Content	89
N.1	Unsafe, Biased and Toxic Text Removal	89
N.2	Image-level Safety Analysis	89
O	Per-Dataset Core Results	90
P	Limitations and Future Directions	93
P.1	Limitations	93
P.2	Future Work	93

A Contributions

This was a large collaborative effort, and the work spanned data curation, infrastructure, experimentation, and analysis. Below we summarize the main contribution areas. Within each area, contributors are listed in a random shuffled order. An author may appear under multiple areas.

Project coordination. Matteo Farina, Vishaal Udandaraao, Thao Nguyen, Nikhil Parthasarathy, Ludwig Schmidt

Data pool construction. Thao Nguyen, Vishaal Udandaraao, Matteo Farina, Selim Kuzucu, Andreas Hochlehnert, Adhiraj Ghosh, Mehdi Cherti, Karsten Roth, Joschka Struber, Yuhui Zhang, Sebastian Dziadzio, Elaine Sui, Dhruba Ghosh, Hasan Hammoud, Thomas De Min, Simone Caldarella, Sedrick Keh

Data filtering. Vishaal Udandaraao, Matteo Farina, Selim Kuzucu, Thao Nguyen, Maximilian Böther, Andreas Hochlehnert, Marianna Nezhurina, Adhiraj Ghosh, Soumya Jahagirdar, Elaine Sui, Jehanzeb Mirza

Train–test decontamination. Matteo Farina, Vishaal Udandaraao, Maximilian Böther, Adhiraj Ghosh, Marianna Nezhurina

Annotation infrastructure. Maximilian Böther, Matteo Farina, Marianna Nezhurina

Data mixing. Vishaal Udandaraao, Matteo Farina

Training infrastructure and scaling. Matteo Farina, Marianna Nezhurina, Vishaal Udandaraao

Evaluation suite. Matteo Farina, Sebastian Dziadzio, Karsten Roth

Transfer and controlled experiments. Vishaal Udandaraao, Matteo Farina, Thao Nguyen, Andreas Hochlehnert, Adhiraj Ghosh

Writing. Vishaal Udandaraao, Matteo Farina, Adhiraj Ghosh, Soumya Jahagirdar, Massimiliano Mancini, Nikhil Parthasarathy, Ludwig Schmidt

Advising and supervision. Nikhil Parthasarathy, Ludwig Schmidt, Massimiliano Mancini, Jenia Jitsev, Alessio Tonioni, Ameya Prabhu, Sewoong Oh, Matthias Bethge, Elisa Ricci, Ana Klimovic, Federico Tombari, Muhammad Ferjad Naem, Serena Yeung-Levy, Bernt Schiele, Hilde Kuehne

B Extended Related Work

In the main paper, we provided a brief overview of the most relevant recent papers for our work. Here, we provide a deeper dive into these related papers.

Vision-Language Model Training Regimes. The development of modern autoregressive VLMs has converged on a modular architecture, consisting of a pretrained vision encoder, a language model backbone, and a lightweight connector between the two. Early methods differed in how this connection was implemented. Notable works include BLIP-2 [155] which used a Q-former to compress visual tokens and Flamingo [8], which inserted cross-attention layers between frozen vision and language features. The dominant blueprint can be attributed to LLaVA [175–177] which popularized the simpler recipe of “pretraining” the connector on predominantly image-text pairs [12], before conducting supervised fine-tuning (SFT) on curated instruction data.

In contrast, recent works have considerably relaxed these constraints. First, frontier works like InternVL3[365], InternVL3.5[300] and LLaVA-OneVision-1.5 [12] fine-tune all model parameters from scratch. The relationship between these training choices and model scale, image resolution and data composition have been studied too [213, 347]. Concurrently, the focus has also shifted to making the data composition more heterogeneous while training VLMs, moving away from an over-reliance on image-text pairs. Idefics [140, 141] trains on interleaved image-text sequences, UReader uses multimodal documents [329], PaliGemma [21] combines image-text pairs with generated VQA, multi-object detection and OCR, Cambrian [279] includes text-only corpora for preserving language capabilities, etc. Most notably, [310, 365] advocate for using instruction-tuning data during pretraining itself. However, the precise mixture ratios, filtering criteria, and formatting choices that drive these systems remain proprietary or only coarsely documented, motivating our systematic benchmark.

Benchmarking Data Curation. Controlled data-curation benchmarks keep model architectures and training pipelines fixed and only vary the data distribution fed to the model. DataPerf [212] established this paradigm, while DataComp [73] brought it to CLIP pretraining, enabling principled comparison of curation strategies at scale. DCLM [158] extended this to language model pretraining, demonstrating that a simple fasttext classifier trained on high-quality text can substantially improve downstream performance. FineWeb [233] and its educational-quality variant, FineWeb-Edu, showed similar gains through quality-based filtering of Common Crawl. In general, quality-based data filtering has shown strong results for text [158, 233] and image-text pairs [73, 301].

Existing data curation methods can be categorized into two groups: filtering and mixing. Common filtering approaches include CLIP-score filtering [101], image quality assessment [5], text quality classifiers [58, 267], and learned multimodal quality estimators [301]. These filtering methods have proved to be quite effective in driving downstream performance for single data-type (image-text or text-only) datasets, training better CLIP models being an example [73]. Using pretrained data-selector models or multimodal quality scores are more recent approaches to quantify whether a data sample is likely to improve pretraining [42, 131, 199, 303, 304].

Beyond model-based filters, offline curation also comprises deduplication, recaptioning and concept-aware selection. Deduplication ranges from general pruning [264] to semantic deduplication [1, 2]. Recaptioning methods aims to replace weak web-scale alt-text with synthetic or fused captions using VLMs or caption augmentation [66, 79, 163, 225, 226, 334]. Concept-aware methods control the training distribution through concept filtering or balancing [79, 230, 319]. Put together, it has specifically been shown that the offline curation of noisy web-scale data results in large pretraining efficiency gains [31, 68, 73, 112, 200, 290].

Prior works have also explored data mixing instead of filtering, with standard approaches relying on strategies such as domain weighting [227, 317], mixture optimization [18, 37, 59, 121, 179, 330] and temperature-scaled sampling [45, 57]. Despite the efforts in releasing curated datasets (e.g., FINEVISION [310]), there exists no systematic study ablating filtering or mixing strategies in the VLM setting. Our work fills this gap by providing a controlled testbed for multimodal data curation, providing the first *scale-aware* study of data-type mixing for VLMs.

Train-Test (De)contamination. Train-test overlap (contamination) is a well-documented concern, especially in language model evaluation, where several works have demonstrated how benchmark scores can be inflated when test-set or their near-duplicate samples appear in the pretraining data

pool [23, 115, 198, 247, 250]. This is a problem for VLM training as well as the contamination can stem from many sources: text, duplicate or near-duplicate images or documents, etc. To mitigate such concerns, and also to ensure that models do not degrade to rote memorization of the training sets, several works conduct robust decontamination procedures on their training sets [9, 21, 75, 217, 230, 281, 343], i.e, they attempt to remove training examples too similar (or, at worst, identical) to evaluation examples. Some canonical methods include embedding-based similarity search [230, 310], MinHash signatures for approximate text-matching [25, 158, 233] and direct string-search using suffix arrays [145]. In our work, we employ two-way decontamination: a form of embedding-based decontamination for multimodal samples and MinHash signatures for text-only samples.

Scaling Laws and Scale-Aware Curation. An important consequence of scaling-law studies is that a data curation strategy chosen at one scale may not remain optimal at others. A growing body of evidence suggests that the effectiveness of these filters is *scale-dependent*: Goyal et al. [81] and Mizrahi et al. [217] show that optimal filtering aggressiveness decreases with compute budget. Our work successfully extends this finding to the multimodal setting, showing that *at sufficient scale and with optimized mixtures, no individual quality filter provides reliable and consistent gains*.

C Model Architecture Details

All our experiments use a single VLM architecture template, parameterised across the four scales of the scaling ladder (Tab. 1). The template follows the InternVL-3 [365] family: a vision encoder \rightarrow a randomly-initialised MLP projector \rightarrow an autoregressive language-model backbone, with all three components trained jointly from the start (single-stage pretraining, no frozen components). Across our four scales, only the language-model backbone changes; the vision encoder and the projector recipe are held fixed. We document each component in turn.

C.1 Vision Encoder

We use **InternViT-300M-448px-V2.5** [41] for all experiments. It is a Vision Transformer [62] with the modifications introduced in the InternVL series [41, 42, 365], kept identical across our four scales. Tab. 4 reports its key structural choices.

Table 4: **Vision encoder architecture (InternViT-300M-448px-V2.5)**. The same vision encoder is used across all four scales. The encoder is fully unfrozen and updated jointly with the projector and the LM backbone.

Component	Value
<i>Input</i>	
Image resolution	448×448 (one tile; dynamic high-res tiling, see Sec. 3)
Patch size	14×14
Tokens per tile (pre-pixel-shuffle)	$32 \times 32 = 1024$
Tokens per tile (post-pixel-shuffle, fed to LM)	$16 \times 16 = 256$
Channels	3 (RGB)
<i>Transformer trunk</i>	
Depth (layers)	24
Hidden size	1024
Attention heads	16
Head dim	64
FFN intermediate size	4096
FFN activation	GELU
FFN style	2-layer MLP (Linear \rightarrow GELU \rightarrow Linear)
Attention style	Standard multi-head; QKV bias \checkmark , O-proj bias \times
QK normalisation	\times
Normalisation	Pre-LayerNorm, $\varepsilon = 10^{-6}$
Positional embeddings	Learned absolute (interpolated to 448px)
Flash attention	\checkmark (FA-2 [54])
<i>Total</i>	
Parameters	$\sim 304\text{M}$

The tile-based tokenisation produces $32 \times 32 = 1024$ patches per 448×448 tile. After the projector’s pixel-shuffle reduction (Sec. C.2), this becomes $16 \times 16 = 256$ visual tokens per tile ($4\times$ reduction) that are fed to the language model. Multiple tiles and a thumbnail image are concatenated into the LM input, following the dynamic high-resolution scheme of InternVL-2.5 [41].

C.2 Projector

The vision encoder and the language model are bridged by a small randomly-initialised MLP-style projector (often called the “connector” in VLM literature [175, 213]). It is the only module that is randomly initialised at the start of training, everything else is loaded from pretrained checkpoints. Tab. 5 gives the exact structure. Following InternVL-2.5/3, we keep depth and activation fixed across scales; only the projector’s hidden width tracks the LM.

C.3 Language Model Backbones

For the four points on our scaling ladder we use four different sizes from the **Qwen2.5** family [240]: 0.5B, 1.5B, 3B, and 7B parameters. All four share the Qwen2 transformer architecture [324]—

Table 5: **Projector architecture.** The projector is a fixed-depth, fixed-activation 2-layer MLP whose width is the only quantity that varies across scales (it tracks D_{LM} , the language-model hidden size).

Stage	Operation
0. Pre-projection	Pixel shuffle, factor 0.5: 1024 tokens of dim $D_V \rightarrow 256$ tokens of dim $4D_V$
1. Norm	LayerNorm($4D_V$), $\varepsilon = 10^{-5}$
2. Linear-1	Linear($4D_V \rightarrow D_{LM}$), bias ✓
3. Activation	GELU
4. Linear-2	Linear($D_{LM} \rightarrow D_{LM}$), bias ✓

<i>Per-scale projector parameter count</i>	
Small (1B; $D_{LM}=896$)	~4.5M
Medium (2B; $D_{LM}=1536$)	~8.6M
Large (4B; $D_{LM}=2048$)	~12.6M
X-Large (8B; $D_{LM}=3584$)	~27.5M

SwiGLU FFN [256], RMSNorm [344], RoPE position embeddings [268], grouped-query attention (GQA) [6], no QK-normalisation. They differ only in their depth/width/head budget and in two minor configuration knobs (max position length and embedding tying), summarised in Tab. 6. Unless specified, we always initialise from the *Base* (non-Instruct) checkpoints by default.

Table 6: **Language-model backbone architecture across the four scaling-ladder scales.** All are Qwen2.5 base checkpoints [240]. “Head dim” is hidden size divided by query head count. Vocabularies are the standard Qwen2.5 tokenizer.

Scale	Small (1B)	Medium (2B)	Large (4B)	X-Large (8B)
Qwen2.5 size	0.5B	1.5B	3B	7B
<i>Shared structural choices (all scales)</i>				
Architecture family	Qwen2 transformer [324]			
Normalisation	Pre-RMSNorm, $\varepsilon = 10^{-6}$			
QK-normalisation	✗			
FFN style	SwiGLU (gated MLP, SiLU activation)			
Attention style	Grouped-query attention (GQA), QKV bias ✓, O-proj bias ✗			
Positional embeddings	RoPE, base $\theta = 10^6$, no scaling			
<i>Per-scale dimensions</i>				
Layers	24	28	36	28
Hidden size	896	1536	2048	3584
Query heads	14	12	16	28
KV heads (GQA)	2	2	2	4
Head dim	64	128	128	128
FFN intermediate	4,864	8,960	11,008	18,944
<i>Per-scale config knobs</i>				
Max position embedding	32,768	131,072	32,768	131,072
Tied input/output embed.	✓	✓	✓	✗
Vocabulary size	151,936	151,936	151,936	152,064
LM parameters	~494M	~1.54B	~3.09B	~7.62B

A few cross-scale observations are worth flagging because they surface in our scaling experiments:

- **Head dimension is not constant.** The 0.5B model uses 64-dim heads, while 1.5B/3B/7B all use 128-dim heads. Practitioners scaling pretraining recipes should be aware that the small scale therefore has a slightly different attention behaviour than the rest of the ladder, even though the rest of the architecture is uniform.
- **KV-head count is heavily compressed.** GQA ratios are 14:2, 12:2, 16:2, and 28:4 from Small to X-Large—all sub-7B models share the same minimal 2 KV heads.
- **Tied embeddings only at sub-7B.** The 7B model is the only scale where input/output embeddings are not tied. This costs ~50M extra LM parameters at the X-Large scale.

- **Layer count is non-monotonic.** The 3B model is deeper (36 layers) than the 7B model (28 layers); 7B grows primarily by widening (hidden size 2048 \rightarrow 3584) rather than deepening.

These idiosyncrasies are inherited from the Qwen2.5 release and we deliberately do *not* smooth them out, since our purpose is to produce a benchmark whose scaling axis can be reproduced from publicly-released checkpoints rather than to study clean architectural scaling.

C.4 End-to-end Parameter Accounting

Tab. 7 sums the three components per scale, giving the total trainable-parameter count behind the “1B / 2B / 4B / 8B” labels used throughout the paper. The vision encoder and projector together contribute roughly 5–60% of parameters at the small scale and \sim 5% at the X-Large scale.

Table 7: **Total parameter count per scale.** Vision encoder (InternViT-300M-448px-V2.5) is fixed; LM backbone is the corresponding Qwen2.5 size.

	Small	Medium	Large	X-Large
Vision encoder	304M	304M	304M	304M
Projector	4.5M	8.6M	12.6M	27.5M
LM backbone	494M	1.54B	3.09B	7.62B
Total trainable	\sim0.80B	\sim1.85B	\sim3.40B	\sim7.95B
Paper label	1B	2B	4B	8B

All parameters are trained jointly—there is no frozen-encoder pretraining stage, no LoRA [106] adapter, and no separate connector-warmup phase. We refer the reader to Tab. 8 for the optimizer, schedule, and packing settings used during this joint training.

D Training and hyperparameter details

We provide the exact hyperparameters we use for all our training runs in Tab. 8. For the most part, these were derived from the InternVL-2.5 [41] and InternVL-3 [365] configurations. However, we did run a small learning rate (LR) sweep of our own to confirm that these were indeed the best performing on a subset of downstream evaluations.

Table 8: **Pretraining hyperparameters.** All values are fixed across scales unless noted in Tab. 1.

Hyperparameter	Value
<i>Optimization</i>	
Optimizer	AdamW [188] ($\beta_1=0.9, \beta_2=0.999, \epsilon=1e-8$)
Learning rate (pretraining)	2×10^{-5}
LR scheduler	Cosine decay [187]
Warmup ratio	0.03
Weight decay	0.01
Precision	BF16 [56]
Global batch size	1024
Per-device batch size	1
Gradient checkpointing	✓
Parallelism	DeepSpeed ZeRO-1 [243]
<i>Sequence packing</i>	
Max sequence length	8192 tokens
Max packed tokens	8192 tokens
Max packed images	24
Sampling	With replacement
<i>Loss</i>	
Loss reduction	Square-averaging [41]
<i>Architecture</i>	
Vision encoder	InternViT-300M-448px-V2.5
Image resolution	448 × 448 (dynamic tiling)
Pixel shuffle	down-sample ratio 0.5
Include thumbnail image	✓
Vision layer for features	last
Drop path rate	0.0
Connector	2-layer MLP

D.1 Learning Rate Selection

To ensure that the InternVL LR configurations were optimal, we conducted a small LR-sweep ourselves². We select the learning rate by sweeping five values ($2 \times 10^{-4}, 4 \times 10^{-5}, 2 \times 10^{-5}, 8.91 \times 10^{-6}, 2 \times 10^{-6}$) at each model scale using 10B training tokens with the base mixture. All other hyperparameters are held fixed (global batch size 1024, cosine schedule, 3% warmup) according to those specified in Tab. 8. As shown in Tab. 9, $lr = 2 \times 10^{-5}$ achieves the best or second-best performance at every model scale and LM backbone setting. Learning rates above 4×10^{-5} cause training instability—particularly at the 1B scale, where $lr = 2 \times 10^{-4}$ collapses to near-chance performance. This behaviour has also been observed in prior works [246, 311, 351]. Learning rates below 10^{-5} underfit, with the gap widening at larger model sizes. We therefore adopt $lr = 2 \times 10^{-5}$ for all our experiments. This hence also corroborates the LR order-of-magnitude used in InternVL-2.5 and InternVL-3.

²We did this LR-sweep quite early on in the project when we had a slightly different experimental setup: (i) we were using the Qwen-Instruct backbones instead of the Qwen-Base LM, and (ii) we had not yet finalized the validation or core evaluation sets, and hence used 12 randomly selected benchmarks for tracking our average-metric.

Table 9: **Learning rate sweep across model scales.** All runs use 10B training tokens with the base mixture. Bold indicates the best average per model-size and LLM-backbone group. $lr = 2 \times 10^{-5}$ is consistently optimal across scales and backbones.

LR	LLM	MMMU	3DSRBench	A12D	BLINK	COCO	Hall.Bench	MMB-CN	MMB-EN	MMStar	Mantis	TextVQA	SEED	Average
<i>1B model</i>														
2×10^{-4}	Qwen-Inst.	21.0	44.1	24.6	38.6	13.4	29.9	2.2	1.1	25.2	30.4	42.1	26.0	24.9
4×10^{-5}	Qwen-Inst.	30.2	45.3	35.4	37.6	15.1	28.4	35.6	42.4	33.5	35.0	49.6	44.6	36.0
2×10^{-5}	Qwen-Inst.	30.2	45.3	36.7	37.8	15.2	24.7	43.1	42.9	34.7	35.0	44.5	46.0	36.3
8.91×10^{-6}	Qwen-Inst.	30.4	45.0	36.3	35.9	14.8	31.9	35.5	38.7	34.4	36.9	42.2	44.6	35.5
2×10^{-6}	Qwen-Inst.	30.8	45.0	39.6	36.4	12.7	26.8	32.0	35.5	34.1	40.6	13.4	43.2	32.5
<i>2B model</i>														
4×10^{-5}	Qwen-Inst.	38.0	45.1	53.6	39.0	16.6	37.6	57.2	57.5	35.8	45.2	53.1	59.2	44.8
2×10^{-5}	Qwen-Inst.	38.0	45.8	53.6	39.2	18.1	36.6	57.9	59.7	35.8	44.7	53.8	58.2	45.1
8.91×10^{-6}	Qwen-Inst.	40.2	45.1	54.7	38.8	13.7	37.3	57.5	59.3	35.5	47.0	51.7	56.4	44.8
<i>4B model</i>														
2×10^{-4}	Qwen-Inst.	31.2	44.6	44.2	38.4	21.9	34.2	51.9	51.3	38.0	40.6	53.7	54.3	42.0
4×10^{-5}	Qwen-Inst.	40.3	47.4	59.3	39.0	21.1	39.7	63.3	63.1	39.1	51.6	58.8	64.4	48.9
2×10^{-5}	Qwen-Inst.	42.2	46.1	61.4	39.5	20.5	37.5	63.6	64.0	40.5	54.8	59.1	63.7	49.4
8.91×10^{-6}	Qwen-Inst.	42.0	46.4	57.0	38.0	15.7	35.8	58.5	60.3	40.5	49.3	52.1	61.1	46.4
2×10^{-6}	Qwen-Inst.	40.6	45.4	54.8	37.4	17.0	31.3	50.3	51.0	35.2	43.3	19.9	53.4	40.0

E DCVLM Pool Details

Our DCVLM pool aggregates 160 publicly available datasets across four data types: **image-caption pairs** (13 datasets), **multimodal interleaved documents** (5), **text-only** (33), and **multimodal instruction-tuning** (109, spanning 8 capability categories following [310]: Captioning & Knowledge, Chart & Table, General QA, Grounding & Counting, Math, Naive OCR, OCR QA, and Science). Our full pool contains 3.9B samples and 6.0T multimodal tokens, averaging 1.5K tokens per sample. All token counts are measured using the InternVL-2.5 [41] tokenizer over the full pool. The complete per-dataset breakdown of our DCVLM pool is given in Tab. 10 (showing number of *samples* per dataset) and Tab. 11 (showing number of multimodal *tokens* per dataset).

E.1 Pool Composition

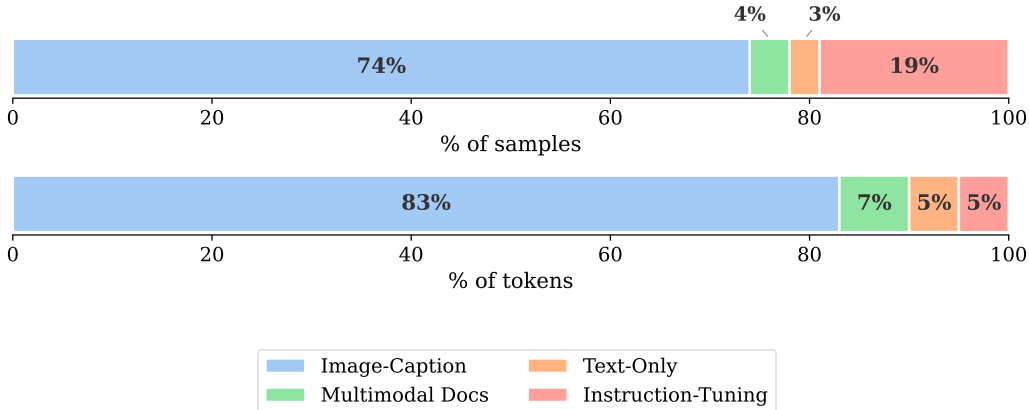


Figure 7: **DCVLM pool composition by data type.** Share of total samples (*top*) vs. share of total multimodal tokens (*bottom*) for each of the four data types. The pool is dominated by image-caption pairs on both axes (83% tokens vs 74% samples). Text-only data exhibits the opposite asymmetry, with 19% of samples but only 5% of tokens. Instruction-tuning data and Multimodal documents are token-dense, i.e. their overall token proportion is much larger than their overall sample proportion, thanks to the presence of potentially many multi-image examples contributing to visual tokens.

Fig. 7 summarises how our pool is split across the four data types, in terms of both samples and multimodal tokens. The detailed per-category sample and token totals are given in Tabs. 10 and 11. The two views differ in informative ways:

- **Image-caption pairs** dominate both axes (74% of samples, 83% of tokens). The token share exceeds the sample share because every image-caption sample contributes 256 visual tokens *per-tile* on top of a short caption, inflating the per-sample token count relative to the text-only data.
- **Text-only** data shows the opposite asymmetry: 19% of samples but only 5% of tokens, because individual text samples are short (440 tok/sample on average) and contribute no visual tokens.
- **Multimodal documents**, despite making up just 4% of samples, contribute 7% of tokens. They are the densest data type, having $\sim 3\text{K}$ tok/sample on average, since each sample typically interleaves several images with surrounding text.
- **Instruction-tuning** data spans 8 capability categories and sits between image-caption pairs and multimodal documents in density ($\sim 2.2\text{K}$ tok/sample). Also in this case, the presence of multi-image samples contributes to a greater per-sample token average.

E.2 Per-Dataset Variation

The 160 datasets in the pool span six orders of magnitude in sample count—from $\sim 1\text{K}$ (e.g. ChartLlama, LIMA, ViQuAE) to $\sim 1.5\text{B}$ (ReLAION-2B-en, DataComp-1B)—and four orders of magnitude in tokens-per-sample (Fig. 8). We dig into the per-data-type statistics below:

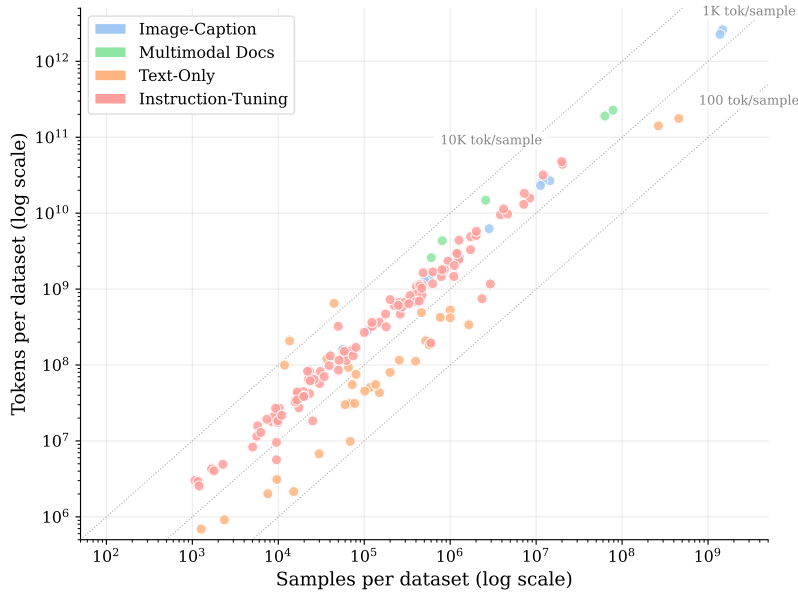


Figure 8: **Samples vs. multimodal tokens (log-log)**. Each marker is one of 160 datasets in our pool, colored by data type. Diagonal reference lines mark constant tokens-per-sample regimes (100, 1K, 10K). Image-caption datasets cluster tightly along the 1–2K tok/sample diagonal driven by the visual-token contribution per image; multimodal documents sit one decade higher (multi-image samples); text-only datasets occupy a much wider band (100–15K tok/sample) reflecting the diversity of short instruction data and long-context corpora; instruction-tuning datasets span the largest dynamic range in size (10^3 – 10^7 samples) but a relatively narrow tokens-per-sample band.

- **Multimodal documents** sit a decade higher, in the 3–6K tok/sample regime, because each sample carries multiple images and longer interleaved text spans.
- **Text-only datasets** occupy the widest tokens-per-sample band of any data type (100–15K), with two distinct clusters: short-form instruction data (Dolly, Unnatural-Instructions, MathQA, GSM8K-Socratic) at ~ 200 – 400 tok/sample, and long-context corpora (LongCite-45K, LongInstruct-Para., LongAlpaca, Long-QLoRA) above 5K tok/sample.
- **Instruction-tuning datasets** span the largest dynamic range in size but a relatively narrow tokens-per-sample band (~ 1 – 3 K). The handful of high-density outliers (MMDU, MapQA, MathWriting) correspond to multi-turn or multi-image conversations.

E.3 Data Sources and Licensing

We source our DCVLM pool from four different data-types, each with a distinct sourcing strategy. Image-caption pairs are primarily sourced from web-crawled image-alt-text corpora (DataComp-1B [73], ReLAION-2B [251]) and synthetic/human-curated caption datasets (PixMo-Cap [57], ShareGPT-4o [34], GRIT [234]). Multimodal documents come from web-crawled interleaved sources (MINT-1T-HTML [14], OmniCC [161], WanJuan [94]) and curated PDF corpora (Multimodal-Textbook [355], MINT-1T-PDF [14]). Instruction-tuning data is aggregated from academic benchmarks with train splits, synthetic generation pipelines, and existing curated datasets across 8 capability categories. Text-only data combines general instruction sets (Dolly, FLAN/FLAN-v2, SlimOrca) with long-context (LongAlpaca, LongCite-45K) and code/math reasoning (Numina-Math, MetaMathQA, Glaive-Code) corpora. In Tab. 12, we provide the licensing information and original sources from which we collected each sub-dataset in our DCVLM pool.

Table 12: **DCVLM pool per-dataset licenses.** Licenses are grouped according to the datamix categories: **captioning data**, **multimodal documents**, **visual instruction-tuning data**, and **text-only data**. “Unknown” indicates that a public license was not clearly specified in the listed source.

Dataset	License	Source
Captioning		
ReLAION-2B-en	CC BY 4.0 (gated; access requires accepting HF terms/contact sharing)	source
DataComp-1B	CC BY 4.0	source
AS-100M	Apache-2.0	source
GRIT (Cap.)	MS-PL	source
InternVL-SA1B	MIT	source
FaceCaption-15M	CC BY 4.0 + research/education notice	source
PixMo-Cap	ODC-BY-1.0	source
ShareGPT-4o	MIT; HF-gated contact-sharing/access terms; source video copyrights/platform terms and academic-research notice apply	source
TextOCR-GPT4V	Apache-2.0	source
TextCaps	CC BY 4.0	source 1; source 2
COCO (Cap.)	CC BY 4.0 for annotations; images retain original COCO/Flickr licenses	source
OpenImages (Cap.)	CC BY 4.0 for annotations; images retain original Open Images licenses	source
SEA-VL	CC BY-SA 4.0	source 1; source 2
Multimodal Docs		
MINT-HTML	CC BY 4.0	source
MINT-PDF	CC BY 4.0	source
OmniCC	CC BY 4.0	source
Multimodal Textbook	Apache-2.0	source
WanJuan	CC BY 4.0	source
Cap. & Know.		
Art500K	Custom non-commercial research-only terms; images retain third-party rights	source 1; source 2
LLaVA-595K	Other: must comply with CC-3M license and BLIP license (HF tag: other)	source
MMInstruct	Apache-2.0	source
ShareGPT4V	CC BY-NC 4.0 + OpenAI ToU	source
SVIT	CC BY 4.0; OpenAI ToU and original Visual Genome/MS-COCO image/annotation licenses apply; HF-gated usage notice applies	source
Chart & Table		
BigDocsBench	CC BY 4.0 for ServiceNow-generated parts; per-sample upstream terms and Llama 3.1 terms may apply	source
Chart2Text	Unknown / not publicly specified	source 1; source 2
ChartGemma	Unknown / not publicly specified	source
ChartLlama	Unknown / not publicly specified	source
ChartQA	Apache-2.0 for Cambrian-10M formatted version; original ChartQA terms may also apply	source 1; source 2
ChartX	Apache-2.0	source
CoSyn-400K	ODC-BY-1.0 (plus AI-generated-output/provider terms stated in card)	source
DocStruct4M	Apache-2.0	source
DVQA	Apache-2.0 for Cambrian-10M formatted version; original DVQA terms may also apply	source 1; source 2
FigureQA	Unknown / dataset files license not clearly stated; generation code MIT	source
FinTabNet	CDLA-Permissive-2.0	source
MMC-Instruct	CC BY-SA 4.0	source
PixMo-Docs	ODC-BY-1.0 (plus AI-generated-output/provider terms stated in card)	source
PlotQA	CC BY 4.0	source 1; source 2; source 3; source 4
PosterSum	Unknown / not publicly specified	source
SBSFigures	Unknown / not publicly specified	source
SciGraphQA	MIT	source
SimChart9K	Unknown / not publicly specified	source
SPIQA	CC BY 4.0	source
TabMWP	CC BY-NC-SA 4.0 for TabMWP dataset; MIT for repository code	source
UniChart	MIT for UniChart-pretrain-images; UniChart-pretrain-data license not publicly specified	source 1; source 2
VisText	Unknown / not publicly specified on The Cauldron repo; original VisText terms may apply	source
General QA		
AlgoPuzzleVQA	Apache-2.0	source
ALLaVA	CC BY-NC 4.0	source
A-OKVQA	Apache-2.0 for official repository; dataset archive has no separate license file; COCO image licenses apply	source
Cambrian-GPT4o	Apache-2.0	source
EST-VQA	Unknown / not publicly specified	source 1; source 2
GQA	CC BY 4.0	source 1; source 2

Continued on next page

Table 12: DCVLM pool per-dataset licenses and sources (continued).

Dataset	License	Source
Hateful Memes	Custom non-commercial/research dataset terms	source
IconQA	CC BY-NC-SA 4.0	source
iNaturalist-2018	Mixed image licenses; check per-image iNaturalist metadata	source 1; source 2; source 3; source 4
LVIS-Instruct4V	Unknown / not publicly specified	source
MMDU	CC BY-NC 4.0 + OpenAI ToU	source
OK-VQA	CC BY 4.0 for annotations; images retain original COCO/Flickr licenses	source 1; source 2
ProVision-10M	CC BY-NC 4.0	source
SoM-LLaVA	Apache-2.0	source
Spot-the-Diff	Unknown / not publicly specified on The Cauldron repo; original Spot-the-Diff terms may apply	source
ViQuAE	Unknown / not publicly specified	source 1; source 2
VisDial	Unknown / not publicly specified	source
Visual7W	MIT for Visual7W toolkit/repo; COCO image licenses apply; annotation license not separately specified	source 1; source 2; source 3
VQAv2	CC BY 4.0 for annotations; images retain original COCO/Flickr licenses	source
VSR	Apache-2.0	source
Grounding & Counting		
All-Seeing-V2	Apache-2.0	source
LRV-Instruction	BSD-3-Clause for repository; source image/data terms may also apply	source
Objects365	Academic-purpose only; annotations/website CC BY 4.0; images under Flickr terms and must not be redistributed; software MIT	source 1; source 2
PixMo-Points	ODC-BY-1.0	source
RefCOCO/+g	MS COCO image licenses; annotations license not clearly specified	source 1; source 2; source 3
TallyQA	Apache-2.0 for TallyQA repo/annotations; referenced VQA 2.0/Visual Genome terms may apply	source
TolokaVQA	CC BY 4.0; images from CC BY-licensed MS COCO subset	source
V3Det	CC BY 4.0 for annotations/category tree/tools; Flickr/image terms for images	source
Math		
CLEVR-Math	CC BY 4.0	source
Geometry3K	Apache-2.0	source
GeomVerse	Unknown / not publicly specified on The Cauldron repo; original GeomVerse terms may apply	source
GeoQA+	Apache-2.0	source
MAVIS-Function	Unknown / not publicly specified	source
MAVIS-Geometry	Unknown / not publicly specified	source
UniGeo (Calc.)	Unknown / not publicly specified	source
UniGeo (Proof)	Unknown / not publicly specified	source
Naive OCR		
AnyWord-3M	Apache-2.0	source
ArT	Unknown / not publicly specified	source
CASIA	Other / free for non-commercial use; Kaggle mirror license is other; original CASIA terms apply	source
Chinese-OCR	Unknown / not publicly specified	source
COCO-Text	CC BY 4.0	source
CTW	Unknown / not publicly specified	source 1; source 2
EATEN	Unknown / not publicly specified	source
HME-100K	Apache-2.0	source
IAM	Custom non-commercial/research license	source
LSVT	Unknown / not publicly specified	source
MTWI	Unknown / not publicly specified	source 1; source 2
ParSynth-OCR-200K	Unknown / not publicly specified	source
POIE	Unknown / not publicly specified	source
ReCTS	Unknown / not publicly specified	source
RenderedText	Unknown / not publicly specified	source
SROIE-2019	Unknown / not publicly specified	source
SynthDoG	MIT for SynthDoG code; generated dataset license not specified on listed HF repos	source 1; source 2; source 3; source 4
SynthText	Custom research-only/non-commercial terms	source
OCR QA		
ArXivQA	CC BY-SA 4.0	source
Docmatix	MIT	source
DocReason25K	Apache-2.0	source
DocVQA	Apache-2.0 for formatted HF version; original DocVQA terms may also apply	source 1; source 2
InfoVQA	Apache-2.0	source
KVQA	Unknown / not publicly specified	source
LLaVAR	CC BY-NC 4.0; research-only/non-commercial; CLIP/LLaMA/Vicuna/GPT-4/LLaVA terms may apply	source 1; source 2; source 3; source 4

Continued on next page

Table 12: DCVLM pool per-dataset licenses and sources (continued).

Dataset	License	Source
MapQA	Unknown / not publicly specified on The Cauldron repo; original MapQA terms may apply	source
MathWriting	CC BY-NC-SA 4.0	source
MultiUI	ODC-BY-1.0; HF-gated contact-sharing/access terms; public-web source content and LLM-provider terms may apply	source
OCR-VQA	Unknown / not publicly specified	source
Screen2Words	CC BY 4.0	source
ST-VQA	Unknown / not publicly specified	source
TextOCR	CC BY 4.0	source 1 ; source 2
TextVQA	Apache-2.0 for Cambrian/LLaVA formatted version; original TextVQA terms may also apply	source 1 ; source 2
VisualMRC	Unknown / not publicly specified	source
Science		
A12D	Apache-2.0 for Cambrian/LLaVA formatted version; original A12D terms may also apply	source 1 ; source 2
ImageCLEF	Unknown / not publicly specified	source
LLaVA-Med (FT)	CC BY-NC 4.0; research/non-clinical-use restrictions; LLaMA/Vicuna/GPT-4 terms may apply	source
LLaVA-Med (PT)	CC BY-NC 4.0; research/non-clinical-use restrictions; LLaMA/Vicuna/GPT-4 terms may apply	source 1 ; source 2 ; source 3 ; source 4 ; source 5
PathVQA	MIT	source
PMC-VQA	CC BY-SA (source PMC OA images/articles CC0 or CC BY)	source
ScienceQA	CC BY-SA 4.0	source
SLAKE	CC BY 4.0	source
TQA	CC BY-NC 3.0	source
VisualWebInstruct	Apache-2.0	source
VQA-RAD	CC0-1.0	source
WebSight	CC BY 4.0 + source-content licenses/disclosure condition	source
Text		
FLAN	CC BY 4.0 (Open-Orca/FLAN HF repo)	source
FLAN-v2	Apache-2.0	source
SlimOrca	MIT	source
UltraChat-200K	MIT	source
UltraFeedback	MIT	source
WizardLM-Evol-70K	MIT	source
LIMA	Other; source-stricter license if applicable, otherwise CC BY-NC-SA	source
No Robots	CC BY-NC 4.0	source
Unnatural Instr.	MIT	source
MOSS	CC BY 4.0	source
Llama3-Magpie-Pro	Llama 3 license (HF license tag: llama3)	source
Magpie-Qwen2-Pro	Unknown / not publicly specified on listed HF repo; generated with Qwen2, so Qwen terms may apply	source
Firefly	Unknown / not publicly specified	source
Dolly	CC BY-SA 3.0	source
KOpen-Hermes-25	MIT	source
OpenAI-TLDR	Unknown / source OpenAI TL;DR data terms not clearly specified	source
Saraswati-CoT	OpenRAIL	source
CodeFeedback	Apache-2.0 for source m-a-p/Code-Feedback; listed HF formatted repo has no license tag; OpenAI usage policy may apply	source
Glaive-Code	Apache-2.0	source
xCoder-80K	Unknown / not publicly specified	source
LeetCode	Llama 2 license (HF license tag: llama2)	source
Evol-Code	CC BY-NC-SA 4.0	source
LongCite-45K	Apache-2.0	source
LongInstruct-Para.	CC BY-SA 4.0	source
Long-QLoRA	Unknown / no license specified on listed HF repo; source dataset licenses may apply	source
LongAlpaca	CC BY-NC 4.0 for data/weights; research/non-commercial only	source
GSM8K (Socratic)	MIT	source
MetaMathQA	MIT	source
MathQA	Apache-2.0	source
Numina-Math-1.5	Apache-2.0	source
Numina-Math-T1R	Apache-2.0	source
Orca-Math	MIT	source
InfinityMath	Apache-2.0	source

E.4 Sample visualizations

To enable quick and easy visualization of random samples from each of our 160 source datasets, we provide a visualization tool built with Streamlit here: <https://dcvml-dataset-browser.streamlit.app/>.

The tool enables browsing samples from datasets, categorized by data type and afforded capability (for multimodal instruction tuning datasets).

E.5 Multilingual Nature

As can be seen from Tab. 12, the DCVLM pool is sourced primarily from English-centric web corpora (DataComp [73], ReLAION-2B-en [251]) and English instruction-tuning datasets, but it inherits a non-trivial multilingual tail from datasets that target other languages explicitly (e.g. Chinese instruction corpora, multilingual OCR generators) and from the natural multilingual contamination present in any large web crawl. To quantify this, we annotate every sample in the pool with two independent language detectors and take their top-1 prediction.

Annotation methodology. For each sample we run two language detectors over the per-sample text (caption, instruction, or document content):

- **Lingua**³: a lightweight character- n -gram detector covering 75 languages with good coverage of European tongues but limited support for languages with non-Latin scripts.
- **NLLB-218** [50]: the multilingual classifier shipped with the No-Language-Left-Behind translation model, covering 218 languages and substantially better than Lingua on Cyrillic, Devanagari, and Southeast-Asian scripts.

Each annotation stores the top-3 language hypotheses with calibrated scores. For the analysis below we report only the top-1 prediction from each detector. Both detectors operate on text with images stripped, so they describe the linguistic distribution of the *textual* content of the pool, not of any visual content.

Pool-wide distribution. Figure 9 shows the top-15 languages by sample count under each detector, over our small pool (each dataset is capped at 100,000 samples as a representative subset). We notice that the distribution is heavily tailed—English alone accounts for 91.1% (Lingua) / 92.8% (NLLB) of all samples. The single largest non-English language is Chinese at 3.9% / 3.5%. Beyond Chinese, the remaining languages are spread thinly: French, German, Japanese, Korean, and Russian each contribute 0.2%–0.8%, and the rest of the long tail (a further 70-110 languages depending on detector) sums to under 1%. Despite the long tail Lingua and NLLB largely agree—the two detectors return the same top-1 language on 96.6% of samples—so our statistics are robust to the choice of detector.

Where does the non-English content come from? Roughly half of the pool’s non-English samples trace to a small number of datasets that were either built for a specific non-English language or rely on synthetic multilingual generation:

- **Chinese:** firefly (a general-purpose Chinese instruction corpus, ~91% non-English, contributing ~33% of all Chinese samples), wanjuan (curated Chinese multimodal interleaved documents, ~84% non-English), internvl_sa1b_caption (Chinese-translated SA-1B captions), and magpie_qwen2_pro (Chinese instruction synthesis).
- **French / German / Russian:** dominated by flanv2, whose translation tasks bundle large numbers of parallel corpora—alone responsible for 72.4% of the pool’s French, 78.3% of its German, and 64.6% of its Russian samples.
- **Japanese / Korean:** synthdog, a synthetic OCR-style image-text generator, supplies 76% of Japanese and 88% of Korean samples; the remaining mass comes from eaten, mavis_geometry, and small instruction sets such as kopen-hq-hermes-25-60K (100% non-English).
- **Vietnamese / Arabic / Indonesian:** largely supplied by sea-v1, a Southeast-Asian vision-language dataset (93% Vietnamese, with smaller contributions to Arabic and Indonesian).

The remaining non-English content is the natural multilingual residue of the web crawls—although relaion-2b-en and datacomp_1b are English-filtered upstream, ~5% of their captions are still flagged as non-English by NLLB, predominantly European languages (German, French, Italian, Dutch). This also corroborates the findings from Nguyen et al. [226] and Pouget et al. [237].

³<https://github.com/pemistahl/lingua-py>

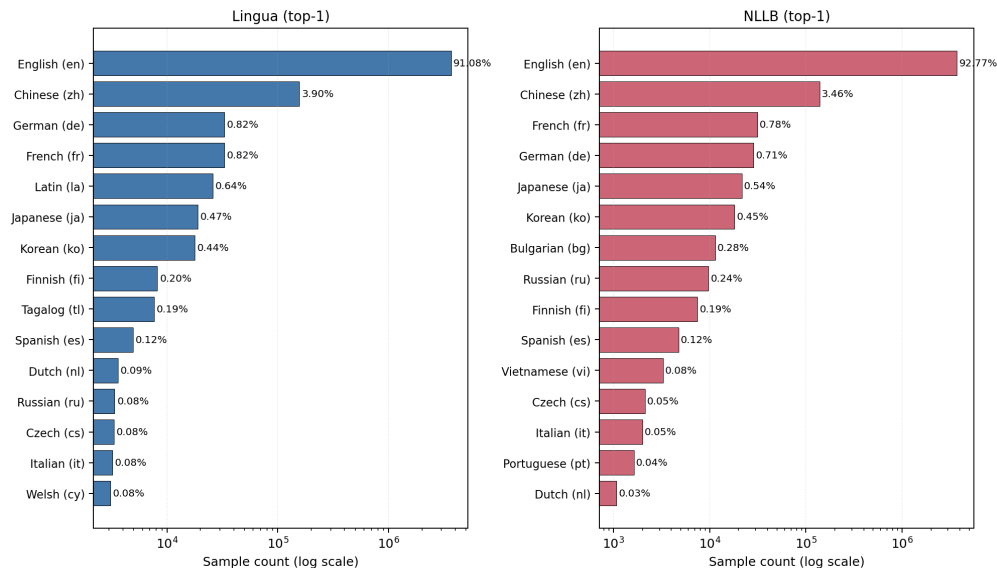


Figure 9: **Language distribution of DCVLM pool.** Top-15 languages by per-sample top-1 prediction from **Lingua** (left, blue) and **NLLB** (right, red), shown on a log-scaled x-axis. The pool is dominated by English (91.1% Lingua / 92.8% NLLB), followed by Chinese (3.9% / 3.5%); the remaining $\sim 5\%$ is spread across 73 languages (Lingua) or 114 languages (NLLB), with French, German, Japanese, Korean, and Russian forming the next tier.

Implications. The pool is overwhelmingly English by design: this matches the practice of most recent open VLM pretraining corpus we are aware of and reflects the asymmetry of available web data [231, 284]. The Chinese sub-pool is large enough to plausibly support Chinese-capable downstream models, as we show in our experiments in Sec. 4. Languages outside the top five are too sparse to sustain meaningful pretraining signal on their own—this is a limitation of the underlying source data rather than of DCVLM per se, and addressing it would require deliberate ingestion of non-English vision-language datasets.

How much does English data drive downstream performance? To test how dominant the English pretraining data is in our pool, we run a set of language ablation experiments at the medium scale (2B model, 25B tokens). Using the per-sample NLLB top-1 language annotations described above, we partition the pool into English and non-English samples and construct three training sets, holding the data mixture fixed: (i) **All languages**, our default pool with no language-based selection; (ii) **English-only**, which retains only samples whose detected language is English; and (iii) **Non-English-only**, which retains only the complementary non-English tail. Since English accounts for $\sim 92\%$ of the pool (Fig. 9), the English-only set is close in size to the full pool, whereas the non-English-only set is far smaller and is consequently seen with more repetitions to fill the same 25B-token budget. We report core-set results in Tab. 13.

Table 13: **Language ablation experiments.** We partition the DCVLM pool by language and train on (i) all languages, (ii) English-only samples, and (iii) non-English-only samples. The English-only model nearly matches the full pool, while training only on the non-English tail is substantially worse. Multilingual benchmark performance is essentially unchanged across all three pools.

Training pool	Gen	Know	OCR	Vision	MTL	Text	Core Avg
All languages (no filter)	58.5	58.4	41.8	44.3	39.8	48.2	49.1
English-only	57.3	58.5	41.5	41.9	39.9	48.4	48.5
Non-English-only	53.2	56.6	38.5	42.0	39.1	45.4	46.1

Three observations stand out. First, **English data carries almost all of the signal**: the English-only model trails the full pool by only 0.6pp, despite discarding $\sim 8\%$ of the pool. Second, **the non-English tail alone is insufficient**: training exclusively on non-English samples drops performance by

3.0pp relative to the full pool, with the largest regressions on General (−5.3pp), OCR (−3.3pp), and Text-Only (−2.8pp). Third, and most notably, **multilingual benchmark performance is essentially flat across all three pools** (39.8 / 39.9 / 39.1): neither removing non-English data nor training solely on it meaningfully moves the Multilingual score.

We caution, however, that two confounders complicate the interpretation of this experiment: *(i)* since English makes up $\sim 92\%$ of the pool, the non-English-only run draws from far fewer unique samples and incurs substantially more repetition than the English-only run to fill the same token budget, so its performance drop cannot be cleanly attributed to language as opposed to repetition; and *(ii)* our per-sample language annotations are noisy—especially for multilingual samples, where the detector is biased toward the head (high-resource) languages—making the English/non-English partition imperfect.

F Evaluation Suite Details

In this section, we highlight our protocol to select and categorize benchmarks for evaluating models. We start with a candidate pool of 65 benchmarks, which we categorize according to the majority consensus of previous work: Cambrian-1 [279], InternVL2.5 & InternVL-3 [41, 365], MM-1 [213], Qwen2-VL & Qwen2.5-VL [17, 297] into 9 distinct categories:

- General Understanding: this category includes established multimodal benchmarks, as well as comprehensive, all-round benchmarks assessing several capabilities at once.
- Knowledge-Centric benchmarks: these test factual, scientific and/or educational knowledge.
- OCR & Charts, testing the ability of models to read and understand visually rendered characters, diagrams, layout, and PDFs.
- Vision-Centric benchmarks, requiring models to perceive spatially, perform fine-grained visual processing such as counting, and compare items across multiple images.
- Multilingual benchmarks, which assess general knowledge in less common languages.
- Text-Only benchmarks, testing whether the unimodal, language-only capabilities of the language model are preserved;
- Safety benchmarks, which measure the ability of models to detect and, potentially, refuse to respond to, unsafe queries (e.g., violent, harmful and/or explicit content).
- Hallucination benchmarks, stress-testing the capabilities of models to assess evident facts such as the existence of explicit objects in images;
- Reasoning benchmarks, with a math- and logic-centric focus (including visual tasks such as rendered geometry problems and language-only tasks such as GSM-8K [46]).

Prior works are far from unanimous in how they organize these benchmarks: the same task is frequently placed in different categories by different works. For instance, AI2D is treated as a Knowledge benchmark by Cambrian-1 but as an OCR/document task by Qwen2-VL and InternVL, and MMMU is variously labelled as Knowledge, Mathematics, or Reasoning. To obtain a single consistent taxonomy, we collect the category each prior work assigns to every benchmark and resolve disagreements by majority vote wherever a consensus exists, adjudicating the remaining ambiguous cases ourselves. Table 14 reports, for each benchmark, the verbatim category given by every prior work alongside our final DCVLM assignment. Table 15 then lists the resulting mapping from each DCVLM category to its benchmarks. Whenever possible, we use benchmark implementations in VLMEvalKit [64] and OpenCompass [49], which we integrate into the toolkit we plan to release.

Table 14: **Benchmark categorization across prior works versus DCVLM.** For each benchmark in our candidate pool we show the category assigned by Cambrian-1 [279], MM-1 [213], Qwen2-VL/Qwen2.5-VL [17, 297], and InternVL2.5/InternVL-3 [41, 365], each *verbatim* as reported by that work. Rows are grouped by the unified DCVLM category. Prior works frequently disagree (e.g. AI2D is placed in Knowledge, OCR, or Diagram understanding by different works). We resolve each benchmark by majority consensus where one exists and adjudicate ambiguous cases ourselves. “–” marks a benchmark not categorized (or not used) by that work.

Benchmark	Cambrian-1	MM-1	Qwen2/2.5-VL	InternVL2.5	InternVL-3
General Understanding					
MM-Bench	General	–	General VQA	General	General
MM-Bench v1.1	General	–	General VQA	General	General
MME	General	–	General VQA	General	General
GQA	General	–	–	–	–
SEEDBench-IMG	General	–	–	–	–
MMT-Bench	–	–	General VQA	–	–
MMStar	–	–	General VQA	General	General
COCO-Caption	–	–	–	–	–
Knowledge					

continued on next page

Benchmark	Cambrian-1	MM-1	Qwen2/2.5-VL	InternVL2.5	InternVL-3
MMMU	Knowledge	–	Maths, Edu	Reasoning & Maths	Reasoning & Maths
ScienceQA	Knowledge	–	–	–	–
AI2D	Knowledge	–	OCR, Docs, Diagram	OCR, Charts & Docs	OCR, Charts & Docs
OCR & Charts					
CharXiv	–	–	–	OCR, Charts & Docs	OCR, Charts & Docs
DocVQA	OCR & Charts	–	OCR, Docs, Diagram	OCR, Charts & Docs	OCR, Charts & Docs
ChartQA	OCR & Charts	–	OCR, Docs, Diagram	OCR, Charts & Docs	OCR, Charts & Docs
OCRBench	OCR & Charts	–	OCR, Docs, Diagram	OCR, Charts & Docs	OCR, Charts & Docs
TextVQA	OCR & Charts	–	OCR, Docs, Diagram	OCR, Charts & Docs	OCR, Charts & Docs
InfoVQA	–	–	OCR, Docs, Diagram	OCR, Charts & Docs	OCR, Charts & Docs
SEEDBench2+	–	–	–	OCR, Charts & Docs	OCR, Charts & Docs
OCRVQA	–	–	–	–	–
Vision-Centric					
CV-Bench	Vision-Centric	–	–	–	–
Mantis-Eval	–	–	–	Multi-Image Understanding	Multi-Image Understanding
3DSRBench	–	–	–	–	–
MMT-Bench-MI	–	–	–	Multi-Image Understanding	Multi-Image Understanding
BLINK	–	–	–	Multi-Image Understanding	Multi-Image Understanding
MuirBench	–	–	–	Multi-Image Understanding	Multi-Image Understanding
NaturalBench	–	–	–	–	–
MMVP	Vision-Centric	–	–	–	–
RealWorldQA	Vision-Centric	–	General VQA	Real-World Comprehension	Real-World Comprehension
MME-RealWorld	–	–	–	Real-World Comprehension	Real-World Comprehension
VizWiz	–	–	–	–	–
Multilingual					
MMMB	–	–	–	Multilingual	Multilingual
MTL-MMBench	–	–	–	Multilingual	Multilingual
MTVQA	–	–	OCR, Docs, Diagram	Multilingual	Multilingual
Reasoning					
MathVista	Knowledge	–	Maths, Edu	Reasoning & Maths	Reasoning & Maths
MathVision	–	–	Maths, Edu	Reasoning & Maths	Reasoning & Maths
MathVerse	–	–	–	Reasoning & Maths	Reasoning & Maths
WeMath	–	–	–	–	Reasoning & Maths
DynaMath	–	–	–	–	Reasoning & Maths
GSM8K	–	–	Maths	Maths	Maths
MATH	–	–	Maths	Maths	Maths
TheoremQA	–	–	–	Maths	Maths

continued on next page

Benchmark	Cambrian-1	MM-1	Qwen2/2.5-VL	InternVL2.5	InternVL-3
Text-Only					
GaoKao	-	-	-	Comprehensive	Comprehensive
C-EVAL	-	-	-	Comprehensive	Comprehensive
MMLU	-	-	-	Comprehensive	Comprehensive
CMMLU	-	-	-	Comprehensive	Comprehensive
TriviaQA	-	General	-	General	General
NaturalQ	-	-	-	General	General
C3	-	-	-	General	General
RACE	-	-	-	General	General
Winogrande	-	General	-	Reasoning	Reasoning
Hellaswag	-	General	-	Reasoning	Reasoning
HumanEval	-	-	Coding	Coding	Coding
Hallucination					
AMBER	-	-	-	-	-
HallusionBench	-	-	General VQA	Hallucination	Hallucination
CRPE	-	-	-	Hallucination	Hallucination
POPE	-	-	-	Hallucination	Hallucination
Safety					
MMSafetyBench	-	-	-	-	-
MLLMGuard	-	-	-	-	-
VLGuard	-	-	-	-	-

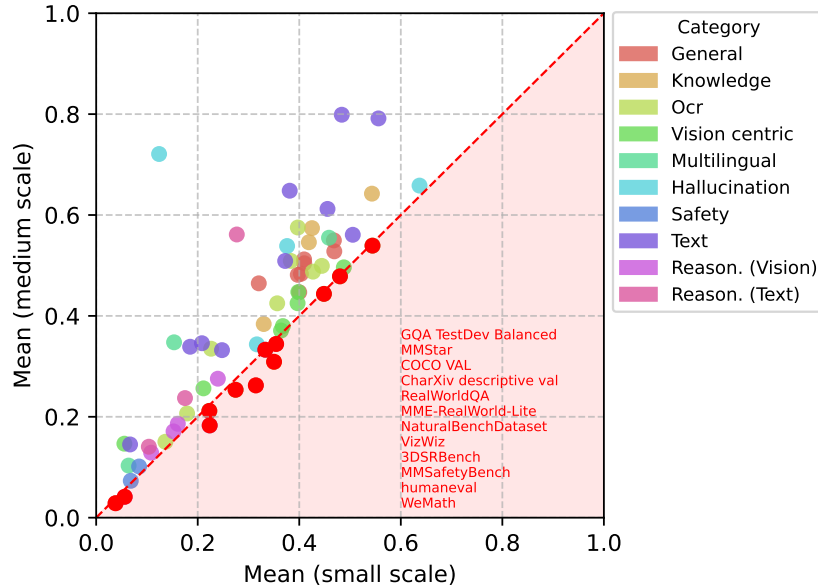


Figure 10: **Not all benchmarks monotonically increase with scale.** We report the performance of 65 benchmarks (averaged over three runs at both the small and the medium scales) with a focus on their scaling behaviour. Average performance at the small scale is arranged on the x-axis; the y-axis displays average performance at the medium scale. Benchmarks falling below the identity line (highlighted in red in the figure) do *not* improve from the small to the medium scale.

Monotonicity selection. We filter 12 out of 65 benchmarks because they do not pass our first selection criterion: *monotonicity*. To ensure reliable evaluation of data-centric decisions across scales, we filter our initial benchmark pool to retain only tasks that exhibit *consistent improvement* with scale. Concretely, we train models at two scales (small and medium) for three different random seeds, for a total of six runs. We evaluate all resulting checkpoints and compute the per-scale average. Since DCVLM is a scale-centric benchmark, we rule out benchmarks for which performance does *not*

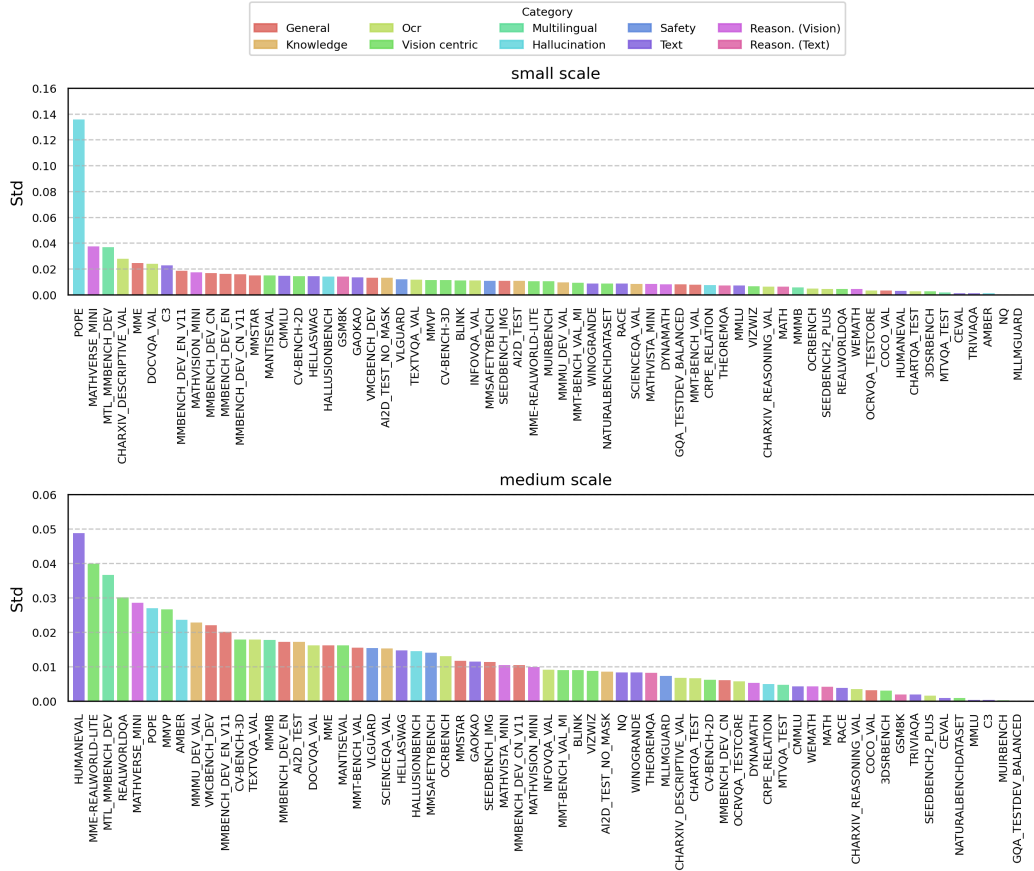


Figure 11: **Seed variance of the initial 65-benchmark pool.** We conduct runs at `small` and `medium` scales, to measure the seed variance of benchmarks in our initial 65 candidates. We remove POPE [164], as it leads to a standard deviation of up to 16% over three runs at the `small` scale.

monotonically improve from small to medium scale, as such benchmarks would risk obscuring the signal attributable to data curation choices. This yields a curated subset of benchmarks (shown in Fig. 10 as points above the diagonal) that exhibit reliable scale-monotonic behavior.

Stability selection. In parallel to assessing monotonicity, we also search for stable evaluations. For this reason, we further inspect the *seed variance* of benchmarks using the same six runs. We perform stability filtering in two steps: globally and locally within each category.

To provide an initial global picture, we display the standard deviation for all the initial 65 benchmarks at both scales in Fig. 11. We find striking instability with POPE [164] compared to all other benchmarks (16% seed variance over three runs at the `small` scale), and we therefore discard it from our pool. This decreases the total from 53 to 52 remaining benchmarks, which we jointly call our **Extended set**.

Extended, Core, and Validation suites. Since we are primarily interested in evaluating pretrained models (according to our definition in Sec. 1), we create a smaller subset of tasks by excluding verticals that are mostly of interest for post-training strategies than they are for pretraining [86, 368]: Safety, Hallucination, and Reasoning benchmarks. We then apply a second iteration of stability filtering that removes the most unreliable tasks per category according to the average of their standard deviation across small and medium scale runs. This corresponds to removing: (i) MME [71] from the General category, (ii) TextVQA and DocVQA from the OCR & Charts category, (iii) C3 and HellaSwag from the Text-Only category, (iv) MMVP and MantisEval from the Vision-Centric category. We do not remove additional tasks from the Knowledge-Centric and the Multilingual categories, as they already contain a more modest number of tasks compared to other categories.

Table 15: The initial unfiltered pool of 65 benchmarks we started from.

Category	Benchmarks
General	MMBench-DEV-EN, MMBench-DEV-CN, MMBench-DEV-EN-V11, MMBench-DEV-CN-V11 [182], MME [71], GQA-TestDev-Balanced [109], SEEDBench-IMG [148], MMT-Bench-VAL [331], MMStar [35], COCO-VAL [39], VMCBench-DEV [359]
Knowledge	MMM-DEV-VAL [338], ScienceQA-VAL [192], AI2D-TEST, AI2D-TEST-NO-MASK [127]
OCR	DocVQA-VAL [209], ChartQA-TEST [206], OCRBench [183], TextVQA-VAL [261], InfoVQA-VAL [210], SEEDBench2-Plus [149], OCRVQA-TESTCORE [215], CharXiv-Descriptive-VAL, CharXiv-Reasoning-VAL [306]
Vision-Centric	MMT-Bench-VAL-MI [331], BLINK [72], MUIRBench [292], RealWorldQA [314], MME-RealWorld-Lite [360], NaturalBenchDataset [150], VizWiz [91], 3DSRBench [196], MantisEval [114], MMVP [280], CV-Bench-2D, CV-Bench-3D [279]
Multilingual	MTVQA-TEST[274], MMBB[269], MTL-MMBench-DEV[182]
Hallucination	HallusionBench[85], CRPE-Relation[298], POPE [164], AMBER[294]
Safety	VLGuard[368], MLLMGuard[84], MMSafetyBench[180]
Text	C3[321], C-Eval[107], CMMLU[153], GaoKao[356], HellaSwag[340], HumanEval[36], MMLU[98], NQ[136], RACE[137], TriviaQA[116], WinoGrande[248]
Reasoning	MathVista-MINI[193], MathVision-MINI[296], MathVerse-MINI[353], WeMath[239], DynaMath[369], GSM8K[47], TheoremQA[38], MATH[99]

Table 16: List of benchmarks in our 52-task **Extended** suite.

Category	Benchmarks
General	MMBench-DEV-EN, MMBench-DEV-CN, MMBench-DEV-EN-V11, MMBench-DEV-CN-V11, MME, SEEDBench-IMG, MMT-Bench-VAL, VMCBench-DEV
Knowledge	MMM-DEV-VAL, ScienceQA-VAL, AI2D-TEST, AI2D-TEST-NO-MASK
OCR	DocVQA-VAL, ChartQA-TEST, OCRBench, TextVQA-VAL, InfoVQA-VAL, SEEDBench2-Plus, OCRVQA-TESTCORE, CharXiv-Reasoning-VAL
Vision-Centric	MMT-Bench-VAL-MI, BLINK, MUIRBench, MantisEval, MMVP, CV-Bench-2D, CV-Bench-3D
Multilingual	MTVQA-TEST, MMBB, MTL-MMBench-DEV
Hallucination	HallusionBench, CRPE-Relation, AMBER
Safety	VLGuard, MLLMGuard
Text	C3, C-Eval, CMMLU, GaoKao, HellaSwag, MMLU, NQ, RACE, TriviaQA, WinoGrande
Reasoning	MathVista-MINI, MathVision-MINI, MathVerse-MINI, DynaMath, GSM8K, TheoremQA, MATH

This additional selection yields a 33-task suite, which we call our **Core** set. As highlighted in the main body, we use this set to report all major results. We additionally create a 13-task **Validation** set, which we use for faster experimentation, by picking the most established subset of benchmarks per category.

To enumerate benchmarks and illustrate tier organization, we list Extended benchmarks in Tab. 16, Core in Tab. 17, and Validation in Tab. 18.

F.1 Issues with Grounding Benchmarks

We initially considered including grounding and dense recognition benchmarks, such as RefCOCO [126], RefCOCO+ [332], RefCOCOg [220], and related COCO-derived tasks, in the DCVLM evaluation suite. We ultimately excluded them, not because grounding is unimportant, but because the standard grounding benchmarks introduce unusually severe contamination and split ambiguities. These issues make it difficult to interpret them as reliable held-out evaluations in a data-centric benchmark at the scale of DCVLM.

The main issue is that many grounding, VQA, and captioning benchmarks are built on overlapping subsets of MS COCO images [167]. The RefCOCO family uses COCO images, as do COCO Captions [39, 124], OK-VQA [204], VSR [171], and TolokaVQA [287]. As a result, the same underlying image can appear as a training example for one task and as an evaluation example for another, possibly with a different annotation format, e.g., a caption, a VQA pair, or a referring expression. This makes image-level decontamination across the full training pool difficult, especially when downstream datasets rename files, construct custom splits, or mix COCO2014 and COCO2017 images in non-standard ways.

The RefCOCO variants further complicate this picture because they are not independent held-out datasets. RefCOCO and RefCOCO+ were collected through ReferItGame and commonly use the unc split, where train/test separation is performed at the image level [126, 332]. RefCOCOg was

Table 17: List of benchmarks in our 33-task **Core** suite.

Category	Benchmarks
General	MMBench-DEV-EN, MMBench-DEV-CN, MMBench-DEV-EN-V11, MMBench-DEV-CN-V11, SEEDBench-IMG, MMT-Bench-VAL, VMCBench-DEV
Knowledge	MMMU-DEV-VAL, ScienceQA-VAL, AI2D-TEST, AI2D-TEST-NO-MASK
OCR	ChartQA-TEST, OCRBench, InfoVQA-VAL, SEEDBench2-Plus, OCRVQA-TESTCORE, CharXiv-Reasoning-VAL
Vision-Centric	MMT-Bench-VAL-MI, BLINK, MUIRBench, CV-Bench-2D, CV-Bench-3D
Multilingual	MTVQA-TEST, MMMB, MTL-MMBench-DEV
Text	C-Eval, CMMLU, GaoKao, MMLU, NQ, RACE, TriviaQA, WinoGrande

Table 18: List of benchmarks in our 13-task **Validation** suite.

Category	Benchmarks
General	MMBench-DEV-EN, MMBench-DEV-CN, VMCBench-DEV
Knowledge	MMMU-DEV-VAL, AI2D-TEST
OCR	OCRBench, SEEDBench2-Plus
Vision-Centric	BLINK, CV-Bench-2D, CV-Bench-3D
Multilingual	MMMB
Text	C-Eval, MMLU

collected through AMT and commonly uses the Google split, where the split is defined at the object or annotation level [220]. The variants also differ in their linguistic supervision: RefCOCO contains relatively short expressions, RefCOCOg contains longer and more detailed expressions, and RefCOCO+ restricts expressions to appearance-based descriptions by excluding certain spatial phrases such as “person on the right” [220, 332]. Thus, a model trained on one variant may have already seen the same image-object pair that appears in another variant, only paired with a different referring expression.

This concern is amplified by common VLM pretraining practices. In several open training recipes, grounding data is not treated as a single isolated benchmark split, but appears to group annotations from RefCOCO and RefCOCOg under the same image⁴. In such cases, evaluation on a nominally separate RefCOCO-style split may partially measure familiarity with recurring COCO image-object pairs rather than genuine grounding generalization.

For these reasons, we exclude grounding benchmarks from the main DCVLM evaluation suite. Including them would make it difficult to attribute performance differences cleanly to data mixing, filtering and other curation choices, since gains could arise from uncontrolled overlap between COCO-derived training and evaluation data.

⁴https://internvl.readthedocs.io/en/latest/get_started/chat_data_format.html#grounding-detection-data

G Train-Test Decontamination

To ensure that conclusions from our data-centric experiments are not confounded by train-test overlap, we perform strict two-way decontamination of all training sources against the DCVLM-Extended evaluation suite. We apply separate procedures for multimodal and text-only samples.

G.1 Image-Based Decontamination

For the multimodal portion of our pool, we decontaminate at the image level using the self-supervised copy-detection descriptor SSCD model [235] (ResNet-50 backbone). We embed every image in both the training pool and the DCVLM-Extended evaluation suite under an *identical* transform: we resize the shorter edge to 288px while preserving aspect ratio, then apply ImageNet normalization. We deliberately favor fidelity over speed here [20]—the 288px input and the absence of any center crop avoid truncating charts, tables, and documents.

Matching procedure. We build a single exact (brute-force, flat L_2) FAISS index [63] over all evaluation-image descriptors. For each training image we retrieve its top-1 nearest evaluation image and convert the squared- L_2 distance d to a cosine similarity $s = 1 - d/2$ (valid because SSCD descriptors are unit-normalized). A training sample is discarded if *any* of its images attains $s \geq 0.75$ to *any* evaluation image. For multi-image samples (e.g. multimodal documents) we thus aggregate by the maximum similarity over the sample’s images. Because the evaluation index is small, exact search resolves millions of queries per second on a single GPU, so we use no approximate indexing. Our pipeline relies on a distributed implementation that builds on top of exact FAISS [63] search.

Choosing the threshold. We use a similarity threshold of 0.75, deliberately more conservative than the 0.95 adopted by FineVision [310]. Fig. 13 shows representative train/test matches across the SSCD similarity scale, drawn from four pool sources that together span natural photographs (ViQuAE, DataComp-1B), diagrams (AI2D), and mixed document/chart-style imagery (MMDU). Only near-pixel-identical copies score ≥ 0.95 . The 0.75–0.95 range is instead dominated by *genuine* duplicates that have undergone benign transformations—diagrams with labels stripped or re-typeset, re-crops with added text overlays, and re-scans of the same source image. A 0.95 threshold retains all of these, i.e. it admits many false negatives. We found this effect to be systematic for non-natural image domains—charts, tables, documents, and UI screenshots—where genuine train/test near-duplicates receive lower SSCD scores than natural photographs do. Our stricter 0.75 cutoff removes this band at the cost of some false positives: we knowingly discard a modest amount of clean training data in exchange for substantially stronger decontamination guarantees. Below 0.75, matches are visually or semantically similar but distinct images (different photographs of the same landmark, same-layout documents with different content) and are correctly retained. Note that even below the 0.75 threshold, there are indeed some false negatives, however we found 0.75 to be a stable threshold that balanced both precision and recall, enabling us to perform decontamination in a faithful manner while minimizing discard rates.

Comparison to FineVision’s threshold. Fig. 12 quantifies the gap between the two protocols: for each of the four sources of Fig. 13 we plot the fraction of pool images removed as a function of the threshold. The region between 0.75 and 0.95 is overlap that FineVision’s protocol leaves in the pool—for AI2D, the 0.75 threshold removes 16.9% of pool images versus only 10.9% at 0.95, i.e. over a third of the detected overlap survives the lenient threshold.

Removal rates on our DCVLM pool. The fraction of training images flagged varies sharply by source and is concentrated in datasets whose training and evaluation splits are drawn from the same underlying image distribution. At the 0.75 threshold, notable per-source removal rates on our DCVLM-pool large scale include ScienceQA (60.4%), ChartQA (25.6%), AI2D (25.0%), OCR-VQA (2.8%), and InfoVQA (100%, as its training and evaluation images entirely coincide).

G.2 Text-Based Decontamination

For the text-only portion of our pool, we decontaminate at the *prompt/question level*, following the protocol of Tulu-3 [138]: for each training sample we concatenate its human turns into a single query string and match it against the prompts of every text-only evaluation. We decontaminate against all text-only benchmarks in the InternVL-3 evaluation pool (16 benchmarks, namely MMLU, CMMLU, C-Eval, GaoKao, TriviaQA, NaturalQuestions, RACE, WinoGrande, HellaSwag, BBH,

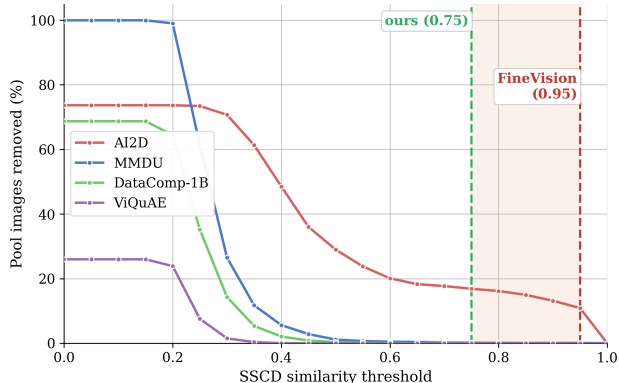


Figure 12: **Image-decontamination removal rate vs. SSCD similarity threshold.** Fraction of pool images removed as a function of the similarity threshold for four sources. The shaded band between our threshold (0.75, green) and FineVision’s (0.95, red) is detected train/test overlap that FineVision’s protocol leaves in the training pool.

GSM8K, MATH, HumanEval, MBPP, MBPP-CN, TheoremQA). We use a two-stage MinHash + exact-substring pipeline, detailed below. Figs. 14 and 15 justify our hyperparameter choices, and Tab. 19 shows representative detected overlaps.

Matching procedure. We tokenize text with the Tulu-3 regular-expression tokenizer (lowercasing and splitting on whitespace and punctuation) and represent each sample by the set of its *word-level* 5-grams.⁵ We summarize each set with a 128-permutation MinHash signature [25] and estimate the Jaccard similarity between a training sample and its nearest evaluation sample by *exact* (brute-force) signature comparison. Because the evaluation index is small ($\sim 86k$ samples), exact top-1 search is fast enough that we do not require LSH-based approximate matching.

Two-stage filtering. In the first stage, we discard every training sample whose top-1 Jaccard similarity to any evaluation prompt exceeds 0.55. In the second stage, training samples falling in the ambiguous band ($0.3 \leq \text{Jaccard} < 0.55$) are subjected to a *bi-directional* exact-substring check: a sample is discarded if either the training query is contained verbatim in the matched evaluation prompt or vice versa. The second stage recovers true positives whose Jaccard score is diluted below 0.55—for example, a short evaluation question embedded inside a longer training prompt. This combination balances recall (fuzzy near-duplicates caught by MinHash) with precision (the substring check filters out spurious matches in the intermediate band). Training samples too short to form a single 5-gram are always retained.

Choosing the hyperparameters. We sweep the three free parameters of the MinHash stage and report per-dataset match rates over 10 instruction datasets in Fig. 14. (i) *Signature size*: increasing the number of permutations from 128 to 1024 leaves match rates essentially unchanged (Fig. 14a), so we use 128 for faster processing. (ii) *n-gram length*: 5-grams detect overlaps at meaningful rates, whereas 8- and 13-grams detect almost nothing (Fig. 14c), because many evaluation prompts are shorter than 8 (let alone 13) tokens and therefore never form a matching *n*-gram. We therefore use 5-grams for higher recall. (iii) *Threshold*: across all datasets the match rate falls steeply with the similarity threshold and is already below $\sim 0.1\%$ by 0.5 (Fig. 14b). At our operating point of 0.55 fewer than 0.02% of text samples are removed, so a conservative threshold costs almost no data.

Selecting the threshold via human annotation. The match-rate sweep does not by itself tell us *which* threshold separates genuine contamination from coincidental overlap. To answer this quantitatively, we ran a small human-annotation study: we drew candidate matches spanning all similarity bins from the 10 datasets above, and seven of the authors independently labeled each candidate as a true overlap (the two prompts ask the same question) or a false positive. Aggregating the true-positive (TPR) and false-positive (FPR) rates per similarity bin (Fig. 15), we find that below ~ 0.55 the FPR is high (most matches are spurious, e.g. questions that share a template but have different answers), while the

⁵This tokenizer is tuned for whitespace-delimited languages and does not segment Chinese well; we treat the resulting under-matching of Chinese text as a known limitation.

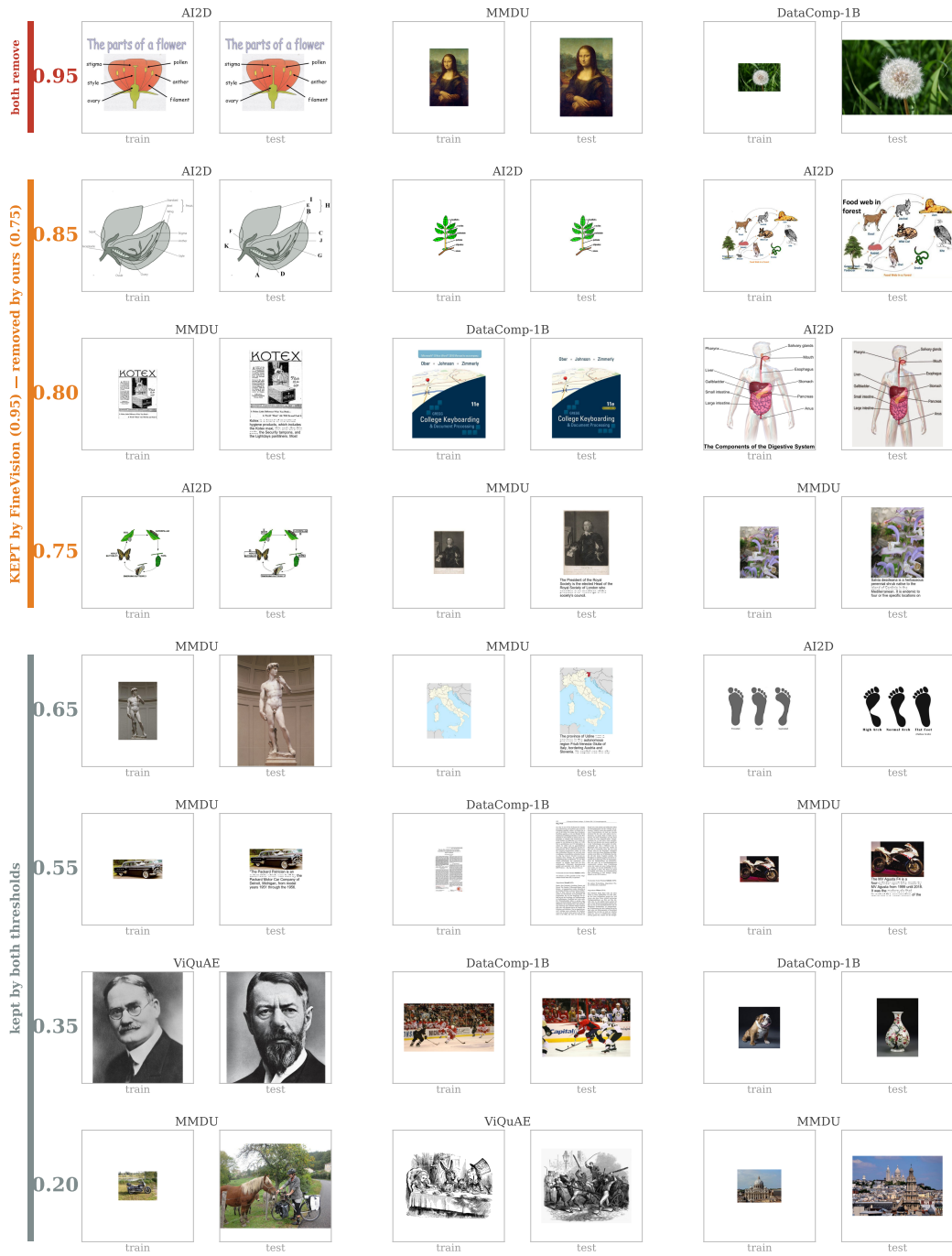
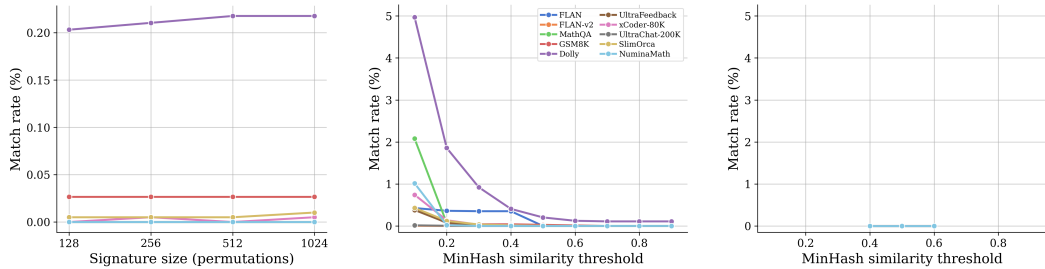


Figure 13: **Qualitative train/test image matches by SSCD similarity band.** Each pair shows a training-pool image (left) and its top-1 match in the DCVLM-Extended evaluation suite (right). Rows are grouped by the maximum SSCD cosine similarity of the pair, and labels above each pair give the pool source. Only near-pixel-identical copies score ≥ 0.95 (top row, removed by both protocols). The 0.75–0.95 band (orange) consists almost entirely of genuine duplicates under benign transformations—label-stripped or re-typeset diagrams, re-crops with text overlays, re-scans—which FineVision’s 0.95 threshold *keeps* but our 0.75 threshold removes. Below 0.75 (gray), matches are similar but genuinely distinct images (different photographs of the same statue, different maps of the same country, same-layout documents) and are retained by both protocols.



(a) signature size has no effect. (b) match rate vs. threshold. (c) 13-grams detect almost nothing.

Figure 14: **Text-decontamination hyperparameter sweep.** Per-dataset match rate (% of training queries flagged) across the three MinHash design choices. Signature size is immaterial (a), so we use 128 permutations. Longer n -grams miss short evaluation prompts and detect almost no overlap (c), so we use 5-grams (b). Match rates are small in absolute terms and fall steeply with the threshold.

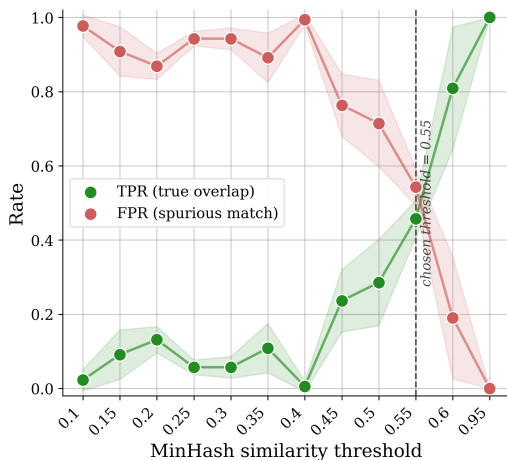


Figure 15: **Threshold selection via human annotation.** True-positive (TPR, green) and false-positive (FPR, red) rates of MinHash matches at each similarity bin, aggregated over seven annotators. Below ~ 0.55 most matches are spurious (high FPR). The TPR climbs steeply just above it and the curves cross around 0.55–0.6, motivating our first-stage threshold of 0.55.

TPR rises steeply just above it and the two curves cross around 0.55–0.6. We therefore adopt 0.55 as the first-stage threshold, and rely on the second-stage substring check to recover the genuine matches that fall just below it.

G.3 Overall removal rates

Fig. 16 reports per-dataset removal rates of the full decontamination protocol (image- and text-based combined) over our medium DCVLM-pool as a representative. Removal is concentrated in datasets whose training and evaluation splits are drawn from the same underlying distribution—InfoVQA loses 100% of its samples (its training and evaluation images almost entirely coincide), ScienceQA 66.4%, TabMWP 63.4%, FigureQA 56.6%, ChartQA 41.7%, and AI2D 21.9%. At the pool level, however, decontamination is cheap—only 0.29% of all training samples are removed. Strict decontamination therefore costs almost no data while removing precisely the samples most likely to inflate benchmark scores.

Table 19: **Representative text overlaps detected by our pipeline.** We present some training/evaluation prompt pairs with their estimated Jaccard similarity. High-similarity pairs are genuine near-duplicates and are removed (Stage 1). Pairs just below the 0.55 threshold typically share a template but have different answers and are correctly retained.

Jaccard	Training prompt	Evaluation prompt	Decision
1.00	What is the loudest animal on Earth?	What is the loudest animal on earth?	Removed
1.00	Who was the first man to walk on the moon?	Who was the first man to walk on the Moon?	Removed
1.00	What is the capital city of Croatia?	What is the capital city of Croatia?	Removed
0.65	in which city was the first public opera house opened	Opened in 1637, in which city was the first public opera house?	Removed
0.62	According to Greek mythology, who was the first woman on earth?	In Greek mythology, who was the first woman on Earth?	Removed
0.59	What is the largest state in the US?	Which is the largest state in the US?	Removed
0.55	What is the tallest building in Seattle?	What is the tallest building in London?	Kept
0.51	who is considered the father of computers?	who is considered the father of modern cosmology	Kept
0.48	Write a Python function to find the maximum of three given numbers.	Write a Python function to find the maximum of two numbers.	Kept
0.48	What is the largest lake in Switzerland?	What is the largest lake in Central America?	Kept
0.45	What is the capital city of Malaysia?	What is the capital city of Colombia?	Kept
0.45	Who is the founder of SpaceX?	Who is the founder of Sikhism?	Kept
0.30	Caroline of Ansbach was the wife of which British monarch?	Mrs Maria Fitzherbert was the wife of which British monarch?	Kept
0.30	Nancy earns \$28 for working 4 hours. How many hours does she have to work to earn \$70?	Mai earns \$5.50 per hour at her after-school job. How many hours does she have to work to earn \$132?	Kept
0.28	What are the types of RVs?	What are the types of scanning?	Kept
0.28	What is the best way to make Indian ginger tea?	Which is the best way to make friends with an American?	Kept

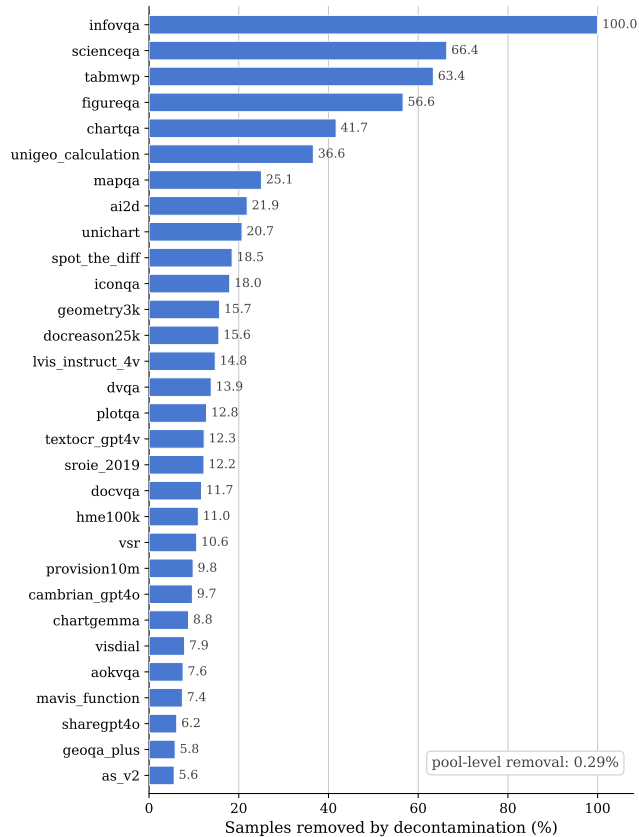


Figure 16: **Per-dataset removal rates of the full decontamination protocol.** The 30 pool sources with the highest fraction of samples removed by the combined image-based (SSCD $s \geq 0.75$) and text-based (MinHash Jaccard ≥ 0.55 + substring check) decontamination against the DCVLM-Extended evaluation suite. Removal concentrates in sources whose training and evaluation splits share an underlying distribution (InfoVQA, ScienceQA, TabMWP). At the pool level only 0.29% of samples are removed.

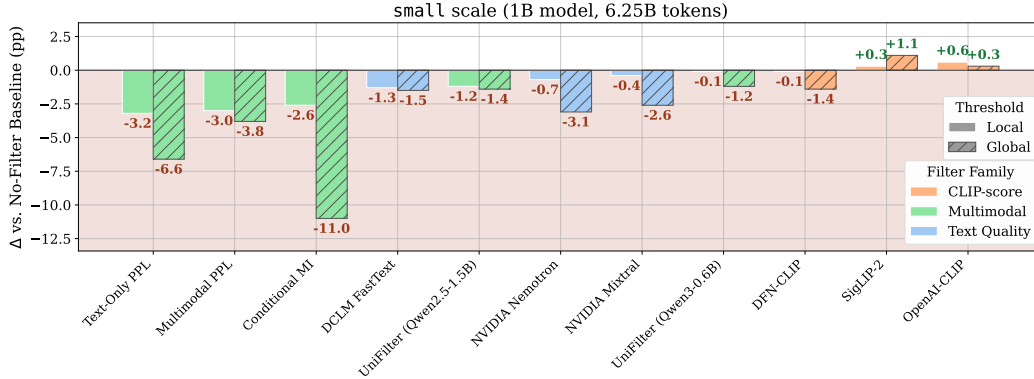


Figure 17: **Filtering rarely helps, but changing the data composition does move performance substantially (cont).** Complementary experiments at the small scale of our benchmark confirm the outcome of Sec. 4.1: established quality filtering rarely provides significant gains over a no-filter baseline, yet variations between global and local filtering remain frequent.

H Additional Experiments and Details

H.1 Filtering rarely helps

Figure 17 reports the filtering results at the small scale (1B model, 6.25B tokens), complementing the medium-scale results in Section 4.1. The conclusions are fully consistent: local filtering uniformly fails to improve over the no-filter baseline. The only positive outcomes still arise from global CLIP-score filtering (SigLIP-2 +1.1pp and OpenAI-CLIP +0.6pp), in line with Fig. 3, yet, once again, these gains are lower than what one would expect from prior work.

Consistently, global filtering and local filtering produce different results also at this scale. Overall, these results reinforce the central finding of Sec. 4.1 that filtering is not a reliable curation strategy, and lead to hypotheses that observed performance variations might be primarily driven by the underlying mixture changes.

Additional filtering experiments. For completeness, we report the full benchmarking we conducted with a variety of additional configurations for the filters in Sec. 4.1. Specifically, we tested several local filtering variants of the common filters presented in Sec. 4.1:

- for CLIP-score, we experiment with filtering not only image-caption pairs alone, but also multimodal documents and instruction tuning data. For the former, we score a document according to the average CLIP-score of all its adjacent image-text snippets. For the latter, we employ image-to-question similarity. In case of multi-image examples, we use an identical strategy to multimodal documents and score the closest image to the question in the conversation history. For multi-turn samples, we average across all adjacent pairs as we do for multimodal documents. We use the same three models described in Sec. 4.1
- for text quality classifiers, we additionally experiment with filtering only the language portion of our pool. This experiment is motivated by the hypothesis that text-only sources might be more in-distribution with respect to the classifiers themselves. Identically to CLIP-score, we report all three classifiers of the main body.
- For Text-Only Perplexity, Multimodal Perplexity, and Conditional Mutual Information filters we experiment with applying them to all multimodal data types jointly (image-caption pairs, instruction-tuning data, multimodal documents), as well as subset combinations.

We additionally experimented with basic filtering strategies:

- Length heuristics, that remove the shortest and the longest samples (removes bottom-10% shortest and top-90% longest at both scales). We experiment with several interpretations of “length”: (i) the count of all tokens, (ii) the count of answer tokens, on top of which the loss is ultimately computed during training; and (iii) the count of language tokens (i.e., question + answer tokens);

Table 20: `small` scale results for our local filtering experiments (for faster experimentation, we use our 13-task **Validation** set). We report within parentheses the data type to which the filter has been applied.

Filter	Gen	Know	OCR	Vision	MTL	Text	Val Avg
<code>small scale</code>							
<i>No Filter</i>	45.6	38.5	43.1	44.4	46.6	35.9	42.4
Conditional MI (all)	40.3	37.5	46.0	41.2	46.9	28.8	39.7
Conditional MI (mmdoc)	44.0	39.0	41.5	43.5	46.1	36.6	41.8
Conditional MI (im-cap+mmdoc+inst)	42.0	37.9	47.0	41.4	45.7	36.1	41.4
DCLM FastText (all)	43.9	36.7	42.8	41.4	45.7	36.0	41.0
DCLM FastText (text-only)	46.8	37.6	42.3	42.9	46.8	36.3	42.2
DFN-CLIP (im-cap)	46.8	39.0	41.2	42.7	48.5	37.1	42.4
DFN-CLIP (im-cap+mmdoc)	46.2	39.1	39.3	44.5	46.5	33.5	41.7
DFN-CLIP (im-cap+mmdoc+inst)	45.8	39.0	39.3	41.0	45.6	35.0	41.0
IQA, Kadid-10k (im-cap+mmdoc+inst)	44.4	37.4	42.1	41.8	46.9	36.4	41.3
IQA, TID 2013 (im-cap+mmdoc+inst)	44.9	37.6	40.4	42.8	47.0	36.6	41.5
Length (all tokens)	45.5	38.8	40.8	42.8	46.5	36.1	41.8
Length (text-only tokens)	41.8	38.2	41.9	41.1	45.7	34.2	40.2
Length Trainable	44.7	38.2	40.6	41.2	46.8	35.9	41.1
Multimodal PPL (all)	43.4	36.5	36.5	41.3	46.7	36.5	40.0
Multimodal PPL (mmdoc)	46.2	39.1	43.8	42.7	45.7	35.8	42.3
Multimodal PPL (im-cap+mmdoc+inst)	40.4	35.3	37.4	41.6	44.9	35.6	39.0
NVIDIA Mixtral (all)	46.4	38.7	40.6	44.0	47.6	37.6	42.5
NVIDIA Mixtral (text-only)	48.6	38.9	42.2	43.7	48.1	37.4	43.2
NVIDIA Nemotron (all)	46.1	38.2	41.8	42.7	47.8	37.7	42.3
NVIDIA Nemotron (text-only)	46.5	37.6	43.0	43.7	47.3	37.1	42.6
OpenAI-CLIP (im-cap)	46.7	38.9	45.1	44.4	46.8	36.7	43.2
OpenAI-CLIP (im-cap+mmdoc)	46.6	38.3	44.3	45.2	46.0	36.1	43.0
OpenAI-CLIP (im-cap+mmdoc+inst)	46.5	37.0	42.6	41.2	45.1	35.3	41.4
SigLIP-2 (im-cap)	45.9	38.1	47.1	42.9	46.4	35.7	42.7
SigLIP-2 (im-cap+mmdoc)	45.8	38.7	46.3	43.6	46.8	35.4	42.8
SigLIP-2 (im-cap+mmdoc+inst)	46.0	37.8	46.8	41.3	45.5	35.7	42.1
Text-Only PPL (all)	42.9	37.9	31.9	42.0	46.4	36.1	39.4
Text-Only PPL (mmdoc)	45.3	38.6	41.4	42.8	46.1	36.0	41.7
Text-Only PPL (im-cap+mmdoc+inst)	40.4	35.7	34.0	42.6	43.6	33.8	38.5
UniFilter (Qwen2.5-1.5B)	43.7	38.8	40.2	41.6	46.5	36.5	41.1
UniFilter (Qwen3-0.6B)	47.7	40.1	40.2	41.3	47.6	35.9	42.1

- Image Quality Assessment (IQA) models, to remove low-quality images. We test with the ARNIQA [5] suite of image quality models. Specifically, we use the regressors trained on the Kadid-10k [166] and Tid-2013 [236] datasets, since they are one of the highest performing image quality assessment classifiers on standard IQA benchmarks.

Except for length heuristics, which use the same (bottom-10%, top-90%) percentiles on both scales, all experiments are conducted with an aggressive top-10% filtering threshold on the small scale and a more lenient top-40% threshold on the medium scale, following Sec. 4.1.

Given the large number of experiments, we opted for faster iterations on our 13-task Validation suite. As anticipated in the main body, no configuration provides significant gains.

For completeness, we report Validation results for global filtering in Tab. 22 as well (we did not test all variants here, but rather only those presented in Fig. 3).

H.2 Data formatting

We use the Chat Markup Language (ChatML) format [229] for training, which assigns a role (“system”, “user”, or “assistant”) to all messages in a conversation. ChatML is a structured format explicitly separating conversation roles (e.g., system, user, assistant) and demarcating message boundaries using specific tokens (e.g., `<|im_start|>` and `<|im_end|>`). Text-only and instruction-tuning data typically come in this format already, while image-caption pairs can be easily formatted

Table 21: medium scale results for our local filtering experiments (for faster experimentation, we use our 13-eval **Validation** set). We report in parentheses the data type to which the filter was applied.

Filter	Gen	Know	OCR	Vision	MTL	Text	Val Avg
medium scale							
<i>No Filter</i>	60.5	50.8	56.9	47.4	60.4	47.3	53.4
Conditional MI (all)	55.7	48.4	57.9	44.7	59.1	39.4	50.1
Conditional MI (mmdoc)	60.1	50.8	57.4	48.9	61.0	47.7	53.8
Conditional MI (im-cap+mmdoc+inst)	59.1	50.0	56.9	45.0	60.4	47.0	52.3
DCLM FastText (all)	54.5	48.1	55.8	43.3	55.3	47.3	50.1
DCLM FastText (text-only)	59.9	50.7	57.3	48.9	61.6	47.4	53.8
DFN-CLIP (im-cap)	59.6	50.5	55.7	49.5	61.5	47.3	53.5
DFN-CLIP (im-cap+mmdoc)	60.0	49.5	56.1	48.4	60.7	47.2	53.2
DFN-CLIP (im-cap+mmdoc+inst)	59.2	50.6	55.2	45.6	60.3	47.7	52.4
IQA, Kadid-10k (im-cap+mmdoc+inst)	60.0	51.1	57.2	45.2	60.0	47.5	52.9
IQA, TID 2013 (im-cap+mmdoc+inst)	59.4	50.2	56.9	47.1	60.6	47.3	53.0
Length, all tokens	58.5	50.2	56.6	46.6	60.4	47.1	52.6
Length, language tokens	55.3	46.8	56.8	45.7	56.0	46.5	50.7
Length, answer tokens	60.1	51.3	57.1	44.5	60.4	47.4	52.8
Multimodal PPL (all)	59.9	50.3	58.8	45.3	61.0	46.9	53.0
Multimodal PPL (mmdoc)	59.6	51.3	56.8	47.6	61.7	47.7	53.5
Multimodal PPL (im-cap+mmdoc+inst)	59.2	50.0	59.9	46.8	58.3	47.0	53.1
NVIDIA Mixtral (all)	60.2	50.1	57.0	47.5	60.1	47.5	53.3
NVIDIA Mixtral (text-only)	61.0	51.0	57.0	48.6	62.6	47.9	54.1
NVIDIA Nemotron (text-only)	61.8	49.9	57.2	46.8	62.0	47.5	53.6
OpenAI-CLIP (im-cap)	59.7	50.4	57.8	47.2	59.9	47.5	53.2
OpenAI-CLIP (im-cap+mmdoc)	60.4	50.1	58.0	47.4	61.6	47.3	53.5
OpenAI-CLIP (im-cap+mmdoc+inst)	55.9	48.5	57.4	44.8	59.1	47.2	51.3
SigLIP-2 (im-cap)	60.2	50.3	57.7	48.7	60.8	47.4	53.7
SigLIP-2 (im-cap+mmdoc)	60.1	50.8	59.2	47.8	61.6	47.5	53.9
SigLIP-2 (im-cap+mmdoc+inst)	58.0	48.9	57.8	43.7	59.6	47.3	51.7
Text-Only PPL (all)	59.5	50.1	57.7	46.0	60.1	47.2	52.8
Text-Only PPL (mmdoc)	60.2	50.3	57.7	48.9	60.6	47.4	53.8
Text-Only PPL (im-cap+mmdoc+inst)	59.3	51.1	57.5	47.1	59.0	47.2	53.1
UniFilter (Qwen2.5-1.5B)	58.2	49.6	55.8	43.8	59.9	47.2	51.6
UniFilter (Qwen3-0.6B)	59.5	50.1	57.1	46.9	61.1	46.8	52.9

using templates such as “*Describe the image*” to serve as user messages. However, there is no consensus on how to format multimodal documents, with prior works exploring contrasting methods [14, 140, 161, 366].

We thus conduct an experiment at the small scale to compare two strategies: **in-context**, which uses the last text chunk as assistant response, and concatenates everything except the last text chunk into the user message vs **multi-turn**, which interleaves images and text chunks into multiple turns, so that images are user messages and all text chunks are assistant responses. To ensure reliable signal, we use a uniform mixture across data types (25% for all data types). From Tab. 23, we find multi-turn formatting superior to in-context formatting. We thus adopt multi-turn formatting for all our experiments.

Table 23: Multi-turn is better than in-context formatting.

Format	Val Avg
In-context	35.5
Multi-turn	37.9

H.3 Diminishing returns of downstream filtering

This section provides further details on the upstream vs. downstream filtering experiment in Sec. 4.1.

We start from the base DCVLM small-pool of 160 source datasets. By virtue of our collection procedure (Sec. 3.1), this pool has already been “upstream-filtered” by the source dataset providers, so we treat it as the “Upstream-filtered=100%” setting. It follows our pool’s base mixture ratio: 75% image-caption pairs, 18% text-only, 4% multimodal documents, and 3% multimodal instruction-tuning data. The datacomp_1b subset alone accounts for roughly 47% of the image-caption portion.

Table 22: Global filtering results (validation set).

Filter	Gen	Know	OCR	Vision	MTL	Text	Val Avg
small scale							
<i>No Filter</i>	45.6	38.5	43.1	44.4	46.6	35.9	42.4
Conditional MI	25.2	30.2	39.1	41.2	41.2	22.1	32.5
DCLM FastText	40.6	36.9	41.8	41.6	45.9	36.8	40.3
DFN-CLIP	43.1	40.5	34.4	42.3	48.0	36.7	40.6
Multimodal PPL	38.2	38.7	40.9	40.9	46.1	36.1	39.6
NVIDIA Mixtral	38.0	37.9	36.7	41.9	47.4	37.2	39.3
NVIDIA Nemotron	35.8	36.8	34.5	42.8	46.3	37.4	38.4
OpenAI-CLIP	45.3	37.6	44.4	43.9	48.3	36.4	42.5
SigLIP-2	46.0	41.3	44.3	44.2	48.5	37.2	43.4
Text-Only PPL	33.3	37.8	22.3	43.6	45.8	37.1	36.2
UniFilter (Qwen2.5-1.5B)	43.9	36.5	37.0	44.2	46.2	37.5	41.0
UniFilter (Qwen3-0.6B)	45.4	38.5	35.7	42.1	46.8	37.8	41.0
medium scale							
<i>No Filter</i>	60.5	50.8	56.9	47.4	60.4	47.3	53.4
Conditional MI	55.9	46.9	56.0	44.4	54.9	35.3	48.6
DCLM FastText	59.5	51.1	58.7	48.0	62.2	47.5	53.8
DFN-CLIP	59.9	49.9	56.3	49.7	63.2	46.9	53.7
Multimodal PPL	60.4	50.6	59.6	45.2	60.3	47.4	53.2
NVIDIA Mixtral	61.1	49.4	59.1	48.3	61.6	47.4	54.0
NVIDIA Nemotron	60.4	50.4	57.0	45.8	61.8	47.5	53.1
OpenAI-CLIP	59.7	52.6	58.5	47.4	63.2	47.5	54.0
SigLIP-2	60.6	52.2	60.1	48.2	62.9	47.0	54.5
Text-Only PPL	58.2	51.4	57.8	45.3	61.1	47.2	52.7
UniFilter (Qwen2.5-1.5B)	58.8	51.1	58.1	44.2	60.3	47.4	52.5
UniFilter (Qwen3-0.6B)	56.5	48.4	54.3	43.8	56.9	47.0	50.6

To construct the other two settings, we progressively replace pre-filtered image-caption data with raw, unfiltered pairs crawled directly from the released [CommonPool](#) URLs:

- **Upstream-filtered=65%**: replace pre-filtered datacomp_1b with its unfiltered counterpart, yielding $0.47 \times 75 \approx 35\%$ unfiltered data.
- **Upstream-filtered=25%**: replace the *entire* image-caption portion with raw unfiltered datacomp_1b, yielding 75% unfiltered data.

For each setting, we apply CLIP-score downstream filtering on image-caption pairs using OpenAI-CLIP-ViT-L/14 [241], and train small-scale models on the resulting datasets.

The gain from downstream filtering shrinks monotonically with the level of upstream curation: +2.4pp at Upstream-filtered=25%, +1.3pp at 65%, and only +0.6pp at 100%. The first result confirms that filtering *does* help on genuinely noisy data, but the trend establishes that *additional filtering on top of already-curated data operates in a regime of diminishing returns*.

H.4 Temperature-scaled sampling

Several of our most task-relevant datasets are already present in the pretraining pool: the training splits of ScienceQA, AI2D, ChartQA, InfoVQA and OCR-VQA all live in our instruction-tuning sources. It is therefore natural to ask why the corresponding benchmarks do not benefit more, and a simple hypothesis presents itself—these datasets may be *severely undersampled* during training. By default we sample sources *within* each data type in proportion to their size, $p(d) \propto \text{len}(d)$ (where $\text{len}(d)$ is the number of samples d contributes to the pool), so a handful of large web-scale sources can dominate the mixture and drown out small but high-quality or diverse datasets.

Prior work tackles this imbalance in two ways: by hand-setting mixture ratios based on intuition (e.g. the Nemotron [59] and OLMo-2 [227] series, and FineWeb [233]), or by principled downweighting of large sources—most notably the square-root sampling used in the Molmo line of work [45, 57].

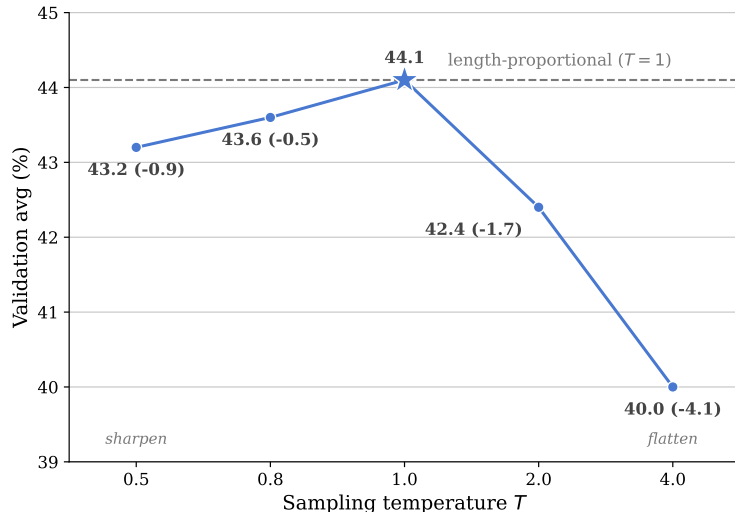


Figure 18: **Length-proportional sampling ($T = 1$) is near-optimal.** Validation average as a function of sampling temperature T in $p(d) \propto \text{len}(d)^{1/T}$. Both sharpening ($T < 1$) and flattening ($T > 1$) degrade performance relative, with the near-uniform $T = 4$ setting losing 4.1pp.

We study this finer, within-type sampling in a principled manner via temperature-scaled sampling,

$$p(d) \propto \text{len}(d)^{1/T}, \quad (1)$$

where $T = 1$ recovers length-proportional sampling, $T = 2$ corresponds exactly to the square-root sampling of Molmo, and $T \rightarrow \infty$ approaches uniform sampling over datasets (*flattening*, which upweights small sources). We additionally explore $T < 1$, which *sharpens* the distribution toward the largest sources. We apply the sweep $T \in \{0.5, 0.8, 1.0, 2.0, 4.0\}$ to our balanced mixture (40% image-caption, 5% multimodal documents, 15% text, 40% instruction-tuning, see Sec. 4.2), holding all other settings fixed, and report results on our 13-task Validation suite. We run these experiments at the small scale of our benchmark.

As shown in Fig. 18, the undersampling hypothesis does not hold. The default $T = 1$ is already near-optimal, and flattening the distribution—which upweights exactly the small, task-relevant datasets we suspected of being undersampled—consistently *hurts*: square-root sampling ($T = 2$) drops to 42.4% (−1.7pp) and the near-uniform $T = 4$ falls to 40.0% (−4.1pp), below even the FINEVISION baseline. Sharpening offers no benefit either ($T = 0.8$: 43.6%, −0.5pp; $T = 0.5$: 43.2%, −0.9pp). We hypothesize the contradicting results between ours and the Molmo series are due to the differing mixture ratios and pool composition—we believe there might be some non-trivial effects between the broader data-type mixing ratios and the within-data-type source sampling, which would be an interesting exploration for future work. Given our results, we hence retain length-proportional ($T = 1$) sampling throughout—noting that the cross-type mixing ratios of Sec. 4.2 are a far more impactful lever than the within-type temperature.

H.5 Synthetic recaptioning

Replacing noisy web-crawled alt-text with synthetic captions is a well-established lever for improving contrastive (CLIP-style) pretraining [79, 163, 334]. We test whether the same holds for generative VLM pretraining. We hold the images and their training order fixed and vary *only* the caption text on the DataComp [73] subset of our image-caption pool, comparing three variants:

- **Alt-text (original):** the raw web alt-text from the original DataComp-1B image-caption pairs.
- **Synthetic short:** concise captions generated by Qwen2-VL-7B [297], conditioned on the image, its alt-text, and a set of detected concepts [79].
- **Synthetic spatial:** the short captions further augmented with explicit 2D/3D spatial-relationship descriptions, to probe whether richer, layout-aware captions help the more vision-centric benchmarks.

Table 24: **Recaptioning image-caption pairs does not improve VLM pretraining.** Validation scores (%) per category for the three caption variants. Synthetic captions are generated by Qwen2-VL-7B [297]. Δ is the change in average relative to the original alt-text baseline.

Captions	General	Knowledge	OCR	Vision	MTL	Text	Val Avg
Alt-text (original)	48.0	40.0	46.8	43.4	49.5	36.6	44.1
Synthetic short	46.6	39.2	44.9	42.5	49.8	36.5	43.2
Synthetic spatial	48.7	39.7	45.9	42.5	50.7	36.7	44.0

We source all caption variants using the released datasets of Ghosh et al. [79]. We run these experiments on a mixture of 65% image-caption, 5% multimodal documents, 15% text, and 15% instruction-tuning data, at the small scale of our benchmark. Crucially, the caption intervention is applied *only* to DataComp-1B, not to the entire image-caption portion: since DataComp-1B accounts for roughly 47% of the image-caption data, the captions are rewritten for only $\approx 47\% \times 65\% \approx 30\%$ of the *overall* training mixture, while every other source (including image-caption datasets that already come with synthetic captions) retains its original text. The images, mixture ratios, and data order are identical across the three runs, so any difference is attributable solely to DataComp-1B captions. As with other experiments, we report on the 13-task Validation suite.

Table 24 shows that **neither synthetic captioning scheme provides a meaningful improvement**: original alt-text reaches 44.1% average, synthetic-short 43.2% (-0.9pp), and synthetic-spatial 44.0% (-0.1pp). Per-category, the changes are small and inconsistent in sign—spatial captions help General (+0.7) and MTL (+1.2) but hurt OCR (-0.9) and Vision (-0.9), netting essentially zero overall. This contrasts with the consistent gains recaptioning brings to CLIP training, and we hypothesize that the difference is in our pretraining setup: the instruction-tuning portion of the mixture already supplies abundant high-quality, densely-described image-text pairs, so perhaps the marginal value of rewriting the image-caption captions is low. In other words, at the VLM pretraining stage the *text content* of image-caption pairs matters far less than their *proportion* in the overall mixture (Sec. 4.2)—given the substantial cost of running a 7B captioner over the full pool, we use the original alt-text throughout.

We caveat that this null result is specific to our setup, and a different configuration could plausibly surface gains. Two factors likely dampen the effect here. First, the intervention touches only $\approx 30\%$ of the training mixture (the DataComp subset)—recaptioning a larger fraction of the pool, or starting from a pool that is entirely alt-text, may tell a different story. Second, and more importantly, our image-caption pool *already* contains several sources beyond DataComp-1B and ReLAION that come with high-quality synthetic or re-written captions. In their presence, rewriting the captions of one additional subset has little marginal effect—much as additional filtering yields diminishing returns once a pool is already well-curated (Sec. H.3). The benefit of recaptioning may thus be largely realized already, and washed out, by the synthetic captions present elsewhere in the pool.

H.6 Online Filtering

Prior data curation benchmarks such as DataComp [73] and DCLM [158] employ *offline resharding*: given a raw pool and a set of metadata annotations, the entire dataset is filtered according to a predicate and written to disk as a new, smaller dataset, which is then used for training. While this makes training efficient—the dataloader never encounters rejected samples—it incurs substantial storage costs, particularly when sweeping over filtering configurations. For example, determining the optimal CLIP-score threshold requires training runs at several percentiles, each of which produces a separate copy of the filtered pool on disk. With dozens of filters and multiple threshold values, storage requirements can grow by an order of magnitude.

Both DataComp and DCLM cite the computational cost of *online filtering*—applying predicate conditions during training—as the reason for preferring offline resharding. We revisit this assumption and find that, in our setting, online filtering matches the throughput of unfiltered training, effectively eliminating the storage bottleneck. We describe our implementation in detail below.

H.6.1 Metadata Annotation

Each sample in our pool is annotated with scores from every filter we consider. Concretely, for a given filter f (e.g., `oai-clip-vit-b-32`), we run a batch inference job over the entire pool that

computes a scalar quality score $s_f(x) \in \mathbb{R}$ for each sample x . These scores are stored as sidecar metadata files alongside the original WebDataset shards: for each shard `xxxxx.tar`, we produce a corresponding `xxxxx.json` file containing a dictionary mapping sample keys to their scores across all filters. A single sample’s metadata entry looks like:

```
"sample_00042_017": {
  ...
  "clip_score_clip_vitl14_224_standard": 31.984,
  "clip_score_siglip2_b16_224_standard": 5.421,
  "text_quality_dclmbaseline_fasttext_score": 0.091,
  "text_quality_nvidia_edu_mixtral_score": 0.122,
  "iqa_kadid10k": 0.574
  ...
}
```

This annotation step is performed once per pool and is fully parallelizable across shards. Adding a new filter (e.g., a new CLIP model variant) requires scoring the entire pool only with the new filter; existing annotations are preserved and do not need to be recomputed. In total, our small pool ships with over 100 precomputed metadata annotations per sample.

H.6.2 Threshold Pre-Computation

Given a filter f and a desired retention rate p (e.g., “keep the top 60%”), we need a threshold $\tau_{f,d}$ for each sub-dataset d (in the local filtering setting) or a single global threshold τ_f (in the global filtering setting). We pre-compute these thresholds for all filters at percentile increments of 5 (i.e., $p \in \{0.1, 0.15, 0.2, 0.25, \dots, 0.9, 0.95\}$) using `torch.quantile()` with linear interpolation over the full score distribution of each sub-dataset (for local filtering) or the full merged score distribution across all sub-datasets (for global filtering). The output is a lightweight JSON lookup table that maps each filter to the score value corresponding to each percentile, which is then used as the filtering threshold:

```
filter_name: {
  "0.10": 0.052,    # 10th percentile
  "0.15": 0.095,    # 15th percentile
  "0.20": 0.118,    # 20th percentile
  ...
  "0.90": 0.891     # 90th percentile
  "0.95": 0.945     # 95th percentile
}
```

H.6.3 Runtime Integration

During training, the dataloader reads samples from WebDataset shards as usual, but before a sample is yielded to the training loop, it passes through a lightweight predicate function registered via `wds.select()`. The predicate performs the following steps for each sample:

1. Look up the sample’s pre-computed filter scores from the sidecar metadata (loaded into a hash map at initialization).
2. For each active filter, compare the sample’s score against the corresponding threshold for that particular sub-dataset.
3. If the score falls below the threshold (i.e., the sample is in the rejected percentile), return `False`; the sample is silently skipped and the dataloader advances to the next sample in the shard.
4. If all active filters pass, return `True`; the sample proceeds to tokenization, packing, and batching as normal.

For multi-filter configurations (e.g., applying both a CLIP-score filter and a text-quality filter simultaneously), the predicates are composed with logical AND: a sample must pass *all* active filters to be retained. Since the predicate evaluation involves only a dictionary lookup and a scalar comparison per filter, the per-sample cost is on the order of microseconds—negligible compared to the millisecond-scale cost of image decoding, tokenization, and GPU transfer.

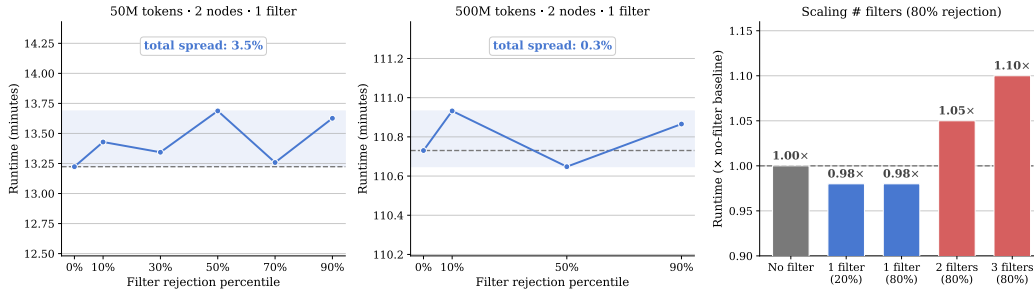


Figure 19: **Online filtering adds negligible overhead to training.** **Left:** Runtime for 50M tokens (2 nodes, 8 GPUs, 1 filter) across rejection percentiles—total spread is 3.5%, within normal run-to-run variance. **Center:** Same measurement at 500M tokens, where the spread shrinks to 0.3%, confirming that any per-sample overhead is amortized over longer runs. **Right:** Scaling the number of simultaneous filters at 80% rejection each. Even with 3 filters applied concurrently, the overhead is only 1.10× baseline, confirming that online filtering is a scalable approach.

Switching between filtering configurations requires changing only the filter names and percentile values in a configuration file; no data needs to be resharded or copied. This enables rapid iteration: practitioners can launch dozens of filtering experiments against the same on-disk pool, varying filters, thresholds, and combinations, without any storage overhead beyond the original pool and its sidecar metadata.

H.6.4 Overhead Measurement

A natural concern is that high rejection rates will cause the dataloader to “spin”—reading and discarding many samples before finding one that passes the filter, thereby starving the GPUs. We benchmark this directly on 2 nodes (8 A100 GPUs), with per-device batch size 4 and effective sequence length 8192, across two representative datasets with different characteristics: ShareGPT4V (high-quality synthetic captions, relatively uniform scores) and MINT-HTML (web-crawled multimodal documents, highly variable scores).

Fig. 19 summarizes the results across three experimental axes.

First, when varying the rejection percentile of a single filter from 0% to 90% (Fig. 19, left), runtime at 50M tokens remains within 3.5% of the unfiltered baseline, and this spread shrinks to just 0.3% at 500M tokens (Fig. 19, center)—well within normal run-to-run variance. The lack of degradation even at 90% rejection (where 9 out of 10 samples are discarded) can be attributed to two factors: (i) the predicate check is cheap relative to downstream processing, allowing the dataloader to skip rejected samples faster than the GPU can consume accepted ones; and (ii) our models are more compute-bound than the CLIP models trained in DataComp, since every sample passes through both a vision encoder and a full language model—this leaves more headroom in the data-loading pipeline, making it naturally tolerant of the additional filtering overhead.

Second, when scaling the number of simultaneous filters at a fixed 80% rejection rate per filter (Fig. 19, right), applying 1–2 filters actually runs *faster* than the unfiltered baseline (0.98×), likely because rejected samples bypass the more expensive downstream stages (image decoding, pixel-shuffle, tokenization, packing). Even with 3 concurrent filters—the most aggressive configuration we tested, where the compound rejection rate exceeds 99%—the overhead is only 1.10× the unfiltered baseline.

These results demonstrate that our online filtering is a practical and storage-efficient alternative to offline resharding for VLM pretraining at our scale, enabling rapid iteration over filtering configurations without materializing each filtered subset on disk.

I Fine-grained sweep of mixture optimization experiments

In Sec. 4.2 of the main paper, we conducted a full scaling grid across 3 model sizes (1B, 2B, 4B), 3 token budgets (6.25B, 12.5B, 25B) and 3 mixing ratios (Instruction-heavy, Balanced, Caption-heavy). Here, we aim to do a finer-grained sweep over 9 total mixtures. To that end, we conduct a dense line search between the orpotion of image-caption data and instruction-tuning data, which means that we gradually decrease the proportion of crawled Image-Text data in favour of Instruction Tuning data. We keep the multimodal docs and text proportion fixed at 5% and 15% respectively, and only vary the image-caption to instruction-tuning data proportion from 65% captioning-15% instruction to 10% captioning-70% instruction.

In Sec. 4.2 of the main paper, we conducted a full scaling grid across 3 model sizes (1B, 2B, 4B), 3 token budgets (6.25B, 12.5B, 25B), and 3 mixing ratios (*Instruction-heavy*, *Balanced*, *Caption-heavy*). Here we complement that grid with a finer-grained sweep over 9 mixtures at two scales: *small* (1B model, 6.25B tokens) and *medium* (2B model, 25B tokens). Concretely, we conduct a dense line search between image-caption and instruction-tuning data gradually trading off image-text data for instruction-tuning data, while keeping the multimodal-document and text-only shares fixed at 5% and 15% respectively. The image-caption-to-instruction-tuning split is varied from 65% caption / 15% instruction (Caption-heavy mix in Sec. 4.2) down to 10% caption / 70% instruction (Instruction-heavy mix in Sec. 4.2) in 8 increments, yielding the 9 operating points reported in Fig. 20.

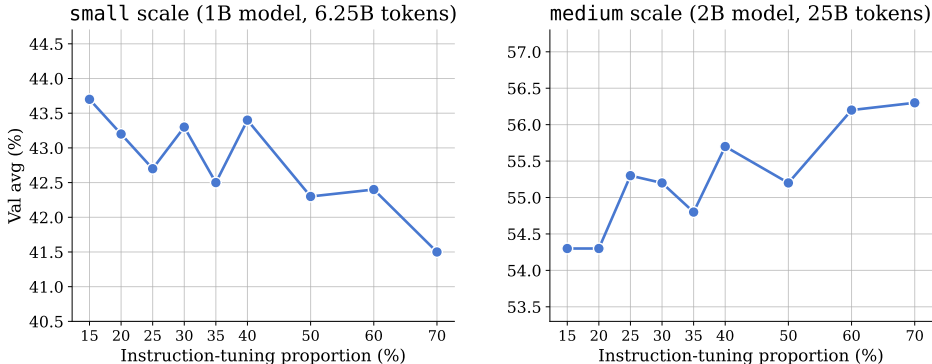


Figure 20: **The optimal pretraining data mixture is scale-dependent.** At the small scale, performance is flat or slightly degrades as the instruction-tuning share grows; at the medium scale, the trend reverses, with performance improving monotonically and peaking at 65% instruction-tuning data. This highlights the need for scale-aware data curation.

We find that the optimal data mixture at the *small* scale is the 65% caption-15% instruction mix, and performance gracefully degrades as we increase the proportion of instruction-tuning data in the mixture. Conversely, the trend reverses for the *medium* experiment, wherein the best mixture ratio is exactly at the opposite operating point of 10% caption-70% instruction. This highlights the importance of *scale-aware data-curation*—strategies that are optimal at one scale need not transfer to larger ones, and hence one must be prudent while making such transfer decisions, which has also been discussed in prior works [81, 217, 278].

J Packing Implementation Details

Our pretraining corpus mixes data types whose sequence lengths span several orders of magnitude—from single image-caption pairs of a few hundred tokens to multimodal documents and multi-turn instruction conversations that approach the full context window. Training on such data without packing forces every example to be padded to a common length, wasting a large fraction of every batch on padding tokens that carry no gradient. The standard remedy is *sequence packing*: concatenating several short samples into one fixed-length training example so that almost every token is a real token. For vision-language models, however, packing must respect a constraint that text-only packing does not: each image is split by dynamic tiling into a variable number of tiles, and every tile expands into a fixed block of visual tokens that the vision encoder must process. A pack is therefore bounded by *two* budgets at once, and we pack against both.

J.1 Dual constraints

Each emitted training example must simultaneously satisfy:

- **Token budget** $L = 8192$: the total number of (text + visual) tokens may not exceed the LLM context length we train at.
- **Image-tile budget** $M = 24$: the total number of image tiles—summed over all images of all samples in the pack—may not exceed M . This caps the vision-encoder footprint of a single forward pass and is the constraint unique to the multimodal setting.

A pack is considered *complete* as soon as it reaches *either* budget exactly, since a single text-heavy multimodal document can saturate L with few images, while a gallery of single-tile thumbnails can saturate M while leaving L far from full.

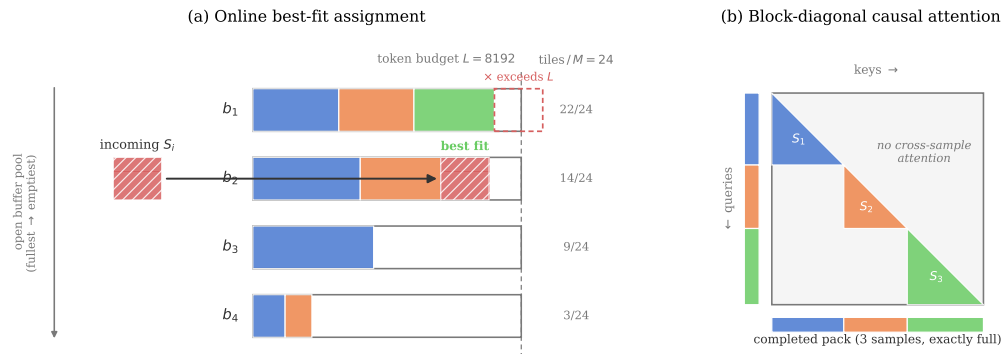


Figure 21: **Streaming best-fit sequence packing.** (a) Incoming samples are assigned online to a pool of open buffers kept sorted from fullest to emptiest. A sample S_i is placed in the *first* (hence fullest) buffer whose token budget L and image-tile budget M both still accommodate it. Here b_1 would overflow L , so S_i lands in b_2 . A buffer is emitted once it hits either budget exactly (or when the pool overflows its size cap). (b) Within a completed pack, a variable-length attention mask enforces block-diagonal causal attention, so each packed sample attends only within its own segment and position ids restart per segment—numerically identical to unpacked training.

J.2 Streaming best-fit bin-packing

Samples are produced one at a time by the WebDataset mixing stream described in Sec. H.6—already filtered, templated, and tokenized—so we never see the full dataset and cannot solve the offline (optimal) bin-packing problem. We instead maintain a pool of *open buffers*, each a partially-filled pack, and assign samples greedily as they arrive (Fig. 21a). The pool is kept sorted in order of *decreasing* image-tile occupancy (fullest first), an invariant we exploit to make assignment a single linear scan. For each incoming sample S_i :

1. **Best-fit search.** Scan the pool from fullest to emptiest and select the *first* buffer b for which adding S_i would violate neither budget, i.e. $\text{tiles}(b) + \text{tiles}(S_i) \leq M$ and $\text{tokens}(b) + \text{tokens}(S_i) \leq L$. Because the pool is fullest-first, this returns the most-occupied pack that can still host S_i —a best-fit choice that drives buffers toward completion rather than leaving many half-full.
2. **Assignment.** If such a buffer is found, S_i is concatenated into it. Otherwise S_i seeds a fresh singleton buffer. Each token of S_i is tagged with a within-pack sample index (`data_index`), recording which original sample it belongs to (used below to prevent cross-sample attention).
3. **Re-insertion.** The (possibly grown) buffer is re-inserted at the position that restores the fullest-first ordering.

A buffer is emitted as a finished training example under one of two conditions:

- **Exactly full:** it reaches either budget exactly ($\text{tiles} = M$ or $\text{tokens} = L$). If this happens, these are the fully well-packed examples and are emitted immediately, fullest first.
- **Pool overflow:** if the number of open buffers exceeds a cap $B = 20$, the fullest buffer is force-emitted even if not exactly full. This bounds the memory and latency cost of holding many partially-filled packs and is the only source of imperfectly-filled examples in practice.

No explicit padding is introduced during packing itself. The only padding is whatever the collator adds to align tensors within a micro-batch, and is compensated for by per-sample loss weighting (see details below).

J.3 Preventing cross-sample contamination

A single pack contains several independent samples that must not attend to one another, or the model would condition predictions on unrelated context. We enforce this at the attention level rather than by re-padding. The per-token `data_index` tags are converted at collation time into cumulative segment boundaries (`cu_seqlens`) and passed—in place of a dense attention mask—into a patched attention kernel (`flash_attn_varlen_func`) for the LM backbone. This yields exact **block-diagonal attention** (Fig. 21b): every sample attends only within its own segment, and position ids restart at zero at each segment boundary, so a packed sample is numerically identical to the same sample trained in isolation. Finally, because samples within a pack have very different lengths, we re-weight the per-token loss by a function of each segment’s effective (loss-bearing) token count, so that a single long sample does not dominate the pack’s gradient relative to the other short samples in a pack.

J.4 Stateful and resumable packing

Packing is both streaming and stateful: at any instant a worker holds a pool of partially-filled buffers that are not derivable from the dataloader position alone. To make training resumable without re-seeing or dropping data at a checkpoint boundary, each $(\text{rank}, \text{worker})$ pair persists its open buffer pool to disk (`rank<r>_worker<w>_buffers.pt`) alongside the WebDataset shard/sample/epoch cursors that record exactly how far each source has been consumed. On resume, the pool and cursors are restored and packing continues from precisely the state it was checkpointed in. This buffer state is written as part of the same `dataset_state` mechanism used for dataloader resumption.

In all our runs we pack to $L = 8192$ tokens and $M = 24$ image tiles, with a pool cap of $B = 20$ open buffers.

K Annotation Infrastructure

To run our filtering experiments, we need to *annotate* all samples in our data pool with the respective filtering metric—for every sample, and for every filter, we compute a scalar score for each filter (e.g., compute the CLIP score or the decontamination score according to a set of reference images). At our scale, this amounts to scoring trillions of tokens under many filters whose resource profiles range from cheap CPU-only text-based classifiers (e.g., length) to GPU-bound encoders (e.g., CLIP score). We build a Ray-based⁶ annotation framework that runs inside a Slurm allocation, parallelizes across nodes, and tolerates common failures (decode errors, OOMs, transient file-system issues, Slurm preemption) that any multi-day multi-node annotation job encounters. We plan to release the framework as part of the DCVLM benchmark.

K.1 Pipeline and Actor Model

A run is parameterised by an input directory of WebDataset [24] shards and a list of filter names. Each filter declares its per-actor CPU, GPU, and memory requirements via a `BaseFilter` interface. Because a worker runs the selected filters sequentially on a shared decoded batch, the framework sizes the actor pool from the maximum declared requirement across filters on each axis (CPU, GPU, DRAM). A run proceeds in three overlapping phases:

1. **Extraction.** One Ray task per tar reads the raw bytes (without decoding any sample components like images, json files etc.), groups samples into batches of B samples, and places each batch in the Ray object store. Decoding is deferred so that PIL images never round-trip through the object store, which keeps inter-node traffic to compressed bytes only.
2. **Processing.** A fixed pool of `FilterExecutionWorker` actors is created once at the start of the run. For each batch, the actor decodes once and runs every filter on the decoded samples in sequence, returning per-sample results and per-filter statistics. The actor model where each actor has some form of state is important. Loading the scoring model dominates per-batch cost for most filters, so we want to pay for it once per actor rather than once per batch, and decoding once per batch lets multi-filter runs share work.
3. **Streaming writes.** As soon as all batches for a tar resolve, we schedule a write task. This decouples write IO from active extraction and inference, and enables continuation after a Slurm interruption.

We support two output modes. In `-only-write-metadata` mode, each input tar `xxxxx.tar` creates a sibling `xxxxx.json` mapping every sample key to a flat dict of filter scores. The original shards are left untouched. In the alternative full-rewrite mode, the same scores are embedded as a `filters.json` field inside a freshly written tar. This is convenient when a downstream consumer wants a self-contained WebDataset.

K.2 Backpressure and Fault Tolerance

A `max_batches_in_flight` cap bounds the number of outstanding processing tasks, which in turn bounds the working set of the Ray object store. Without it, extraction outruns the GPU pool by orders of magnitude and PIL-decoded images blow up RAM (and trigger costly object-store spilling).

We tolerate failures at four levels with intentionally minimal recovery logic. (i) Sample decode errors are retried N times then skipped or raised per a configurable handler. (ii) A batch that raises during filtering is resubmitted up to `max_resubmits` times (default 5); the actor is health-checked via a ping before being returned to the idle pool. (iii) If an actor dies (`RayActorError`, e.g. kernel OOM-kill or a CUDA fault), the in-flight task is resubmitted and a best-effort replacement actor is spawned, so the pool does not shrink monotonically over the run. (iv) At the job level, atomic `.tmp+rename` writes are combined with a startup pass that skips tars whose final output already exists, so an interrupted job can be resumed by simply re-running it. The core assumption we rely on is that filters are deterministic, which makes duplicate computation under retry safe.

⁶<https://github.com/ray-project/ray>

L Additional SFT Results

The SFT control experiments in Sec. 4.3 rely on two design choices that we fix once and reuse across all 54 runs: the SFT token budget and the peak learning rate. We first describe how we set both (Sec. L.1), and then verify that our conclusions are robust to the choice of SFT dataset (Sec. L.2).

L.1 SFT Compute Budget and Learning-Rate Selection

SFT token budget. Rather than fine-tuning each checkpoint with an arbitrary fixed schedule, we scale the SFT budget proportionally to each checkpoint’s pretraining budget. We estimate the SFT-to-pretraining token ratio of InternVL3 [365] to be ≈ 0.29 and apply this multiplier to each of our three pretraining budgets, yielding SFT budgets of 1.8B, 3.6B, and 7.2B tokens for checkpoints pretrained on 6.25B, 12.5B, and 25B tokens, respectively. Holding this ratio fixed ensures that larger-compute checkpoints receive proportionally more fine-tuning, rather than coupling our conclusions to a single absolute SFT schedule.

Hyperparameter considerations. For pretraining we use a fixed peak LR of 2×10^{-5} across all scales, chosen via a sweep spanning three orders of magnitude. For SFT, common practice across recent VLMs is to keep the LR within the same order of magnitude as pretraining: InternVL-2.5 [41] and Nemotron Nano-V2-VL [58] reuse the pretraining LR unchanged, MM1 [213] and MiMo-VL [275] stay within the same order of magnitude (the latter increasing it slightly), and PerceptionLM [44] reuses the LR but halves the global batch size, effectively doubling it under the linear-scaling rule. Only Molmo [57] drops by a full order of magnitude (from 2×10^{-4} to 1×10^{-5}), and additionally uses separate LRs for the language model, projector, and vision encoder.

Learning-rate sweep. Guided by this, we keep the global batch size fixed at 1024 (matching pretraining) and sweep the SFT peak LR over $\{4 \times 10^{-6}, 2 \times 10^{-5}, 4 \times 10^{-5}\}$, i.e. one order of magnitude around the pretraining value. All runs use a cosine-decay schedule with 3% warmup and weight decay 0.01. We sweep on two representative 2B checkpoints—both pretrained on the Instruction-heavy mix that is optimal at this scale—one pretrained for 6.25B tokens (fine-tuned for 1.8B SFT tokens) and one for 12.5B tokens (fine-tuned for 3.6B), using LLaVA-665K [175] as the SFT dataset.

Tab. 25 reports the results. The lowest LR, 4×10^{-6} , is best on Core Avg at both checkpoints, while the highest LR, 4×10^{-5} , is consistently worst—most visibly on Text (-4.4pp and -4.5pp relative to 4×10^{-6} at the 6.25B and 12.5B checkpoints), indicating that an overly aggressive SFT LR erodes the language capabilities established during pretraining. The pretraining LR (2×10^{-5}) is a close runner-up. We therefore adopt a peak LR of 4×10^{-6} for all SFT runs in this work. Reassuringly, SFT lifts every checkpoint above its pretrained starting point at the chosen LR ($+2.4$ and $+2.8$), without altering the underlying pretraining ranking.

Table 25: **SFT learning-rate sweep.** We fine-tune two 2B checkpoints (pretrained on the Instruction-heavy mix for 6.25B and 12.5B tokens) with LLaVA-665K [175] at three peak LRs, keeping the global batch size fixed at 1024. The lowest LR (4×10^{-6}) is best at both checkpoints, while the highest LR degrades Text performance. Best Core Avg per block in **bold**. Categories follow Tab. 3.

Setting	Gen	Know	OCR	Vision	MTL	Text	Core Avg
2B model, 6.25B pretraining tokens \rightarrow 1.8B SFT tokens							
Pretrained (no SFT)	55.1	56.9	41.0	45.0	39.6	47.2	47.9
+ SFT, LR 4×10^{-6}	59.9	60.5	43.1	45.9	43.8	47.2	50.3
+ SFT, LR 2×10^{-5}	61.3	60.5	42.4	45.7	43.6	45.0	49.9
+ SFT, LR 4×10^{-5}	59.5	58.3	40.3	44.1	41.6	42.8	47.8
2B model, 12.5B pretraining tokens \rightarrow 3.6B SFT tokens							
Pretrained (no SFT)	59.0	59.0	43.9	45.1	42.4	48.1	50.0
+ SFT, LR 4×10^{-6}	63.8	62.1	48.6	47.6	44.4	48.0	52.8
+ SFT, LR 2×10^{-5}	64.5	61.1	46.4	47.8	45.9	45.6	52.0
+ SFT, LR 4×10^{-5}	63.8	59.9	44.8	47.5	44.6	43.5	50.8

L.2 Robustness to the SFT Dataset

Figure 22 replicates the analysis of Section 4.3 using Mammoth-VL-12M [89] as the SFT dataset in place of LLaVA-665K [151]. Results are fully consistent: pretraining and post-SFT scores remain near-perfectly correlated (Pearson $r=0.99$; Spearman $\rho=0.99$), and the pretraining ranking is preserved across all 27 checkpoints. These results confirm that the conclusion is robust to the choice of SFT dataset: a stronger pretraining checkpoint consistently yields a stronger fine-tuned model. In particular, the advantage of Instruction-heavy pretraining mixtures is *not* diminished by the additional instruction-tuning signal introduced during SFT.

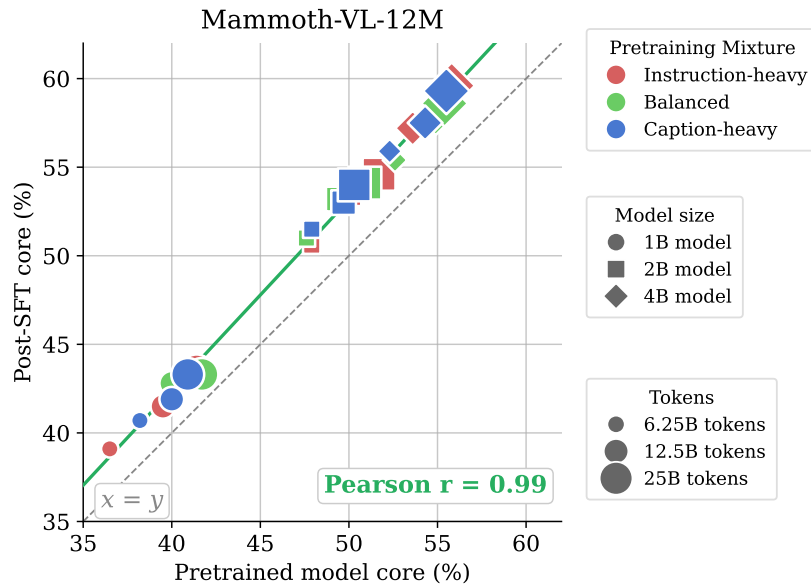


Figure 22: **Pretraining performance predicts post-SFT performance (*cont.*)**. We extend the analysis of Sec. 4.3 (Fig. 6) to another SFT dataset, i.e., Mammoth-VL-12M [89]. Remarkably, this leaves our main observation that *pretraining predicts post-SFT performance* unchanged.

M Extended Suite Evaluations

Table 26 reports results on our 52-task Extended evaluation suite, which augments the Core suite with Hallucination (**Hall**), Safety (**Safety**), and Reasoning (**Reason**) categories.

DCVLM-BASELINE maintains its overall advantage over FINEVISION at both scales, achieving +3.0pp and +3.9pp gains at `large` and `x-large`, respectively. Gains are most pronounced in Vision-Centric (+23.1pp / +21.3pp) and General Understanding (+8.4pp / +8.5pp) tasks, consistent with the Core suite trends.

Regarding newly added categories, Hallucination scores also favor DCVLM-BASELINE at both scales (+3.4pp / +3.0pp), suggesting our data mixture yields better-calibrated visual grounding. However, two notable weaknesses emerge. First, DCVLM-BASELINE substantially underperforms on Safety (−17.7pp / −19.2pp across both scales). Second, FINEVISION retains an advantage on OCR (+6.5pp / +5.2pp) and Reasoning (+4.3pp / +4.9pp), suggesting its document-rich sources benefit reading and understanding details, text, figures, and equations in images. We leave the exploration of how to address these gaps, particularly regarding safety alignment, for future work.

Table 26: **Extended** benchmark results comparing DCVLM-BASELINE to FINEVISION at `large` (top) and `x-large` (bottom) scales. Category acronyms: **Gen** = General – **Know** = Knowledge-Centric – **OCR** = OCR & Charts – **Vision** = Vision-Centric – **MTL** = Multilingual – **Hall** = Hallucination – **Text** = Text-Only – **Reason** = Reasoning.

Dataset	Gen	Know	OCR	Vision	MTL	Hall	Safety	Text	Reason	Ext Avg
large scale (4B model, 100B tokens)										
FINEVISION	60.7	70.7	65.2	30.9	45.1	65.6	39.8	56.3	38.9	53.0
DCVLM-BASELINE (ours)	69.1	67.6	58.7	54.0	50.9	69.0	22.1	59.7	34.6	56.0
x-large scale (8B model, 200B tokens)										
FINEVISION	65.3	72.8	64.1	39.2	48.4	65.9	43.5	60.4	44.4	56.6
DCVLM-BASELINE (ours)	73.8	73.0	58.9	60.5	56.1	68.9	24.3	67.1	39.5	60.5

N Safety Preprocessing: Harmful, Toxic and Biased Content

As with all large uncurated data-pools (largely sourced from the web), especially for ones that are fully open-sourced, there can be concerns regarding releasing harmful and unsafe content openly. We invested some efforts into ensuring that our raw data-pool has as little unsafe content as possible, via two main analysis methods—(1) Text-based removal and (2) Image-level Safety Analysis.

N.1 Unsafe, Biased and Toxic Text Removal

To ensure data quality and safety, we filter unsafe, biased, and toxic content in DCVLM pool using Detoxify [93], a transformer-based multilingual toxicity classifier. The classifier produces scores across multiple harm categories which include toxicity, severe toxicity, obscenity, threat, insult and identity attack. We retain a sample only if every category score falls at or below a threshold of 0.1, following the filtering criteria used in DataComp-CLIP [73].

N.2 Image-level Safety Analysis

We analyze a subset of 20,000 image–text pairs to assess image safety. For this purpose, we employ ShieldGemma2 [342], a 4B-parameter image content moderation model built on Gemma 3, which provides robust safety risk predictions across multiple categories, including sexually explicit content, violence and gore, and dangerous content. Within the sampled dataset, the model classifies 19 images as dangerous, 117 as sexually explicit, and 4 as containing violence and gore, each with a confidence score of at least 90%. This suggests that potentially unsafe content represents only a small fraction of the dataset, indicating a relatively low overall presence of harmful imagery and further supports using the analyzed subset as a representative estimate of the safety characteristics of the full dataset.

O Per-Dataset Core Results

For completeness and ease of inspection, we report the per-dataset scores underlying the per-category Core averages of Tab. 3. Each table below corresponds to one of our six Core categories and lists every constituent benchmark for the same set of models and scales as Tab. 3. Averaging a row’s datasets within a table recovers that row’s per-category number in Tab. 3, and averaging across all 33 datasets (six tables) recovers its Core Avg.

Table 27: **Per-dataset Core results: General Understanding (Gen)**. Per-dataset breakdown of the **Gen** column of Tab. 3. The final **Gen** column is the row mean over these 7 datasets and reproduces Tab. 3. **Bold** = best per scale.

Method	MMB-EN	MMB-CN	MMB-EN _{VL1}	MMB-CN _{VL1}	SEED-IMG	MMT	VMC	Gen
small scale (1B model, 6.25B tokens)								
LLAVA-ONEVISION-1.5	12.0	22.3	11.5	21.7	35.1	22.5	31.9	22.4
NEMOTRON-VL-2	16.4	5.4	14.0	4.6	38.6	29.2	31.7	20.0
FINEVISION	39.2	34.3	36.7	33.2	50.7	39.5	47.4	40.1
DCVLM-BASELINE (ours)	38.7	38.3	36.9	37.2	46.6	38.3	47.5	40.5
medium scale (2B model, 25B tokens)								
LLAVA-ONEVISION-1.5	29.9	31.1	29.1	30.4	44.2	29.9	38.2	33.3
NEMOTRON-VL-2	49.2	48.5	46.4	47.0	62.1	41.1	46.2	48.6
FINEVISION	59.3	49.8	57.4	47.3	64.8	47.4	61.2	55.3
DCVLM-BASELINE (ours)	66.2	62.6	64.2	60.9	64.5	51.6	66.0	62.3
large scale (4B model, 100B tokens)								
NEMOTRON-VL-2	27.1	26.3	21.8	23.1	42.5	35.2	44.6	31.5
FINEVISION	67.1	53.4	64.0	51.5	66.5	42.7	67.5	59.0
DCVLM-BASELINE (ours)	72.2	71.2	69.3	68.0	71.5	57.8	69.1	68.4
x-large scale (8B model, 200B tokens)								
FINEVISION	67.3	69.2	63.9	66.1	66.0	46.2	65.6	63.5
DCVLM-BASELINE (ours)	79.0	75.2	76.7	73.0	74.1	59.7	73.4	73.0

Table 28: **Per-dataset Core results: Knowledge-Centric (Know)**. Per-dataset breakdown of the **Know** column of Tab. 3. The final **Know** column is the row mean over these 4 datasets and reproduces Tab. 3. **Bold** = best per scale.

Method	MMMU	SciQA	AI2D	AI2D _{nm}	Know
small scale (1B model, 6.25B tokens)					
LLAVA-ONEVISION-1.5	21.4	40.7	38.2	39.0	34.8
NEMOTRON-VL-2	24.7	48.9	42.1	43.0	39.7
FINEVISION	35.7	58.1	43.3	45.5	45.7
DCVLM-BASELINE (ours)	36.8	52.4	42.0	43.4	43.6
medium scale (2B model, 25B tokens)					
LLAVA-ONEVISION-1.5	25.6	48.4	47.2	51.0	43.0
NEMOTRON-VL-2	28.2	65.1	59.3	65.6	54.5
FINEVISION	42.7	72.2	63.0	72.7	62.7
DCVLM-BASELINE (ours)	44.4	70.6	58.5	68.5	60.5
large scale (4B model, 100B tokens)					
NEMOTRON-VL-2	32.7	65.2	54.6	62.7	53.8
FINEVISION	40.9	89.5	69.7	82.6	70.7
DCVLM-BASELINE (ours)	47.1	75.8	68.2	79.5	67.7
x-large scale (8B model, 200B tokens)					
FINEVISION	46.1	87.4	73.3	84.4	72.8
DCVLM-BASELINE (ours)	51.1	79.5	75.1	86.3	73.0

Table 29: **Per-dataset Core results: OCR & Charts (OCR)**. Per-dataset breakdown of the **OCR** column of Tab. 3. The final **OCR** column is the row mean over these 6 datasets and reproduces Tab. 3. **Bold** = best per scale.

Method	ChartQA	OCRBench	InfoVQA	SEED2+	OCRVQA	CharXiv	OCR
small scale (1B model, 6.25B tokens)							
LLAVA-ONEVISION-1.5	0.0	17.4	0.0	27.1	0.0	4.7	8.2
NEMOTRON-VL-2	0.2	2.6	0.2	30.6	8.5	5.2	7.9
FINEVISION	35.3	53.2	31.2	45.7	35.6	9.1	35.0
DCVLM-BASELINE (ours)	22.9	43.6	28.8	42.7	44.2	15.6	33.0
medium scale (2B model, 25B tokens)							
LLAVA-ONEVISION-1.5	0.0	59.1	0.0	36.1	0.0	30.8	21.0
NEMOTRON-VL-2	0.0	45.3	0.1	48.2	7.6	18.3	19.9
FINEVISION	57.4	75.6	57.3	57.7	48.9	14.1	51.8
DCVLM-BASELINE (ours)	44.2	63.3	41.8	57.8	46.5	21.1	45.8
large scale (4B model, 100B tokens)							
NEMOTRON-VL-2	1.3	45.2	3.3	44.7	26.9	20.4	23.6
FINEVISION	67.1	82.4	69.9	63.6	42.6	27.8	58.9
DCVLM-BASELINE (ours)	49.0	75.6	47.1	63.0	65.6	24.0	54.0
x-large scale (8B model, 200B tokens)							
FINEVISION	76.4	83.8	72.7	65.0	17.9	29.3	57.5
DCVLM-BASELINE (ours)	34.8	66.9	53.3	66.0	69.9	29.9	53.5

Table 30: **Per-dataset Core results: Vision-Centric (Vision)**. Per-dataset breakdown of the **Vision** column of Tab. 3. The final **Vision** column is the row mean over these 5 datasets and reproduces Tab. 3. **Bold** = best per scale.

Method	MMT-MI	BLINK	MUIR	CVB-2D	CVB-3D	Vision
small scale (1B model, 6.25B tokens)						
LLAVA-ONEVISION-1.5	22.8	16.1	23.1	28.7	48.2	27.8
NEMOTRON-VL-2	29.7	28.2	26.4	46.0	37.3	33.5
FINEVISION	39.6	39.2	23.0	48.5	54.5	41.0
DCVLM-BASELINE (ours)	38.9	38.2	26.3	43.0	49.1	39.1
medium scale (2B model, 25B tokens)						
LLAVA-ONEVISION-1.5	28.9	19.4	25.7	31.9	46.2	30.4
NEMOTRON-VL-2	41.3	33.1	33.3	51.4	46.2	41.1
FINEVISION	47.1	39.6	31.0	59.0	52.1	45.8
DCVLM-BASELINE (ours)	51.3	41.6	30.2	55.7	58.0	47.4
large scale (4B model, 100B tokens)						
NEMOTRON-VL-2	35.4	37.6	25.4	44.1	50.2	38.5
FINEVISION	41.5	16.0	40.5	58.8	38.5	39.1
DCVLM-BASELINE (ours)	56.9	46.0	36.9	68.3	77.6	57.1
x-large scale (8B model, 200B tokens)						
FINEVISION	44.4	35.6	41.5	58.7	67.9	49.6
DCVLM-BASELINE (ours)	59.6	50.7	47.6	75.2	84.2	63.5

Table 31: **Per-dataset Core results: Multilingual (MTL)**. Per-dataset breakdown of the **MTL** column of Tab. 3. The final **MTL** column is the row mean over these 3 datasets and reproduces Tab. 3. **Bold** = best per scale.

Method	MTVQA	MMMB	MTL-MMB	MTL
small scale (1B model, 6.25B tokens)				
LLAVA-ONEVISION-1.5	2.8	32.2	5.7	13.6
NEMOTRON-VL-2	1.0	41.6	5.8	16.1
FINEVISION	10.3	47.7	26.5	28.2
DCVLM-BASELINE (ours)	7.5	46.7	22.1	25.4
medium scale (2B model, 25B tokens)				
LLAVA-ONEVISION-1.5	13.5	35.3	15.7	21.5
NEMOTRON-VL-2	16.3	60.9	33.0	36.7
FINEVISION	18.6	61.2	42.1	40.6
DCVLM-BASELINE (ours)	16.4	63.8	52.3	44.2
large scale (4B model, 100B tokens)				
NEMOTRON-VL-2	16.4	49.5	16.6	27.5
FINEVISION	21.0	65.5	48.7	45.1
DCVLM-BASELINE (ours)	20.2	71.6	60.9	50.9
x-large scale (8B model, 200B tokens)				
FINEVISION	23.1	71.0	51.0	48.4
DCVLM-BASELINE (ours)	22.8	76.8	68.5	56.0

Table 32: **Per-dataset Core results: Text-Only Understanding (Text)**. Per-dataset breakdown of the **Text** column of Tab. 3. The final **Text** column is the row mean over these 8 datasets and reproduces Tab. 3. **Bold** = best per scale.

Method	C-Eval	CMMLU	GAOKAO	MMLU	NQ	RACE	TriviaQA	WinoG	Text
small scale (1B model, 6.25B tokens)									
LLAVA-ONEVISION-1.5	1.5	0.0	6.3	13.3	2.9	12.9	16.9	1.0	6.8
NEMOTRON-VL-2	17.0	14.2	19.6	32.3	0.6	44.1	5.8	32.3	20.7
FINEVISION	21.9	20.2	17.9	37.4	9.7	55.6	23.0	45.3	28.9
DCVLM-BASELINE (ours)	26.5	49.3	21.6	46.2	7.0	58.8	18.3	50.3	34.8
medium scale (2B model, 25B tokens)									
LLAVA-ONEVISION-1.5	10.7	0.5	23.0	24.7	8.3	24.3	33.0	3.6	16.0
NEMOTRON-VL-2	22.8	29.9	25.2	43.5	0.3	59.2	5.8	41.8	28.6
FINEVISION	31.9	61.1	38.3	59.7	15.5	75.6	37.1	51.3	46.3
DCVLM-BASELINE (ours)	32.8	64.5	39.7	61.6	13.6	80.0	35.3	55.2	47.8
large scale (4B model, 100B tokens)									
NEMOTRON-VL-2	30.7	55.5	35.2	48.9	4.0	57.2	9.9	49.6	36.4
FINEVISION	33.0	66.8	37.4	65.1	18.4	82.0	46.1	60.8	51.2
DCVLM-BASELINE (ours)	35.5	69.5	41.4	66.2	19.5	86.8	43.8	67.8	53.8
x-large scale (8B model, 200B tokens)									
FINEVISION	36.0	72.5	46.3	70.4	21.5	85.0	51.4	62.2	55.7
DCVLM-BASELINE (ours)	38.2	75.4	51.9	71.5	26.9	90.7	55.5	78.9	61.1

P Limitations and Future Directions

Here, we highlight some limitations and open questions of our work which may be useful for exciting future research.

P.1 Limitations

- **Scaling to longer token budgets.** In this work, we see clear scaling trends in downstream performance across our model and token budget scales from `small` (1B, 6.25B tokens) to `x-large` (8B, 200B tokens), however we remain compute-limited to a 200B token pretraining budget. Competitive open-weight alternatives such as Qwen2-VL and Qwen3-VL train on 1.4T and 2.2T tokens during pretraining respectively.
- **Scaling to larger model sizes.** In addition to scaling token budgets during the pretraining stage, stress-testing our data pool and data mixture conclusions at model sizes $\geq 20\text{B}$ is an important research question not empirically addressed in this work.
- **Testing DCVLM data on native multimodal pretraining.** The pretraining strategy adopted in this work starts with already pretrained vision encoders and language models. However, recent works have advocated for native pretraining paradigms to enable training VLMs from scratch, which is not something we have implemented in this work. Effectively the dominance of instruction-heavy data mixtures may not exist when the inductive biases of separately pretrained encoders are absent, and is an important research question worth addressing.
- **More principled scaling ladder.** Our four-scale ladder is not tied to the joint optimization of model size and data token budgets, currently set up as practical defaults instead of theoretical optimal estimates. Designing a more principled scaling ladder, especially for vision-language models that use pretrained components, is a research problem of its own and we consider this out-of-scope in this work.
- **Capability Coverage.** While DCVLM spans a broad range of capabilities in both the extended and core set of evaluations, certain tasks like *grounding and detection* are absent. While we observe issues with these benchmarks and tasks and highlighted them in Sec. F.1, incorporating these tasks in a faithful evaluation setup is a natural and critical followup.
- **Poor OCR performance.** We conclude that our OCR performance lags behind FINEVISION due to their better coverage and priority given to high-resolution scanned documents and pdf-derived sources. Whether simply scaling up OCR data or curating extremely high-quality and information-dense samples can bridge this gap is not adequately resolved in this work.

P.2 Future Work

- **Towards instructional multimodal documents.** Our results suggest that instruction-heavy mixtures scale more favorably with compute, raising an important question: how can we better exploit and transform the vast supply of raw multimodal documents? Such transformations may preserve the breadth and diversity of naturally occurring multimodal data while providing a denser and more targeted learning signal for instruction-tuned multimodal models.
- **Fine-grained mixture optimization.** Our mixing experiments only involved dense line-search sweeps over the proportion of image-caption pairs and instruction-tuning data in the mixture. A direction worth exploring is to use swarm-based methods [37, 179, 317] using proxy models and verify if they discover similar findings of instruction-heavy mixes being more scalable.
- **Token-level mixing.** Our mixing experiments operationalized data mixing at the *sample level*, i.e. we fix the data type ratios based on the number of samples in each dataset. However, a good direction to explore would be to operationalize mixing at the token-level. As seen from Sec. E.2, the distribution of token-counts and sample-counts per-dataset can vary, and hence it would be an interesting avenue to compare and contrast different mixing strategies across.
- **Quality-aware upsampling.** In our work, we only used all the quality-scores for data filtering, i.e. to discard data samples below a certain threshold. This is exclusionary by

nature, and might induce harmful biases into models [61, 102]. Another potential use for such quality-scores which does not involved discarding data is using those scores for upsampling data samples rather than filtering.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: This work proposes a benchmark for controlled data-centric experiments to improve VLM training. Additionally, we show that data mixing is the key to assembling high-quality training data. Finally, the baseline dataset (DCVLM-Baseline) obtained from our results enables training models to achieve an accuracy improvement of +5.4pp. The benchmark is described in Sec. 3, results on data mixing in Sec. 4.2, and improvements over existing pretraining datasets in Sec. 5.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The limitations are discussed in the “Limitations and Future Directions” section (Sec. P.1).

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate “Limitations” section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren’t acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results and proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: While general details regarding data collection, curation, and training are provided in the main paper, the Appendix fully complements these, reporting trained model architecture details (Sec. C), the training hyperparameters used (Sec. D), the annotation infrastructure (Sec. K), train-test decontamination (Sec. G), DCVLM data pool details (Sec. E), our evaluation suite (Sec. F), and safety preprocessing details (Sec. N).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The submission comes with a hyperlink to where the data is stored, while the Appendix fully reports the necessary details to reproduce our results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The main experimental procedure is described in Secs. 3 to 5, and more details are provided in the Appendix (Secs. C to G).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Error bars are not reported as they are too computationally expensive to compute.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: For each experiment “scale” (small, medium, large, x-large), Tab. 1 reports the number of ‘H100 hrs’ required to reproduce our results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn’t make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We carefully reviewed the NeurIPS Code of Ethics and believe our work complies with it.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: Since DCVLM is a benchmark for controlled data-centric experiments with the goal of improving VLM training, our work fits as fundamental research and, therefore, it is not tied to particular applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: Section N reports the safety processing details we used to reduce as much as possible harmful, toxic, and biased content.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Section E.3 and Tab. 12 report links pointing to the datasets we used in the DCVLM pool, with all available licenses associated. Furthermore, we properly credited authors by citing their work.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The link to the data attached to this submission contains a README file explaining how the data is organized and how to navigate it.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Our work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper is about training large vision-language models (rather than using them as components). We describe the architecture and the pretrained LLM we use in Section 3.2 and Appendix C.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.