Eagle 2: Building Post-Training Data Strategies from Scratch for Frontier Vision-Language Models

Zhiqi Li[1,*] **Guo Chen**[1,*] **Shilong Liu**[2,*] **Shihao Wang**[3,*] **Vibashan VS**[4,*] **Yishen Ji**[1] **Shiyi Lan** **Hao Zhang**
Yilin Zhao[5,*] **Subhashree Radhakrishnan** **Nadine Chang** **Karan Sapra** **Amala Deshmukh** **Tuomas Rintamaki**
Matthieu Le **De-An Huang** **Ilia Karmanov** **Lukas Voegtle** **Philipp Fischer** **Timo Roman** **Tong Lu**[1]
Jose M. Alvarez **Bryan Catanzaro** **Jan Kautz** **Andrew Tao** **Guilin Liu**[†] **Zhiding Yu**[†]

Abstract: Recently, promising progress has been made by open-source vision-language models (VLMs) in bringing their capabilities closer to those of proprietary frontier models. However, most open-source models only publish their final model weights, leaving the critical details of data strategies and implementation largely opaque. In this work, we address VLM post-training from a data-centric perspective, showing the key role of data strategy in developing frontier VLMs. By studying and building our post-training data strategy from scratch, we share detailed insights into the development processes, aiming to benefit the development of competitive models for the open-source community. Our introduced data strategy, together with training recipes and model design, leads to a family of performant VLMs named *Eagle 2*. Specifically, Eagle2-9B achieves state-of-the-art results across various multimodal benchmarks, matching certain competitive models with up to 70B parameters.

Links: Github Code | HF Models | Demo

1. Introduction

Built upon large language models (LLMs), vision-language models (VLMs) [1, 2, 3, 4] aim to enable LLMs to see. With the ability to visually perceive the world, VLMs are able to take in multimodal information, and as a result, handle a broader range of intelligent applications. There is thus a growing interest to use VLMs as the backbone for reasoning and decision making in various applications, such as intelligent agents [5], autonomous driving [6, 7], and embodied AI [8, 9, 10].

The community has delved deeply into the architecture and training methodologies of VLMs with significant advances. A predominant strategy to align the vision and language modalities is through post-training on pretrained LLMs, with the LLaVA family [4] being the representative examples. Based on the level of transparency, current VLM models can also be broadly categorized into three types: 1) commercially closed-source models (e.g., GPT-4v/o [11] and Claude [12]), 2) frontier models with publicly available weights (e.g., Qwen2-VL [13], InternVL2 [14] and Llama 3.1 [15]), and 3) fully open-source models (e.g., Cambrian-1 [16] and the LLaVA family [4, 17]).

Recently, some frontier models with publicly available weights have been shown to match closed-source commercial models on key benchmarks while offering better customization for downstream applications. However, the technical details provided by these models are often insufficient for reproduction. On the other hand, fully open-source models tend to disclose extensive technical details, including both the dataset strategies and training recipes. These details unveils the secret sources in building customized VLM models, which enables easier reproduction and helps the community to develop technologies faster. However, most of the open source models still lag behind their frontier counterparts. For instance, on the OpenCompass [18] benchmark, LLaVA-OneVision-72B [17] still ranks slightly behind InternVL2-40B [14] despite having a stronger LLM backbone. We thus ask the following question: *What could help the community to develop more competitive open-source frontier VLMs?*

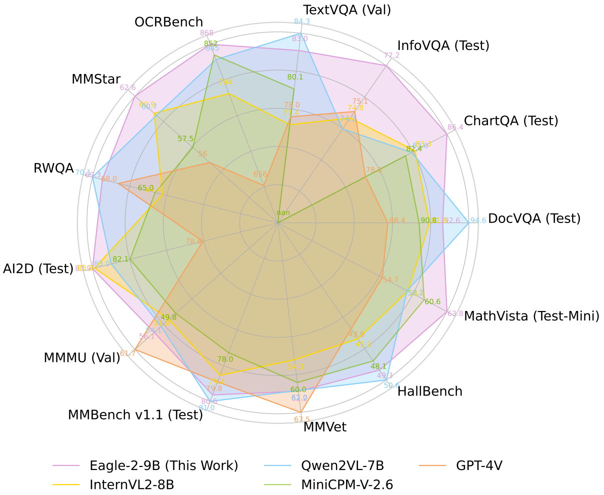


Figure 1 | Overview of Eagle2-9B's result across different multimodal benchmarks, in comparison to state-of-the-art open-source and commercial frontier models.

---

[*] *Work Done during an internship at NVIDIA.* [†] *Equal advising and corresponding authors. Additional affiliations:* [1] *with Nanjing University,* [2] *with Tsinghua University,* [3] *with Hong Kong Polytechnic University,* [4] *with Johns Hopkins University,* [5] *with New York University.*

© 2025 NVIDIA. All rights reserved.

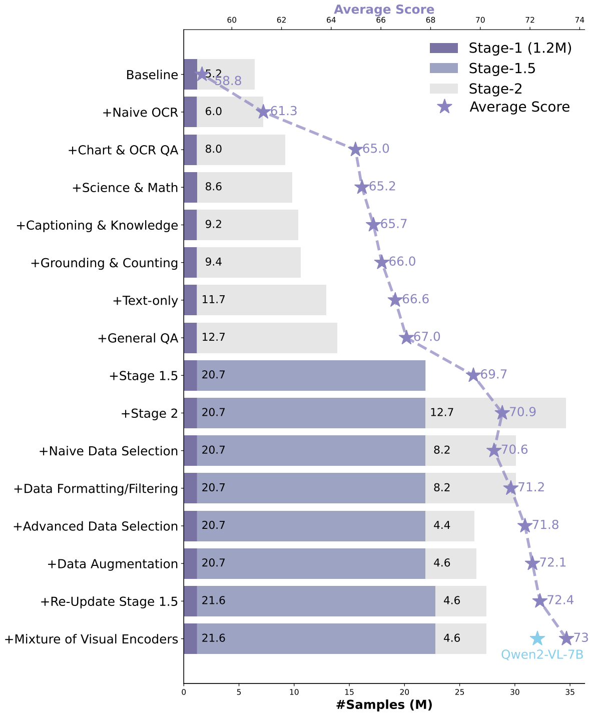


Figure 2 | Step-by-Step ablation of Eagle 2. We show the average scores of 13 benchmarks, detailed numbers are in Table 5 and 6.

1.1. Data Strategy

Our answer to the above question is the data-strategy. Assuming the same pre-trained LLM backbone, we posit that data is the most decisive factor to obtain high-quality models. We thus adopt a centralized strategy to build our post-training data. For fully open-source models, various constraints such as computing resources may limit the study on more dataset sources, despite their intention to make the data recipe publicly available. This limitation often affects their capability compared to models that can access and utilize a wider range of data sources.

"Diversity first, then quality". We follow this principle throughout our development and push it to the extreme. Our optimization of the data results in consistent improvements in model. Our contributions involve: 1) a data collection strategy leading to a large-scale highly diverse data pool with **180+** sources, 2) a data filtering strategy to remove low-quality samples, 3) a data selection strategy to construct high-quality subsets, and 4) a series of data augmentation techniques to enrich the existing data. This series of strategies are shown to improve the model significantly.

1.2. Model Architecture

Besides data strategy, another important factor considered in this work is model architecture. Since different architecture designs have been well-studied in open-source models, their properties are relatively transparent to the community. Despite various designs such as Q-Former [19] and Hybrid-Attention [20], simple MLP connector is still the most popular choice to connect the vision encoder and LLM. With the rapid advances in architecture designs in contemporary VLMs, model structure is no longer a primary factor driving performance differences among models. However, this does not imply that there is no room for further improvement in architecture.

Tiled mixture of vision encoders. Inspired by the works of InternVL [21, 14], Eagle [22] and Cambrian-1 [16], we follow a vision-centric design where we adopt both dynamic tiling and mixture of vision encoders (MoVE) in one unified design. Specifically, each image tile is encoded by channel-concatenated MoVE, therefore allowing high-resolution input from tiling while maintaining the robust perception from MoVE. Similar to [22], we follow a "SigLIP [23] + X (ConvNeXt [24])" configuration. Compared to SigLIP alone, tiled MoVE yields significant improvements despite having tiling, particularly in tasks like OCR and Chart/Document VQA.

1.3. Training Recipe

Which training recipe to be used? In this context, the training recipe primarily refers to various configurations for training a VLM. With the same dataset, different recipes can still have a significant impact on the final performance. Although the training recipes for the state-of-the-art VLMs [25, 14, 26] are somewhat unclear, the details shared by existing work [16, 4, 17] can offer a solid baseline. However, to further improve model performance, it is necessary to explore more effective training recipe.

Three-stage training. We adopt a three-stage training strategy to best leverage the training data. In particular, the first stage (Stage-1) is used to align language and image modality via training the MLP connector. The next stage (Stage-1.5) trains the full model with a large-scale diverse data. The final stage (Stage-2) continues training the full model with a carefully crafted, high-quality visual instruction tuning dataset. In Stage-1.5, we incorporate all available visual instruction data, rather than limiting it to captioning or knowledge data alone. Our results show that this approach yields substantial improvements over the commonly used two-stage training strategy [4]. We also identify limitations in existing open-source frameworks concerning data packing and introduce a balanced data packing approach to address the issue.

1.4. Summary

Our extensive exploration on data strategy, model architecture and training recipe is shown in Fig. 2, resulting in a family of VLMs named *Eagle 2*. Through sharing

Name		Eagle2-Baseline
Model	Vision Encoder	Siglip-400M [23]
	Connector	MLP
	LLM	Qwen2.5-7B-Instruct [27]
Resolution		$448\times\{(1,1),(1,2),(2,2)\cdots,(1,12)\}$
Stage-1	Training Data	ALLaVA(1.2M) [4]
	Trainable Module	Connector
Stage-2	Training Data	Cambrian-1 (5.2M) [16]
	Trainable Module	Full Model

Table 1 | Baseline Settings.

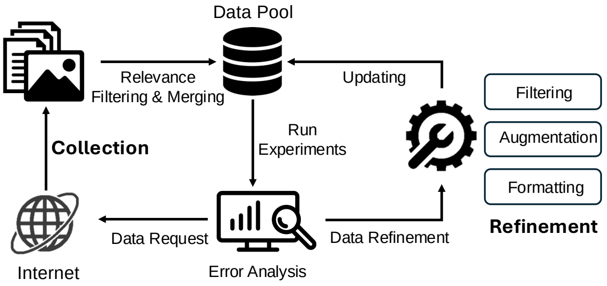


Figure 3 | An overview of our data strategy. The upper part shows the date collection pipeline and the lower part shows the data refinement pipeline.

the journey of exploration, we aim to "teach readers to fish than to give them a fish", by sharing our data strategies, model design and training recipe, detailing the iterative development process than merely displaying the final benchmark results and demonstrations. The Eagle 2 model family spans a range of various scales, including 1B, 2B and 9B parameters. As shown in Fig. 1, Eagle2-9B can match or outperform frontier open-source and commercially closed-source VLMs on a set of common multimodal benchmarks.

2. Method

2.1. Baseline Setting

As shown in Tab. 1, our initial baseline starts with the Cambrian dataset [16] using LLaVA's [4] two-stage training recipe. We remove some low-quality data from Cambrian-7M, such as ShareGPT-4V, GPT-77K and Data-Engine-161K, ultimately resulting in a subset of 5.2M samples. The model incorporates an MLP connector to bridge the vision encoder with the LLM and employs image tiling for dynamic resolution. Starting from this baseline, we enhance Eagle 2 in three key aspects: (1) data strategy, (2) training recipe, and (3) model architecture. These optimizations enable the model to achieve state-of-the-art performance.

2.2. Data Strategy

Training data is essential for defining a VLM's capabilities. However, most commercial VLMs and leading VLMs with publicly available weights keep their data strategies confidential. In this work, we conducted an in-depth exploration to create a diverse and high-quality dataset with a series of data strategies to iteratively refine and optimize our data pool. The resulting dataset significantly boosts model performance, far surpassing the initial baseline. Fig. 3 illustrates our overall data strategy consisting of two core components: data collection and optimizing existing data. More technical details have been provided in the appendix.

Data collection - diversity is the key. A model's capability is strongly correlated with the diversity of data. As such, collecting data as diverse as possible is a key principle of this work, leading to two main strategies:

- *Passive gathering:* Monitoring the latest related datasets from arXiv manuscripts and HuggingFace Datasets and adding them into our candidate list.
- *Proactive searching:* Addressing the bucket effect. As shown in Fig. 3, for each update of the data pool, we generate error analysis to identify model weaknesses and perform targeted searches for new data.

Our diverse data sources are summarized in Tab. 2a and generally publicly available. We utilize some pre-organized dataset collections [16, 17, 151] to speed up preparation but also conducted careful inspection to prevent issues like test data leakage[1]. We also collect a large amount of public non-QA data, such as Google Landmark [31], and convert them into VQA data using specific rules or auto-labeling tools.

To reduce training costs, we avoid performing ablation for each dataset individually. Instead, datasets with similar domains are added in batches to the data pool when meeting the following criteria:

- *Maintaining overall accuracy without noticeable regression for every considered benchmark.*
- *Introducing meaningful diversity to the current domains.*

To help quantify the diversity, we define a metric called *Similarity Score* to measure the relevance between a new data source and the current data pool as follows:

$$S_k = \frac{1}{N}\sum_{i=1}^{N} \max_{1\leq j\leq M_k} \left(\mathrm{Sim}(I_i, I_j) \times \mathrm{Sim}(T_i, T_j)\right), \quad (1)$$

where i is the index of a new data source with N samples, and j is the index of the existing pool with M samples, with k denoting the data category. We compute similarity scores only within the same category, as inter-category similarity is generally low. Image embeddings I_i and I_j are generated from SSCD [178], and text embeddings T_i and T_j from all-mpnet-base-v2 [179]. The similarity score

---

[1] The test split of AI2D is used in Cambrian-1 training data.

Category	Dataset
Captioning & Knowledge	ShareGPT4o [28], KVQA [29], Movie-Posters [30], Google-Landmark [31], WikiArt [32], Weather-QA [33], Coco-Colors [34], music-sheet [35], SPARK [36], Image-Textualization [37], SAM-Caption [38], Tmdb-Celeb-10k [39]
Mathematics	GeoQA+ [40], MathQA [41], CLEVR-Math/Super [42, 43], Geometry3K [44], MAVIS-math-rule-geo [45], MAVIS-math-metagen [45], InterGPS [46], Raven [47], GEOS [48], UniGeo [49]
Science	AI2D [50], ScienceQA [51], TQA [52], PathVQA [53], SciQA [54], Textbooks-QA, VQA-RAD [55], VisualWebInstruct [56]
Chart & Table	ChartQA [57], MMC-Inst [58], DVQA [59], PlotQA [60], LRV-Instruction [61], TabMWP [62], UniChart [63], Vistext [64], TAT-DQA [65], VQAonBD [66], FigureQA [67], Chart2Text [68], RobuT-{Wikisql, SQA, WTQ} [69], MultiHiertt [70]
Naive OCR	SynthDoG [71], MTWI [72], LVST [73], SROIE [74], FUNSD [75], Latex-Formula [76], IAM [77], Handwriting-Latex [78], ArT [79], CTW [80], ReCTs [81], COCO-Text [82], SVRD [83], Hiertext [84], RoadText [85], MapText [86], CAPTCHA [87], Est-VQA [88], HME-100K [89], TAL-OCR-ENG [89], TAL-HW-MATH [89], IMGUR5K [90], ORAND-CAR [91], Invoices-and-Receipts-OCR [92], Chrome-Writting [93], IIIT5k [94], K12-Printing [89], Memotion [95], Arxiv2Markdown, Handwritten-Mathematical-Expression [96], WordArt [97], RenderedText [98], Handwriting-Forms [99]
OCR QA	DocVQA [100], InfoVQA [101], TextVQA [102], ArxivQA [103], ScreencQA [104], DocReason [105], Ureader [106], FinanceQA [107], DocMatrix [108], A-OKVQA [109], Diagram-Image-To-Text [110], MapQA [111], OCRVQA [112], ST-VQA [113], SlideVQA [114], PDF-VQA [115], SQuAD-VQA, VQA-CD [116], Block-Diagram [117], MTVQA [118], ColPali [119], BenthamQA [120]
Grounding & Counting	TallyQA [121], OODVQA [122], RefCOCO/+/g (en) [123, 124], GroundUI [125]
General VQA	LLaVA-150K [4], LVIS-Instruct4V [126], ALLaVA [127], Laion-GPT4V [128], LLAVAR [129], SketchyVQA [122], VizWiz [130], IDK [131], AlfworldGPT, LNQA [132], Face-Emotion [133], SpatialSense [134], Indoor-QA [135], Places365 [136], MMinstruct [137], DriveLM [138], YesBut [139], WildVision [140], LLaVA-Critic-113k [141], RLAIF-V [142], VQAv2 [143], MMRA [144], KONIQ [145], MMDU [146], Spot-The-Diff [147], Hateful-Memes [148], COCO-QA [149], NLVR [150], Mimic-CGD [151], Datikz [152], Chinese-Meme [153], IconQA [154], Websight [155]
Text-only	Orca [156], Orca-Math [157], OpenCodeInterpreter [158] MathInstruct [159], WizardLM [160], TheoremQA [161], OpenHermes2.5 [162], NuminaMath-CoT [163], Python-Code-25k [164], Infinity-Instruct [165], Python-Code-Instructions-18k-Alpaca [166], Ruozhiba [167], InfinityMATH [168], StepDPO [169], TableLLM [170], UltraInteract-sft [171]

(a) Summary of the collected Eagle 2 SFT datasets

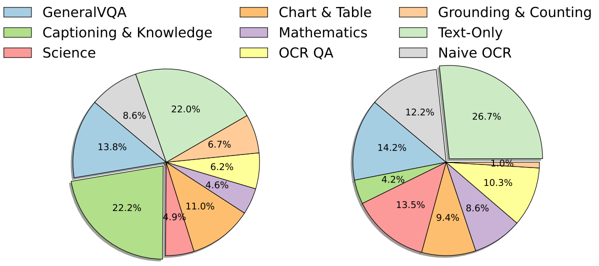
Category	Dataset
Captioning & Knowledge	CC3M [172], TextCaps [173], ShareGPT-4V [174], DenseFusion-1M [175]
Grounding & Counting	Object 365 [176]
Text-only	OpenMathInstruct [177]

(b) Summary of the additional Stage 1.5 datasets

Table 2 | Dataset used in Eagle 2. Dataset in Magenta is internal data.

between samples is the product of image and text similarity. This metric shows most sources have low similarity, with a few high-similarity samples removed as duplicates.

Following our data collection protocol and the refinement steps stated below, our final model uses 21.6 M samples in Stage-1.5 and 4.6 M samples in Stage-2, with the distribution illustrated in Fig. 4. We make sure text-only data occupy over 20%. Captioning data account for the largest proportion in Stage-1.5; however, in Stage-2, we reduce its share primarily due to concerns about the overly monotonous instructions.



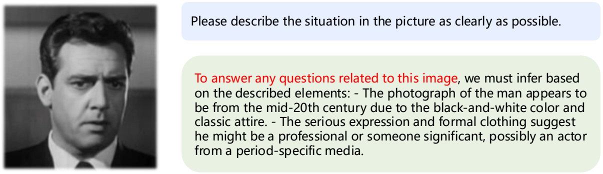
(a)Data Distribution in Stage 1.5 (b)Data Distribution in Stage 2

Figure 4 | Data Distribution for Stage-1.5 and Stage-2.

Data filtering - "A rotten apple spoils the barrel." Public datasets often contain many low-quality samples. We find that most low-quality cases belong to the following categories, which we use as our filtering criteria:

- *Mismatching question-answer pair.* E.g., Fig. 5 (a) from ShareGPT4o [28].
- *Irrelevant image-question pair with unrelated image and question.* E.g., Fig. 5 (b) from Cambrian-1 [16].
- *Repeated texts.* E.g., Fig. 5 (c) from ShareGPT-4V [174].
- *Numeric formatting issue.* Excessive decimal precision or overly precise numerical answers lacking corresponding information in the image. E.g., Fig. 6.

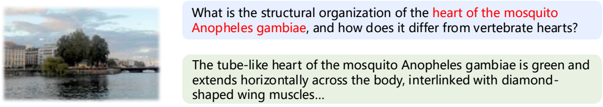
Since most low-quality data are generated from synthesis, they often present characteristics making them distinguishable for removal through rule-based filtering.



Please describe the situation in the picture as clearly as possible.

To answer any questions related to this image, we must infer based on the described elements: - The photograph of the man appears to be from the mid-20th century due to the black-and-white color and classic attire. - The serious expression and formal clothing suggest he might be a professional or someone significant, possibly an actor from a period-specific media.

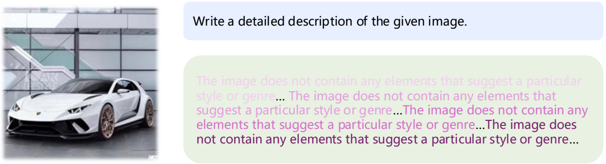
(a) Irrelevant question-answer pair from (**ShareGPT-4o**)



What is the structural organization of the heart of the mosquito Anopheles gambiae, and how does it differ from vertebrate hearts?

The tube-like heart of the mosquito Anopheles gambiae is green and extends horizontally across the body, interlinked with diamond-shaped wing muscles...

(b) Irrelevant image-question pair (from **Cambrian-1 Data Engine**)

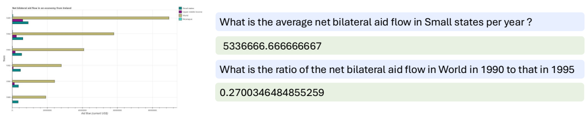


Write a detailed description of the given image.

The image does not contain any elements that suggest a particular style or genre... The image does not contain any elements that suggest a particular style or genre...The image does not contain any elements that suggest a particular style or genre...The image does not contain any elements that suggest a particular style or genre...

(c) Repeated texts (from **ShareGPT-4V**)

Figure 5 | Typical low-quality samples.



What is the average net bilateral aid flow in Small states per year ?

5336666.666666667

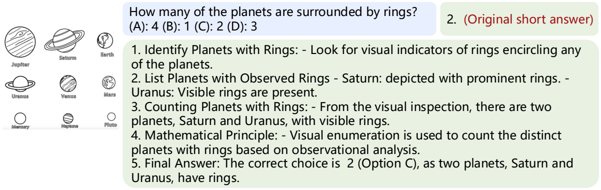
What is the ratio of the net bilateral aid flow in World in 1990 to that in 1995

0.2700346484855259

Figure 6 | Public datasets [60, 58] often do not rigorously handle numerical precision, resulting in high decimal precision impossible to directly extract from the image.

Subset selection - "every data comes with a price". Selecting optimal subsets is key to high-quality training. Recent work [16] suggests limiting the number of samples from each source to be typically no more than K (e.g. 350K). Our data selection adopts on two main principles:

- *Subset quantity determination.* Data source diversity and distribution determine the sample quantity. Auto-labeled sources are featured by larger sizes, but often contain errors and lack diversity. Instead, manually labeled datasets are often smaller. Thus, datasets with larger original sizes are generally applied with smaller sampling ratios. In our Stage-2 data, the average size per source is around 20K, with the largest subset VisualWebInstruct [56] having 263K samples.
- *K-means clustering selection.* Once the subset size is determined, the next step is to select the samples. Current methods often use random selection, which is suboptimal. For example, in chart data, histogram samples are more frequent than other types like line charts or pie charts, and random sampling wouldn't ensure balance across these types. To address this, we applied unsupervised K-means clustering on SSCD [178] image embeddings, where samples with similar chart types are clustered closer, allowing for target data selection, such as including all the line and pie chart samples as needed. While K-means using SSCD image embeddings performs poorly on natural scene images, it excels with mathematical, medical, and document-based data.

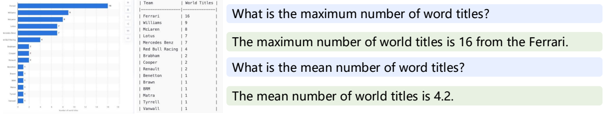


How many of the planets are surrounded by rings? (A): 4 (B): 1 (C): 2 (D): 3

2. (Original short answer)

1. Identify Planets with Rings: - Look for visual indicators of rings encircling any of the planets.
2. List Planets with Observed Rings - Saturn: depicted with prominent rings. - Uranus: Visible rings are present.
3. Counting Planets with Rings: - From the visual inspection, there are two planets, Saturn and Uranus, with visible rings.
4. Mathematical Principle: - Visual enumeration is used to count the distinct planets with rings based on observational analysis.
5. Final Answer: The correct choice is 2 (Option C), as two planets, Saturn and Uranus, have rings.

(a) Chain-of-Thought Augmentation



```
| Team           | World Titles |
|----------------|--------------|
| Ferrari        | 16           |
| Williams       | 9            |
| McLaren        | 8            |
| Lotus          | 7            |
| Mercedes Benz  | 7            |
| Red Bull Racing | 4           |
| Brabham        | 2            |
| Cooper         | 2            |
| Renault        | 2            |
| Benetton       | 1            |
| Brawn          | 1            |
| BRM            | 1            |
| Matra          | 1            |
| Tyrrell        | 1            |
| Vanwall        | 1            |
```

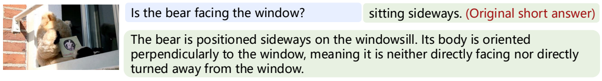
What is the maximum number of word titles?

The maximum number of world titles is 16 from the Ferrari.

What is the mean number of word titles?

The mean number of world titles is 4.2.

(b) Rule-based Augmentation



Is the bear facing the window?

sitting sideways. (Original short answer)

The bear is positioned sideways on the windowsill. Its body is oriented perpendicularly to the window, meaning it is neither directly facing nor directly turned away from the window.

(c) Short Answer -> Long answer Augmentation

Figure 7 | Our three most commonly used data augmentation methods.These methods rely on rule-based approaches or utilize VLM models for automatic labeling.

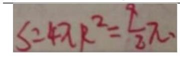
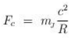
Formular Image	LaTeX Annotation
	`S=4\piRˆ2=\frac{9}{2}\pi`
 $F_c = m_J \frac{c^2}{R}$	`\begin{align*}F_c=m_{J}\frac{cˆ2}{R}\end{align*}`

Table 3 | Two samples with same "Extract LaTeX from image" task but with different format.

Data augmentation. Data augmentation aims to mine the rich information from input images that is not fully present in the existing QA annotations. In order to mine the potentially useful information from image space, a common approach is to use third-party VLMs to generate fine-grained descriptions of the images. We adopt this approach, as illustrated in Fig. 7.

- *Adding CoT (Chain-of-Thought) explanations.*
- *Rule-based QA generation.*
- *Expanding short answers into longer responses.*

For details of the above generation process, kindly refer to the supplementary.

Data formatting. Transforming data into the correct format is also a crucial step in data preparation. One of our basic principles is: *"same task, similar format; different tasks, clearly distinct formats."* Our data formatting includes but is not limited to:

- *Removing unnecessary decorations.* We present a seemingly inconspicuous example that, nonetheless, has a profound impact on the final results in Tab. 3. We show two samples from different sources for the task of extracting LaTeX formulas from images. The second sample includes an unnecessary fixed equation environment.

		Stage-1	Stage-1.5	Stage-2
Vision	**Resolution**	$\{448_{\text{SigLIP}}, 512_{\text{ConvNeXt*}}\} \times \{(i,j) \mid i,j \in \mathbb{Z}^+,\, i \times j \leq 12\}$		
	Tokens	$(i \times j + 1) \times 256$		
Data	**Dataset**	ALLaVA	Rich Diverse Data	High-Quality Data
	#Samples	1.2M	21.6M	4.6M
Model	**Trainable**	MLP Connector	Full Model	Full Model
	Qwen2.5-0.5B	4.9M	0.9B	0.9B
	Qwen2.5-1.5B	9.4M	2.0B	2.0B
	Qwen2.5-7B	40.0M	8.9B	8.9B
Training	**Batch Size**	1024	1024	256
	Learning Rate	2×10^{-4}	$\{2, 4\} \times10^{-5}$	$\{2, 4\} \times10^{-5}$
	Max Length	4096	8192	16384

Table 4 | We present our three-stage training settings, where Eagle2-9B/2B/1B builds upon Qwen2.5-32B/7B/1.5B/0.5B [27], respectively. *: For small scale model with 0.5/1.5B LLM, we only use SigLIP as visual encoder and learning rate of 4×10^{-5} in Stage-1.5 & 2.

Even with limited use of such data, the model consistently outputs this fixed template (in red texts). After removing the fixed equation environment, the model returns to normal behavior, leading to a significant improvement on OCRBench [180].

- *Appending more specific instructions.* Adding detailed instructions to the original instruction based on the response is a common approach. For example, appending "Provide a short answer" to brief responses helps prevent a model from becoming an "answering machine" that is used to giving short answers. However, overusing this can also hinder generalization. For instance, adding "Please answer yes or no" to every yes-or-no question may impair the model's ability to answer correctly without such prompt during inference.

2.3. Training Recipe

Our data strategy enables us to build a high-quality and diverse dataset, but applying different training recipes to the same data pool still has a decisive impact on the final results. Our recipe is built upon the following core points.

Post-pretraining stage is necessary. We initially begin with LLaVA [4]'s two-stage training strategy, where we train an MLP connector followed by full model training with SFT data. While efficient, this approach proved unsuitable for quick SFT data updates, as the expanding SFT data makes it harder to track the impact of new data and reduces the experimental efficiency. For instance, we observe improvements from expanding the Cambrian-1 [16] SFT data. However, the gap remains between the model and state-of-the-art ones. Considering that the main limitation of the two-stage strategy is the lack of robust pre-training, we add an additional pre-training stage (Stage-1.5). Stage-1.5 pre-trains the model on a larger dataset to reduce dependency on SFT data in subsequent training.

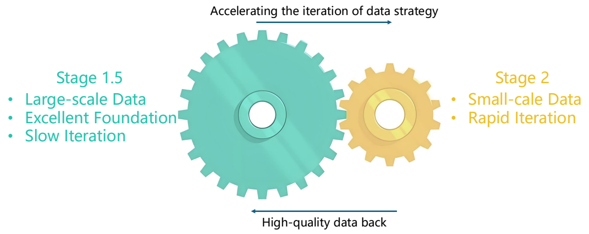
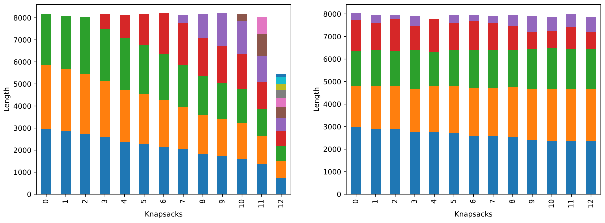


Figure 8 | Stage-1.5 accelerates Stage-2 and Stage-2 gives Stage-1.5 feedback.

"Large wheel drives small wheel." Three-stage pre-training is, in fact, widely used in existing works, such as LLaVA-OneVision [17]. However, we have a distinctly different view to the data that using in Stage-1.5. Other works tend to use more knowledge-related data, such as captioning data, at this stage. In this work, we add all data sources intended for visual instruction to Stage-1.5, simultaneously introducing several other datasets as shown in Tab. 2b. As shown in Fig. 8, training Stage-2 based on Stage-1.5 enables rapid iteration on a high-performance foundation. The derived conclusions are more robust than those obtained from ungeneralizable ablation experiments on toy-scale data. In addition, the effective conclusions obtained from Stage-2 can be used to update Stage-1.5, further driving improvements in model performance. Detailed settings are shown in Tab. 4.



(a) Knapsacks of naive greedy knapsack method. (b) Knapsacks of balanced knapsack method.

Figure 9 | We pack 64 samples of varying lengths into combined samples, each with a length of less than 8192. The naive greedy knapsack approach in LLaMa-Factory [181] leads to uneven length distributions (left), while the proposed method gives more balanced length distributions within every knapsack (right).

Balance-aware data packing matters. Data packing speeds up training by concatenating shorter samples, reducing padding use. In our experiments, using packing accelerate the training by 2-3 times. A key step in packing is arranging N short samples of varying lengths into M long samples without exceeding max length. Existing frameworks such as LLaMa-Factory use a naive greedy knapsack algorithm[2], but this often produces packs with

---

[2] https://github.com/hiyouga/LLaMA-Factory/blob/main/src/llamafactory/data/processors/

```python
# Our proposed greedy knapsack method
def balanced_greedy_knapsack(samples, L):
    # Step 1: Sort the samples
    samples.sort(reverse=True)
    total_length = sum(samples)
    min_knapsacks = (total_length + L - 1) // L
    # Step 2: Initialize knapsacks
    knapsacks=[[] for _ in range(min_knapsacks)
        ]
    knapsack_lengths = [0] * min_knapsacks
    # Step 3: Distribute samples across
        knapsacks
    ks_index = 0
    sample_index = 0
    while sample_index < len(samples):
        length = samples[sample_index]
        if knapsack_lengths[ks_index]+length<=L
            :
            knapsacks[ks_index].append(length)
            knapsack_lengths[ks_index] +=
                length
            sample_index += 1
        else:
            knapsacks.append([])
            knapsack_lengths.append(0)
        ks_index = argmin(knapsack_lengths)

    return knapsacks
```

Figure 10 | Python code for the proposed balance-aware greedy knapsack method. L is the max length and "samples" is a list of token lengths.

uneven length distributions. As shown in Fig. 9, the naive greedy knapsack method groups long and short samples separately, which is not desirable to model training.

Therefore, we design a balance-aware greedy knapsack algorithm that creates packs with a more uniform length distribution, as shown in Fig. 10, ensuring that each pack contains both long and short samples. Unlike SPFHP (Shortest-Pack-First Histogram Packing) [182], our method prioritizes balanced length distribution over packing efficiency, helping balance loss weights between long and short samples. Further details are in the appendix.

2.4. Tiled Mixture of Vision Encoders

Following Eagle [22], we use SigLIP [23] and ConvNeXt-XXLarge [24, 183] as vision encoders. Additionally, to handle arbitrarily high-resolution images, we employ image tiling following InternVL-1.5 [21]. The input resolution of every image tile of SigLIP is 448×448, while the input size of ConvNeXt is 512×512. To make sure they output same number of image tokens, we use PixelShuffle to conduct a $2\times$ downsampling on the image features from SigLIP, resulting a feature shape of 16×16, matching the output size of ConvNeXt ($32\times$ downsampling of input). We then concatenate these features along the channel dimension and align with LLM via an MLP layer.

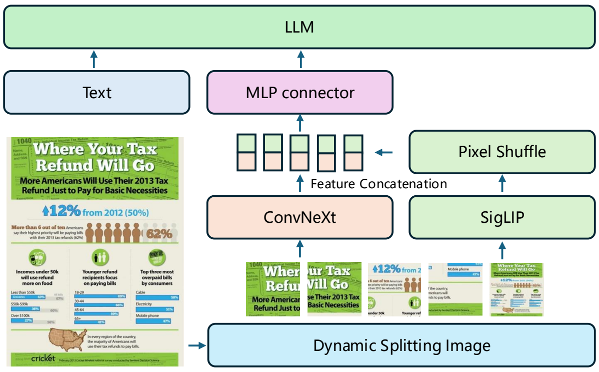


Figure 11 | Tiled Mixture of Vision Encoders.

3. Experiments

3.1. Evolution of Eagle 2

Scaling Stage-2 training data. We initially explore the impact of scaling Stage-2 data, as shown in Tab. 5. Our findings reveal that model's overall performance improved steadily with additional data, with the most notable gains arising from the inclusion of 2M (million) VQA samples focused on charts, tables, and OCR. While data scaling indicates potential for further gains beyond 10M samples, our experiments' costs have risen sharply, and the efficiency of data iteration has decreased. Moreover, we observe considerable performance fluctuations across specific benchmarks at this scale, especially in challenging benchmarks like MMMU, MathVista, and MMVet. Another obstacle is that, as illustrated by the data-performance growth trend in Fig.2, reaching the performance of frontier VLMs like Qwen2-VL would be difficult. These challenges leads us to consider adopting a more effective training strategy.

Introducing Stage-1.5. To build a robust pre-trained model, we implement Stage-1.5 where we focus on maximizing the data utilization to strengthen the model's foundational capabilities. As shown in Tab. 6, the Stage-1.5 checkpoint is competitive by itself, and subsequent Stage-2 training further improves the previous best model's performance by average 3.9%.

Naive data selection. Using a naive data selection strategy with maximum thresholds and random sampling, we reduce the training data to 8.6M; unfortunately, this led to a decline in performance. We speculate it might be that the randomly selected data have inadvertently excluded some valuable samples, while also failing to adequately ensure a balanced data distribution.

Data formatting & filtering. After filtering low-quality data and formatting the training set, we see clear improvements on 8 out of 14 benchmarks, including a remarkable 45-point gain on OCRBench [180]. This implies the im-

---

`processor_utils.py`

Model	Stage-1	Stage-2	DocVQA Test	ChartQA Test	InfoVQA Test	TextVQA Val	OCRBench Test	MMstar Test	RWQA Test	AI2D Test	MMMU Val	$\text{MMB}_{1.1}$ EN-Val	MMVet Test	HallB Test	MathVista Test-Mini	Average score
Cambrian-7B [16]	1.2M	7M	77.8	73.3	-	71.7	624	50.7	64.2	74.6	42.7	-	48.0	30.6	49.0	-
Eagle2-Baseline	1.2M	5.2M	77	65.9	50	69.9	589	47.3	60.3	74.7	49.7	71.2	45.9	44.8	49.2	58.8
+ Naive OCR	1.2M	+0.8M	78	67.0	57	73.7	719	49.3	59.9	74.4	50.7	72.1	45.5	46.3	50.7	61.3
+ Chart, Table & OCR QA	1.2M	+2.0M	88	78.3	65	77.6	783	51.7	62.7	76.2	50.1	72.7	50.1	39.9	54.1	65.0
+ Science & Math	1.2M	+0.7M	89	78.8	64	77.7	777	52.5	63.4	77.1	50.7	75.6	51.5	38.9	51.2	65.2
+ Caption	1.2M	+0.5M	88	79.0	63	78.0	784	53.7	61.0	77.2	52.9	77.9	55.8	39.5	49.7	65.7
+ Grounding & Counting	1.2M	+0.2M	88	79.4	64	77.9	792	54.3	61.7	77.1	51.8	77.6	54.2	39.6	53.7	66.0
+ Text-Only	1.2M	+2.3M	88	78.5	63	77.9	792	55.5	65.4	76.5	51.4	76.5	58.8	37.3	57.5	66.6
+ General	1.2M	+1.0M	88	80.0	63	77.8	797	55.9	65.9	76.9	52.2	78.9	56.8	40.8	55.5	67.0

Table 5 | **Data ablation under 2-Stage training.** The Stage-2 baseline uses Cambrian-1 data. In subsequent experiments, we gradually increase the SFT data, adding only specific categories each time.

Model	Stage-1	Stage-1.5	Stage-2	DocVQA Test	ChartQA Test	InfoVQA Test	TextVQA Val	OCRBench Test	MMstar Test	RWQA Test	AI2D Test	MMMU Val	$\text{MMB}_{1.1}$ EN-Val	MMVet Test	HallB Test	MathVista Test-Mini	Average Score
Cambrian-7B [16]	1.2M	-	7M	77.8	73.3	-	71.7	624	50.7	64.2	74.6	42.7	68.2	48.0	30.6	49.0	-
Introducing Stage 1.5	1.2M	21M	-	89.9	82.7	71.1	79.5	812	58.5	69.4	78.6	50.9	81.7	55.7	47.1	60.1	69.7
+ Stage 2	1.2M	21M	12.7M	91.0	84.0	72.5	81.0	825	61.4	69.0	81.0	52.0	83.0	56.3	46.4	61.4	70.9
+ Naive Subset Selection	1.2M	21M	8.2M	90.4	83.7	72.0	79.7	798	62.8	67.8	83.6	51.7	81.9	56.1	46.7	61.0	70.6
+ Data Formatting/Filtering	1.2M	21M	8.2M	91.1	84.5	72.3	81.2	843	62.1	67.7	83.0	53.2	82.2	55.8	46.6	62.0	71.2
+ Advanced Subset Selection	1.2M	21M	4.6M	90.8	84.1	73.3	81.4	843	62.7	68.9	84.1	52.5	82.5	59.3	49.2	60.5	71.8
+ Data Augmentation	1.2M	21M	4.6M	91.8	85.1	73.6	81.1	839	61.4	68.4	83.9	53.6	82.4	60.1	48.4	63.5	72.1
+ Re-Update Stage 1.5	1.2M	22M	4.6M	91.3	85.9	73.3	81.9	842	61.7	68.2	83.5	53.6	82.4	61.3	49.0	65.2	72.4
+ Mixture of Vision Encoders	1.2M	22M	4.6M	92.6	86.4	77.2	83.0	868	62.6	69.3	83.9	56.1	81.9	62.2	49.3	63.8	73.5

Table 6 | Employing three-stage training strategy allows us to reduce the amount of training data required in the Stage-2, thereby expediting the data iteration process. The resultant efficient data strategies can then be leveraged to refresh and optimize the data in Stage-1.5.

Model	DocVQA Test	ChartQA Test	InfoVQA Test	TextVQA Val	OCRBench Test	MMstar Test	RWQA Test	AI2D Test	MMMU Val	$\text{MMB}_{1.1}$ Test	MMVet Test	HallB Test	MathVista Test-Mini	**Open-Compass**
Closed-Source Models														
GPT-4o-0513 [184]	92.8	85.7	-	-	736	63.9	75.4	84.6	69.2	82.2	69.1	55.0	61.3	69.9
GPT-4V [11]	88.4	78.5	75.1	78.0	656	56.0	68.0	78.6	61.7	79.8	67.5	43.9	54.7	63.5
Gemini-1.5-Pro [185]	93.1	87.2	81.0	78.7	754	-	70.4	-	62.2	-	-	-	63.9	64.4
Publicly Available Models														
LLaVa-OneVision-0.5B [17]	70.0	61.4	41.8	-	565	37.7	55.6	57.1	31.4	50.3	32.2	31.7	33.8	41.3
InternVL2-1B [14]	81.7	72.9	50.9	70.0	754	45.7	50.3	64.1	36.7	59.7	32.7	34.0	37.7	48.3
Eagle2-1B	81.8	77.0	54.8	76.6	767	48.5	55.4	70.9	38.8	63.0	40.9	35.3	45.3	52.4*
InternVL2-2B [14]	86.9	76.2	58.9	73.4	784	50.1	57.3	74.1	36.3	69.6	39.5	37.9	46.3	54.0
Qwen2-VL-2B [25]	90.1	73.0	65.5	79.7	809	48.0	62.6	78.9	41.1	72.2	49.5	41.7	43.0	57.2
Eagle2-2B	88.0	82.3	65.8	79.1	818	56.4	63.1	79.3	43.1	74.9	53.8	45.8	54.7	61.2*
InternVL2-8B [14]	91.6	83.3	74.8	77.4	794	60.9	64.4	83.8	51.8	79.4	54.2	45.2	58.3	64.1
Qwen2-VL-7B [27]	94.6	83.0	74.3	84.3	845	60.7	70.1	83.0	54.1	81.0	62.0	50.5	58.2	67.0
MiniCPM-V2.6 [186]	90.8	82.4	-	80.1	852	57.5	65.0	82.1	49.8	78.0	60.0	48.1	60.6	65.2
LLaVA-One-Vision-7B [17]	87.5	80.0	68.8	-	622	61.7	66.3	81.4	48.8	80.9	57.5	31.6	63.2	60.1
InternVL2-26B [14]	92.9	84.9	75.9	82.3	825	61.0	68.3	84.5	50.7	81.2	62.1	50.7	59.4	66.4
LLaVA-One-Vision-72B [17]	91.7	83.7	74.9	-	741	66.1	71.9	85.6	56.6	84.5	60.6	47.5	68.4	68.0
LLaMa-3.2-90B-Vision [15]	90.1	85.5	-	-	783	55.3	-	-	60.3	77.3	64.1	44.1	57.3	63.4
Eagle2-9B	92.6	86.4	77.2	83.0	868	62.6	69.3	83.9	56.1	80.6	62.2	49.3	63.8	68.2*

Table 7 | **Comparison with SoTA models on Various Benchmarks**. *: We obtain the OpenCompass [18] score by averaging across Eagle benchmarks (OCRBench, MMStar, AI2D, MMMU, $\text{MMB}_{1.1}$, MMVet, HallusionBench, and MathVista).

portance of carefully reviewing and utilizing existing data, as well as exercising caution with publicly available data.

Advanced data selection. By employing the comprehensive data selection strategy introduced in Sec. 2.2, we further reduce the dataset to 4.6M samples. By selecting a more balanced data subset and ensuring data quality, we achieve a further improvement in average score with a reduced amount of data.

Data augmentation. By employing our data augmentation strategy, we introduce a greater volume of automatically generated CoT training data, leading to notable performance improvements on MMMU and MathVista. The rule-based data augmentation on the chart data also brings 1 point improvement on ChartQA.

Re-updating stage-1.5. The effective data strategies we explored in Stage-2, such as data filtering, formatting, and augmentation, can be applied to update the Stage-1.5 data, thereby further enhancing the model's ultimate capability. By updating Stage-1.5 checkpoint, we obtain clear improvement on ChartQA, MMVet and MathVista.

Mixture of vision encoders. Introducing mixture of vision encoder has brought performance improvements on 12 out of 14 benchmarks, particularly in benchmarks related to documents, charts, and OCR. This clearly demonstrates that the mixture of vision encoders significantly enhances the model's understanding to visual spaces.

3.2. Comparison with SOTA Models

As shown in Tab. 7, we conducted comparisons across 14 diverse benchmarks with the representative state-of-the-art public avaiable and closed-source models. Our Eagle2-9B, building on top of Qwen2.5-7B [27], outperforms InternVL2-8B [14] and MiniCPM-v2.6 [186] across all 14 benchmarks and leads Qwen2-VL-7B [25] in 9 out of the 14 benchmarks and beats it on OpenCompass. Eagle2-9B holds its ground against much larger VLMs such as InternVL2-26B, LLaVa-OneVision-72B [17] and LLaMa-3.2-90B-Vision [15]. Apart from MMVet and MMMU, we comprehensively surpass GPT-4V. Eagle2-9B surpasses GPT-4o [184] on ChartQA, OCRBench, and MathVista, while achieving performance very close to GPT-4o on DocVQA, MMStar, AI2D and OpenCompass.

4. Related Work

Vision-Language Models (VLMs) LLMs [187, 188, 3] have transformed natural language processing (NLP) and reshaped the broader AI landscape. The advancement of LLMs has spurred significant progress in visual understanding by integrating visual features with LLMs, leading to the emergence of Visual-Language Models(VLMs) [189, 11, 4, 190]. The performance of VLMs with public available weights [4, 3, 191, 192, 193, 17, 194, 14, 25, 195, 15, 196, 197, 198, 199] continues to make breakthroughs, reaching or even surpassing the most advanced commercial models such as GPT-4V/4o [184, 11] and Gemini-1.5 [185]. Fully open-source VLMs [17, 16, 200] have released their training data and code base, further accelerating the VLM research.

Vision-Centric VLMs. Our work adopts a vision-centric VLM design that emphasizes strong vision foundation and HD input. This is aligned with the spirit of various related areas, including: 1) Vision foundation for VLMs [201, 202, 203] and improved designs [204, 23, 195, 205, 206, 207], 2) Mixture of vision encoder designs [208, 209, 210, 211, 212, 16, 22], and 3) Tiling and HD input designs [213, 214, 215, 216, 21, 193, 194, 217, 14, 25]. To our best knowledge, this work is the first to explore the tiled mixture of vision encoder (MoVE) design, which is shown to inherit the benefits from both worlds. The proposed tiled MoVE design also introduces additional flexibility to incorporate advanced vision foundation models.

Data Efforts in VLMs. Data strategy is crucial in training VLMs, encompassing aspects of data set construction, balance and filtering, and training methodologies. Early endeavors such as LLaVA-150K [4] used instructed tuning with GPT-4 [11], which was later enriched by successors [218, 219, 193, 204] incorporating academic training data from various tasks into the supervised fine-tuning stage. Studies also broadened data types to include video [220, 15], multi-image inputs [17, 14], image-text interleaved data [221, 222], multilingual data [186], and synthetic datasets [15]. However, simply expanding data sets can compromise model performance due to varying quality and size. Approaches like Instruct-BLIP [219] and Cambrian-1 [16] addressed this by devising optimal data ratios and balancing techniques, while others like Llama3 [15] and Molmo [223] focused on enhancing data quality by removing duplicates with SSCD [178] and incorporating human-annotated images, respectively. In addition, Training strategies have also evolved, with LLaVA [4] proposing a two-stage training process that has become a standard, and later models [17] introducing intermediate stages. VLM surveys [224, 225, 226] also discuss various training recipes and data strategies for building VLMs, however, they lack qualitative analysis and do not provide a detailed enough path for training cutting-edge VLMs.

5. Conclusion

As publicly available frontier VLMs continue to approach or even surpass proprietary commercial models, the detailed data strategies of these leading VLMs remains unknown to the community. In this paper, we have unveiled many details on the post-training data strategy for training frontier VLMs. Our covered data strategy is effective and comprehensive. We hope this work offers a transparent practice to inspire the community.

6. Demos

This section provides some examples to demonstrate Eagle2 capabilities. To avoid cherry-picking, we directly select demo cases from other works (Qwen2-VL and InternVL2) as our test cases.

7. Acknowledgments

The team would like to thank the valuable discussions and input from Jihao Liu, Yunze Man, Yue Zhao, Min Shi, Fuxiao Liu, Jindong Jiang, Ming-Chang Chiu, Wonmin Beyon, Jason Lu, Wei Ping, Jon Barker, Zhuolin Yang, Wenliang Dai, Nayeon Lee, Boxin Wang, Mohammad Shoeybi, Tyler Poon, Jim Fan, Yuke Zhu, Danny Yin, Mike Ranzinger, Greg Heinrich, Yonggan Fu, Xin Dong, Shizhe Diao, Pavlo Molchanov, Yi Dong, Shiwei Sheng, Xiaolong Li, Vidya Murali, Parthasarathy Sriram, Song Han, Monika Jhuria, Shubham Agrawal, Wenyuan Zhang, Ke Chen, Yan Wang, Byung-Kwan Lee, Yueh-Hua Wu, Min-Hung Chen, Ryo Hachiuma, Chao-Han Huck Yang, Yao Xu, Osvald Nitski, Elena Lantz, Qing Miao, Ofri Masad, Ofer Baratz, Niket Agarwal, Sriharsha Niverty, and Padmavathy Subramanian.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 2022.

[2] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

[3] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. In *ICLR*, 2022.

[4] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[5] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv:2312.08914*, 2023.

[6] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. DriveVLM: The convergence of autonomous driving and large vision-language models. In *CoRL*, 2024.

[7] Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. LingoQA: Video question answering for autonomous driving. In *ECCV*, 2023.

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv:2307.15818*, 2023.

[9] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv:2303.03378*, 2023.

[10] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv:2406.09246*, 2024.

[11] OpenAI. GPT-4V(ision) system card. `https://cdn.openai.com/papers/GPTV_System_Card.pdf`, 2023.

[12] Anthropic. The claude 3 model family: Opus, sonnet, haiku. `https://www.anthropic.com`, 2024.

[13] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv:2407.10671*, 2024.

[14] OpenGVLab Team. InternVL2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy. `https://internvl.github.io/blog/2024-07-02-InternVL-2.0/`, 2024.

[15] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv:2407.21783*, 2024.

[16] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv:2406.16860*, 2024.

[17] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv:2408.03326*, 2024.

[18] OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. `https://github.com/open-compass/opencompass`, 2023.

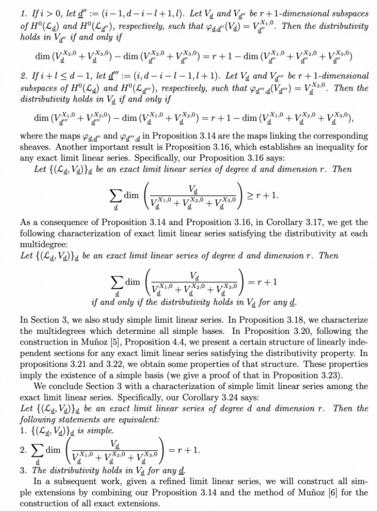
[19] Qiming Zhang, Jing Zhang, Yufei Xu, and Dacheng Tao. Vision transformer with quadrangle attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[20] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv:2308.12966*, 2023.

[21] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv:2404.16821*, 2024.

[22] Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin,

Document Parsing with Dense Formulas (Example borrowed from Qwen2-VL paper)



1. *If $i > 0$, let $\underline{d}'' := (i-1, d-i-l+1, l)$. Let $V_{\underline{d}}$ and $V_{\underline{d}''}$ be $r+1$-dimensional subspaces of $H^0(\mathcal{L}_{\underline{d}})$ and $H^0(\mathcal{L}_{\underline{d}''})$, respectively, such that $\varphi_{\underline{d},\underline{d}''}(V_{\underline{d}}) = V_{\underline{d}''}^{X_1,0}$. Then the distributivity holds in $V_{\underline{d}''}$ if and only if*

$$\dim\left(V_{\underline{d}}^{X_2,0} + V_{\underline{d}}^{X_3,0}\right) - \dim\left(V_{\underline{d}''}^{X_2,0} + V_{\underline{d}''}^{X_3,0}\right) = r + 1 - \dim\left(V_{\underline{d}''}^{X_1,0} + V_{\underline{d}''}^{X_2,0} + V_{\underline{d}''}^{X_3,0}\right)$$

2. *If $i+l \le d-1$, let $\underline{d}''' := (i, d-i-l-1, l+1)$. Let $V_{\underline{d}}$ and $V_{\underline{d}'''}$ be $r+1$-dimensional subspaces of $H^0(\mathcal{L}_{\underline{d}})$ and $H^0(\mathcal{L}_{\underline{d}'''})$, respectively, such that $\varphi_{\underline{d}''',\underline{d}}(V_{\underline{d}'''}) = V_{\underline{d}}^{X_3,0}$. Then the distributivity holds in $V_{\underline{d}}$ if and only if*

$$\dim\left(V_{\underline{d}'''}^{X_1,0} + V_{\underline{d}'''}^{X_2,0}\right) - \dim\left(V_{\underline{d}}^{X_1,0} + V_{\underline{d}}^{X_2,0}\right) = r + 1 - \dim\left(V_{\underline{d}}^{X_1,0} + V_{\underline{d}}^{X_2,0} + V_{\underline{d}}^{X_3,0}\right),$$

where the maps $\varphi_{\underline{d},\underline{d}''}$ and $\varphi_{\underline{d}''',\underline{d}}$ in Proposition 3.14 are the maps linking the corresponding sheaves. Another important result is Proposition 3.16, which establishes an inequality for any exact limit linear series. Specifically, our Proposition 3.16 says:

Let $\{(\mathcal{L}_{\underline{d}}, V_{\underline{d}})\}_{\underline{d}}$ be an exact limit linear series of degree d and dimension r. Then

$$\sum_{\underline{d}} \dim\left(\frac{V_{\underline{d}}}{V_{\underline{d}}^{X_1,0} + V_{\underline{d}}^{X_2,0} + V_{\underline{d}}^{X_3,0}}\right) \ge r + 1.$$

As a consequence of Proposition 3.14 and Proposition 3.16, in Corollary 3.17, we get the following characterization of exact limit linear series satisfying the distributivity at each multidegree:

Let $\{(\mathcal{L}_{\underline{d}}, V_{\underline{d}})\}_{\underline{d}}$ be an exact limit linear series of degree d and dimension r. Then

$$\sum_{\underline{d}} \dim\left(\frac{V_{\underline{d}}}{V_{\underline{d}}^{X_1,0} + V_{\underline{d}}^{X_2,0} + V_{\underline{d}}^{X_3,0}}\right) = r + 1$$

if and only if the distributivity holds in $V_{\underline{d}}$ for any $\underline{d}$.

In Section 3, we also study simple limit linear series. In Proposition 3.18, we characterize the multidegrees which determine all simple bases. In Proposition 3.20, following the construction in Muñoz [5], Proposition 4.4, we present a certain structure of linearly independent sections for any exact limit linear series satisfying the distributivity property. In propositions 3.21 and 3.22, we obtain some properties of that structure. These properties imply the existence of a simple basis (we give a proof of that in Proposition 3.23).

We conclude Section 3 with a characterization of simple limit linear series among the exact limit linear series. Specifically, our Corollary 3.24 says:

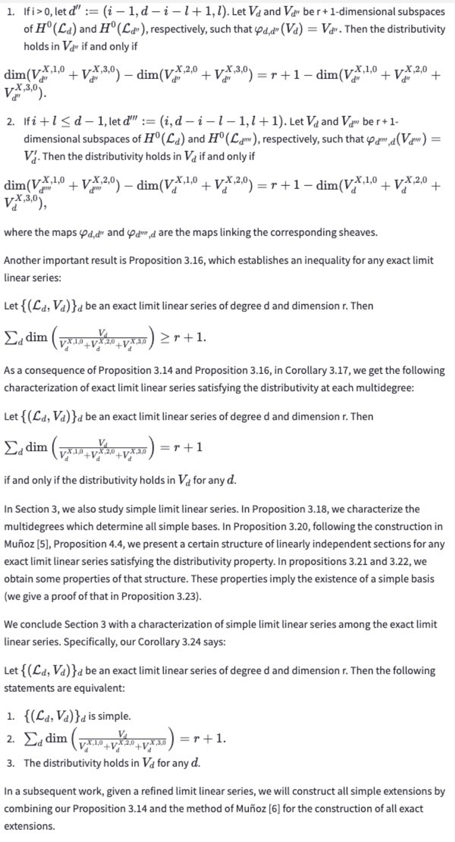
Let $\{(\mathcal{L}_{\underline{d}}, V_{\underline{d}})\}_{\underline{d}}$ be an exact limit linear series of degree d and dimension r. Then the following statements are equivalent:

1. *$\{(\mathcal{L}_{\underline{d}}, V_{\underline{d}})\}_{\underline{d}}$ is simple.*
2. $\sum_{\underline{d}} \dim\left(\frac{V_{\underline{d}}}{V_{\underline{d}}^{X_1,0} + V_{\underline{d}}^{X_2,0} + V_{\underline{d}}^{X_3,0}}\right) = r + 1.$
3. *The distributivity holds in $V_{\underline{d}}$ for any $\underline{d}$.*

In a subsequent work, given a refined limit linear series, we will construct all simple extensions by combining our Proposition 3.14 and the method of Muñoz [6] for the construction of all exact extensions.

Input: Extract the Text content.

Model Response



1. If i > 0, let $d'' := (i-1, d-i-l+1, l)$. Let V_d and $V_{d''}$ be r + 1-dimensional subspaces of $H^0(\mathcal{L}_d)$ and $H^0(\mathcal{L}_{d''})$, respectively, such that $\varphi_{d,d''}(V_d) = V_{d''}$. Then the distributivity holds in $V_{d''}$ if and only if

$$\dim(V_{d''}^{X,1,0} + V_{d''}^{X,3,0}) - \dim(V_{d''}^{X,2,0} + V_{d''}^{X,3,0}) = r + 1 - \dim(V_{d''}^{X,1,0} + V_{d''}^{X,2,0} + V_{d''}^{X,3,0}).$$

2. If $i + l \le d - 1$, let $d''' := (i, d-i-l-1, l+1)$. Let V_d and $V_{d'''}$ be r + 1-dimensional subspaces of $H^0(\mathcal{L}_d)$ and $H^0(\mathcal{L}_{d'''})$, respectively, such that $\varphi_{d''',d}(V_{d'''}) = V'_d$. Then the distributivity holds in V_d if and only if

$$\dim(V_{d'''}^{X,1,0} + V_{d'''}^{X,2,0}) - \dim(V_d^{X,1,0} + V_d^{X,2,0}) = r + 1 - \dim(V_d^{X,1,0} + V_d^{X,2,0} + V_d^{X,3,0}),$$

where the maps $\varphi_{d,d''}$ and $\varphi_{d''',d}$ are the maps linking the corresponding sheaves.

Another important result is Proposition 3.16, which establishes an inequality for any exact limit linear series:

Let $\{(\mathcal{L}_d, V_d)\}_d$ be an exact limit linear series of degree d and dimension r. Then

$$\textstyle\sum_d \dim\left(\frac{V_d}{V_d^{X,1,0} + V_d^{X,2,0} + V_d^{X,3,0}}\right) \ge r + 1.$$

As a consequence of Proposition 3.14 and Proposition 3.16, in Corollary 3.17, we get the following characterization of exact limit linear series satisfying the distributivity at each multidegree:

Let $\{(\mathcal{L}_d, V_d)\}_d$ be an exact limit linear series of degree d and dimension r. Then

$$\textstyle\sum_d \dim\left(\frac{V_d}{V_d^{X,1,0} + V_d^{X,2,0} + V_d^{X,3,0}}\right) = r + 1$$

if and only if the distributivity holds in V_d for any d.

In Section 3, we also study simple limit linear series. In Proposition 3.18, we characterize the multidegrees which determine all simple bases. In Proposition 3.20, following the construction in Muñoz [5], Proposition 4.4, we present a certain structure of linearly independent sections for any exact limit linear series satisfying the distributivity property. In propositions 3.21 and 3.22, we obtain some properties of that structure. These properties imply the existence of a simple basis (we give a proof of that in Proposition 3.23).

We conclude Section 3 with a characterization of simple limit linear series among the exact limit linear series. Specifically, our Corollary 3.24 says:

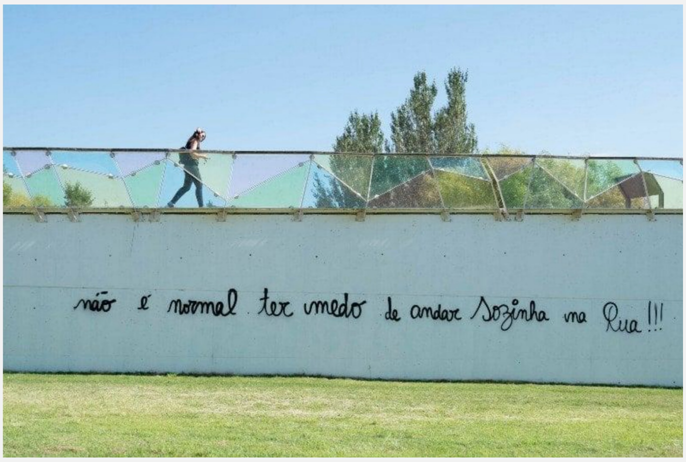
Let $\{(\mathcal{L}_d, V_d)\}_d$ be an exact limit linear series of degree d and dimension r. Then the following statements are equivalent:

1. $\{(\mathcal{L}_d, V_d)\}_d$ is simple.
2. $\sum_d \dim\left(\frac{V_d}{V_d^{X,1,0} + V_d^{X,2,0} + V_d^{X,3,0}}\right) = r + 1.$
3. The distributivity holds in V_d for any d.

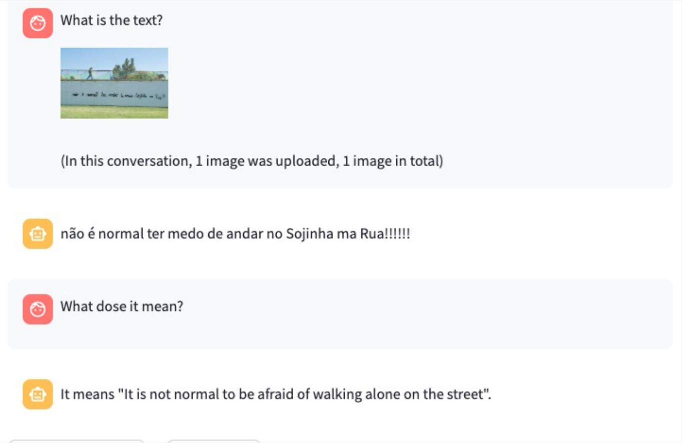
In a subsequent work, given a refined limit linear series, we will construct all simple extensions by combining our Proposition 3.14 and the method of Muñoz [6] for the construction of all exact extensions.

Figure 12 | Eagle2-9B has strong OCR recognition capabilities.

Multilingual Text Recognition (Example borrowed from Qwen2-VL paper)



Input and Model Response



What is the text?

(In this conversation, 1 image was uploaded, 1 image in total)

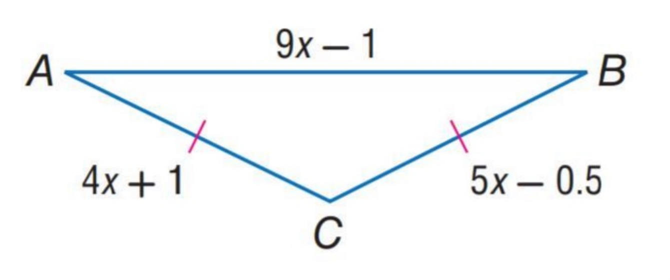
não é normal ter medo de andar no Sojinha ma Rua!!!!!!

What dose it mean?

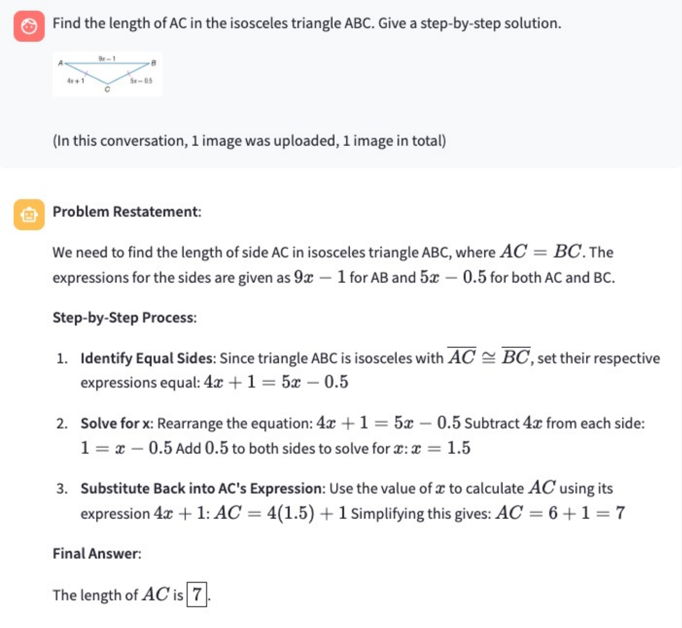
It means "It is not normal to be afraid of walking alone on the street".

Figure 13 | Eagle2-9B has Multilingual OCR Recognition Capability.

Mathematical Problem Solving (Example borrowed from Qwen2-VL paper)



Input and Model Response



Find the length of AC in the isosceles triangle ABC. Give a step-by-step solution.

(In this conversation, 1 image was uploaded, 1 image in total)

Problem Restatement:

We need to find the length of side AC in isosceles triangle ABC, where $AC = BC$. The expressions for the sides are given as $9x - 1$ for AB and $5x - 0.5$ for both AC and BC.

Step-by-Step Process:

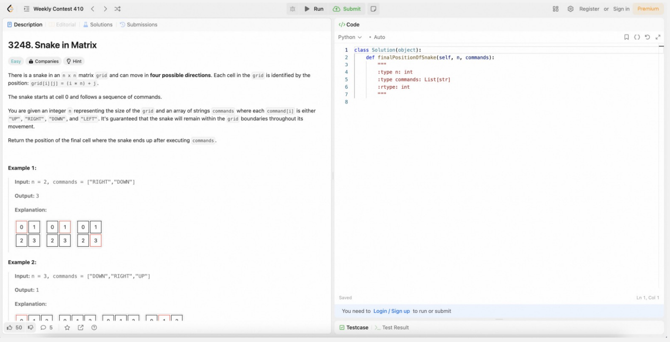
1. **Identify Equal Sides:** Since triangle ABC is isosceles with $\overline{AC} \cong \overline{BC}$, set their respective expressions equal: $4x + 1 = 5x - 0.5$
2. **Solve for x:** Rearrange the equation: $4x + 1 = 5x - 0.5$ Subtract $4x$ from each side: $1 = x - 0.5$ Add 0.5 to both sides to solve for x: $x = 1.5$
3. **Substitute Back into AC's Expression:** Use the value of x to calculate AC using its expression $4x + 1$: $AC = 4(1.5) + 1$ Simplifying this gives: $AC = 6 + 1 = 7$

Final Answer:

The length of AC is $\boxed{7}$.

Figure 14 | CoT Inference ability.

Algorithmic Problem Solving (Example borrowed from Qwen2-VL paper)



3248. Snake in Matrix

There is a snake in an `n x n` matrix `grid` and can move in **four possible directions**. Each cell in the `grid` is identified by the position: `grid[i][j] = (i * n) + j`.

The snake starts at cell 0 and follows a sequence of commands.

You are given an integer `n` representing the size of the `grid` and an array of strings `commands` where each `command[i]` is either `"UP"`, `"RIGHT"`, `"DOWN"`, and `"LEFT"`. It's guaranteed that the snake will remain within the `grid` boundaries throughout its movement.

Return the position of the final cell where the snake ends up after executing `commands`.

Example 1:

Input: `n = 2, commands = ["RIGHT","DOWN"]`

Output: 3

Explanation:

Example 2:

Input: `n = 3, commands = ["DOWN","RIGHT","UP"]`

Output: 1

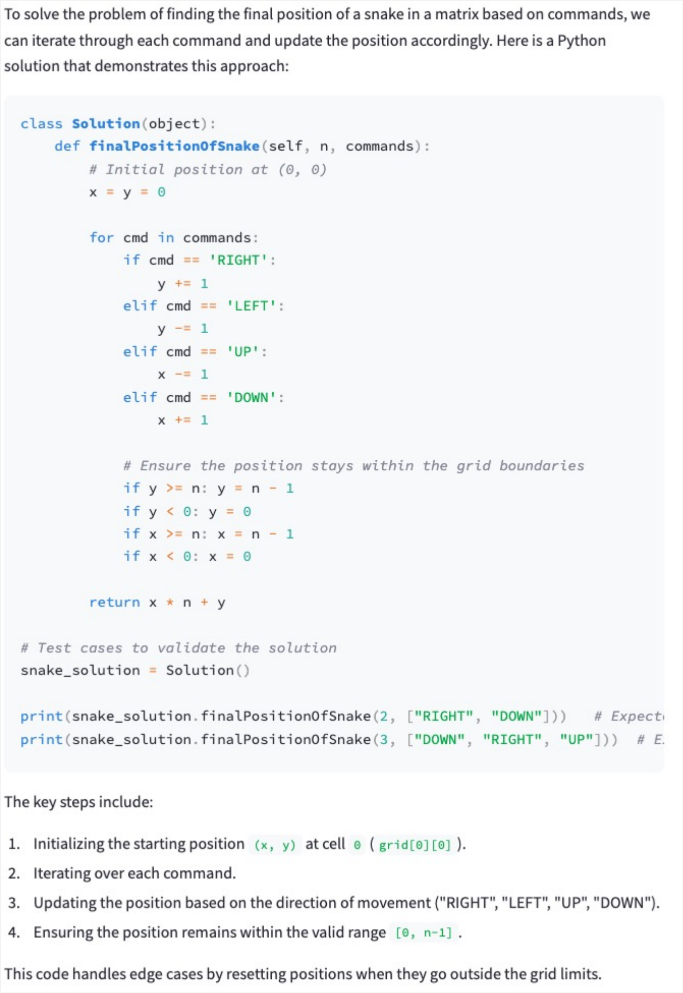
Explanation:

```
class Solution(object):
    def finalPositionOfSnake(self, n, commands):
        """
        :type n: int
        :type commands: List[str]
        :rtype: int
        """
```

You need to Login / Sign up to run or submit

Input: Solve the algorithmic problem in the web page.

Model Response



To solve the problem of finding the final position of a snake in a matrix based on commands, we can iterate through each command and update the position accordingly. Here is a Python solution that demonstrates this approach:

```
class Solution(object):
    def finalPositionOfSnake(self, n, commands):
        # Initial position at (0, 0)
        x = y = 0

        for cmd in commands:
            if cmd == 'RIGHT':
                y += 1
            elif cmd == 'LEFT':
                y -= 1
            elif cmd == 'UP':
                x -= 1
            elif cmd == 'DOWN':
                x += 1

            # Ensure the position stays within the grid boundaries
            if y >= n: y = n - 1
            if y < 0: y = 0
            if x >= n: x = n - 1
            if x < 0: x = 0

        return x * n + y

# Test cases to validate the solution
snake_solution = Solution()

print(snake_solution.finalPositionOfSnake(2, ["RIGHT", "DOWN"]))   # Expect
print(snake_solution.finalPositionOfSnake(3, ["DOWN", "RIGHT", "UP"]))  # E
```

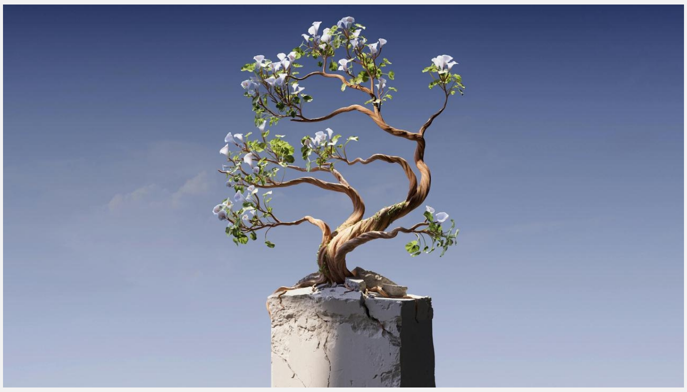
The key steps include:

1. Initializing the starting position `(x, y)` at cell `0` (`grid[0][0]`).
2. Iterating over each command.
3. Updating the position based on the direction of movement ("RIGHT", "LEFT", "UP", "DOWN").
4. Ensuring the position remains within the valid range `[0, n-1]`.

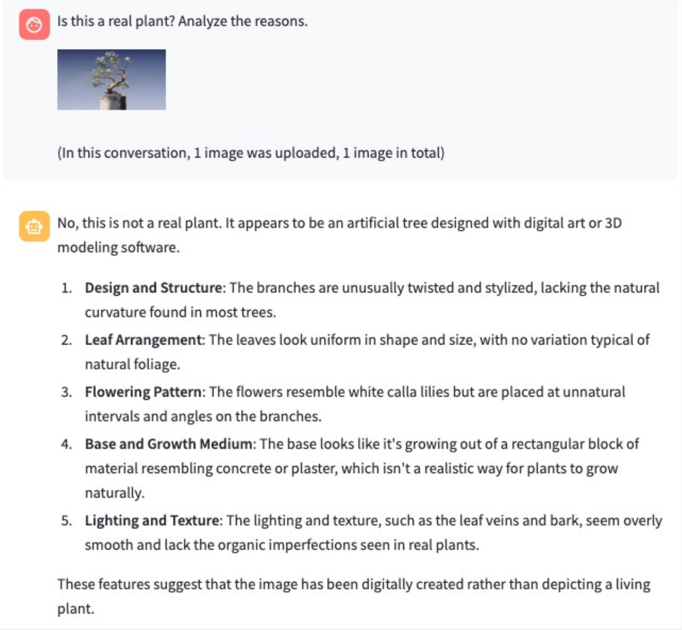
This code handles edge cases by resetting positions when they go outside the grid limits.

Figure 15 | Solving “Easy" algorithmic problem via Eagle2-9B.

Image Analysis (Example borrowed from InternVL2 demos)



Input and Model Response



Is this a real plant? Analyze the reasons.

(In this conversation, 1 image was uploaded, 1 image in total)

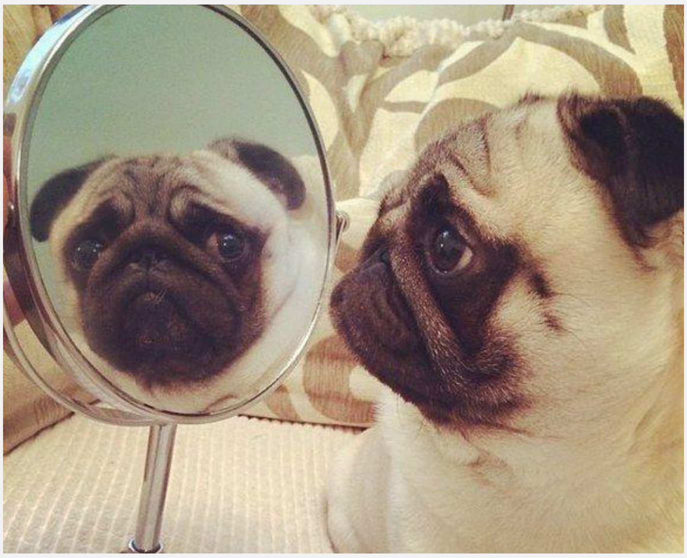
No, this is not a real plant. It appears to be an artificial tree designed with digital art or 3D modeling software.

1. **Design and Structure**: The branches are unusually twisted and stylized, lacking the natural curvature found in most trees.
2. **Leaf Arrangement**: The leaves look uniform in shape and size, with no variation typical of natural foliage.
3. **Flowering Pattern**: The flowers resemble white calla lilies but are placed at unnatural intervals and angles on the branches.
4. **Base and Growth Medium**: The base looks like it's growing out of a rectangular block of material resembling concrete or plaster, which isn't a realistic way for plants to grow naturally.
5. **Lighting and Texture**: The lighting and texture, such as the leaf veins and bark, seem overly smooth and lack the organic imperfections seen in real plants.

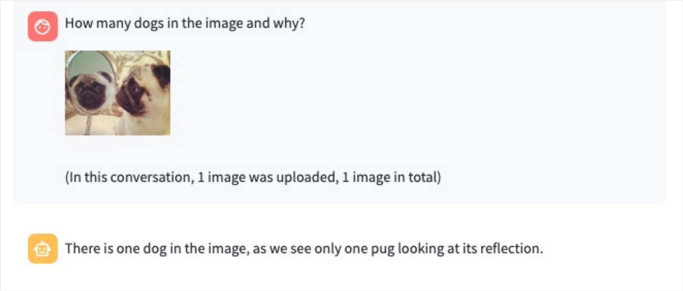
These features suggest that the image has been digitally created rather than depicting a living plant.

Figure 16 | Eagle2-9B is able to analyze anomalies in images.

Image Analysis (Example borrowed from InternVL2 demos)



Input and Model Response



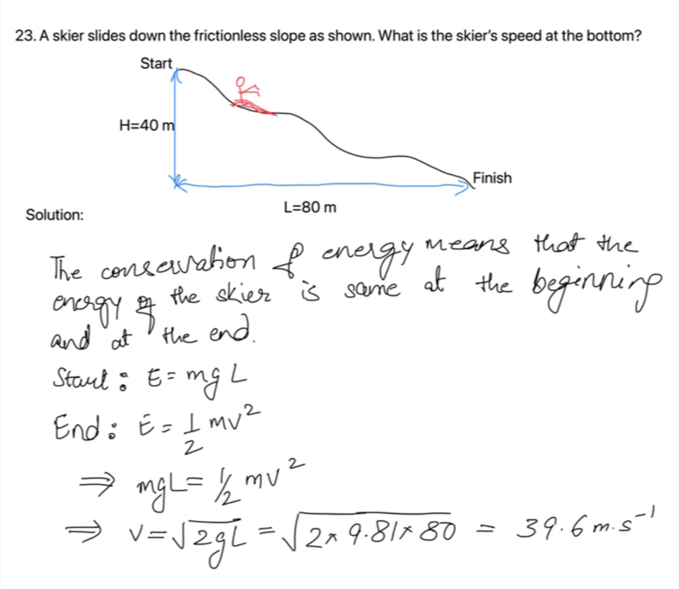
How many dogs in the image and why?

(In this conversation, 1 image was uploaded, 1 image in total)

There is one dog in the image, as we see only one pug looking at its reflection.

Figure 17 | Eagle2-9B can correctly distinguish objects in the mirror.

Handwritting OCR (Example borrowed from InternVL2 demos)



23. A skier slides down the frictionless slope as shown. What is the skier's speed at the bottom?

Start
H=40 m
Finish
L=80 m

Solution:

The conservation of energy means that the energy of the skier is same at the beginning and at the end.

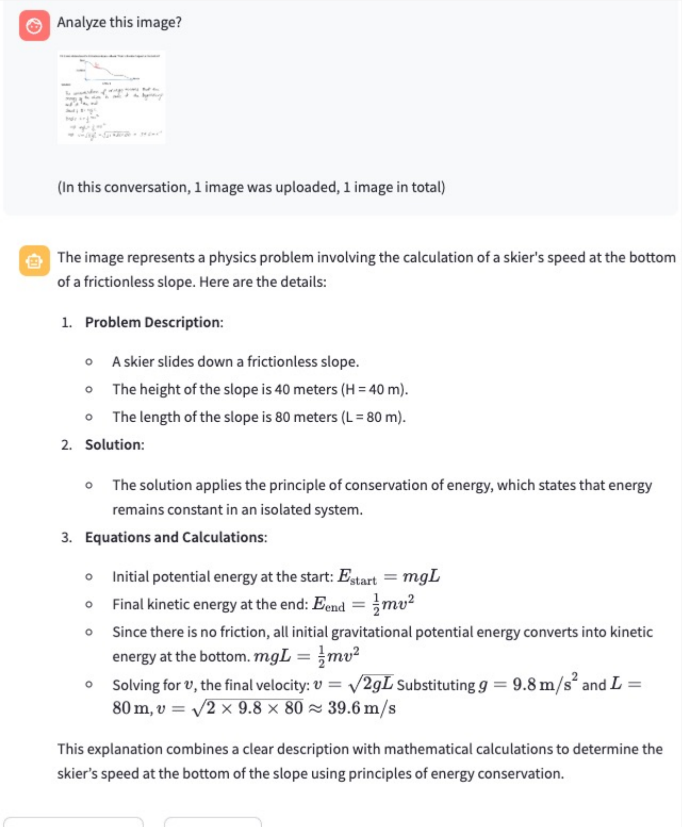
Start : $E = mgL$

End : $E = \frac{1}{2} mv^2$

$\Rightarrow mgL = \frac{1}{2} mv^2$

$\Rightarrow v = \sqrt{2gL} = \sqrt{2 \times 9.81 \times 80} = 39.6\ m.s^{-1}$

Input and Model Response



Analyze this image?

(In this conversation, 1 image was uploaded, 1 image in total)

The image represents a physics problem involving the calculation of a skier's speed at the bottom of a frictionless slope. Here are the details:

1. **Problem Description**:
   - A skier slides down a frictionless slope.
   - The height of the slope is 40 meters (H = 40 m).
   - The length of the slope is 80 meters (L = 80 m).
2. **Solution**:
   - The solution applies the principle of conservation of energy, which states that energy remains constant in an isolated system.
3. **Equations and Calculations**:
   - Initial potential energy at the start: $E_{\text{start}} = mgL$
   - Final kinetic energy at the end: $E_{\text{end}} = \frac{1}{2}mv^2$
   - Since there is no friction, all initial gravitational potential energy converts into kinetic energy at the bottom. $mgL = \frac{1}{2}mv^2$
   - Solving for v, the final velocity: $v = \sqrt{2gL}$ Substituting $g = 9.8\,\mathrm{m/s^2}$ and $L = 80\,\mathrm{m}$, $v = \sqrt{2 \times 9.8 \times 80} \approx 39.6\,\mathrm{m/s}$

This explanation combines a clear description with mathematical calculations to determine the skier's speed at the bottom of the slope using principles of energy conservation.

Figure 18 | Eagle2-9B has excellent handwriting recognition capabilities.

Karan Sapra, Yaser Yacoob, Humphrey Shi, Bryan Catanzaro, Andrew Tao, Jan Kautz, Zhiding Yu, and Guilin Liu. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv:2408.15998*, 2024.

[23] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[24] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

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

[26] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv:2303.08774*, 2023.

[27] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[28] OpenGVLab. Sharegpt-4o dataset. `https://huggingface.co/datasets/OpenGVLab/ShareGPT-4o`, 2024.

[29] Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. Kvqa: Knowledge-aware visual question answering. In *AAAI*, 2019.

[30] skvarre. Movie posters-100k dataset. `https://huggingface.co/datasets/skvarre/movie_posters-100k`, 2024.

[31] Tobias Weyand, André Araujo, Bingyi Cao, and Jack Sim. Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *CVPR*, 2020.

[32] HugGAN. Wikiart dataset. `https://huggingface.co/datasets/huggan/wikiart`, 2024.

[33] Chengqian Ma, Zhanxiang Hua, Alexandra Anderson-Frey, Vikram Iyer, Xin Liu, and Lianhui Qin. Weatherqa: Can multimodal language models reason about severe weather? *arXiv:2406.11217*, 2024.

[34] hazal karakus. mscoco-controlnet-canny-less-colors dataset. `https://huggingface.co/datasets/hazal-karakus/mscoco-controlnet-canny-less-colors`, 2024.

[35] Emile Esmaili. sheet music clean ataset. `https://huggingface.co/datasets/EmileEsmaili/sheet_music_clean`, 2024.

[36] Youngjoon Yu, Sangyun Chung, Byung-Kwan Lee, and Yong Man Ro. Spark: Multi-vision sensor perception and reasoning benchmark for large-scale vision-language models. *arXiv:2408.12114*, 2024.

[37] Renjie Pi, Jianshu Zhang, Jipeng Zhang, Rui Pan, Zhekai Chen, and Tong Zhang. Image textualization: An automatic framework for creating accurate and detailed image descriptions. *arXiv:2406.07502*, 2024.

[38] PixArt-alpha. Sam-llava-captions10m dataset. `https://huggingface.co/datasets/PixArt-alpha/SAM-LLaVA-Captions10M`, 2024.

[39] Ashraq. Tmdb-celeb-10k dataset. `https://huggingface.co/datasets/ashraq/tmdb-celeb-10k`, 2024.

[40] Jie Cao and Jing Xiao. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *COLING*, 2022.

[41] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap your own mathematical questions for large language models. *arXiv:2309.12284*, 2023.

[42] Adam Dahlgren Lindström and Savitha Sam Abraham. Clevr-math: A dataset for compositional language, visual and mathematical reasoning. *arXiv:2208.05358*, 2022.

[43] Zhuowan Li, Xingrui Wang, Elias Stengel-Eskin, Adam Kortylewski, Wufei Ma, Benjamin Van Durme, and Alan L Yuille. Super-clevr: A virtual benchmark to diagnose domain robustness in visual reasoning. In *CVPR*, 2023.

[44] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv:2105.04165*, 2021.

[45] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Yichi Zhang, Ziyu Guo, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, Shanghang Zhang, et al. Mavis: Mathematical visual instruction tuning. *arXiv:2407.08739*, 2024.

[46] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv:2105.04165*, 2021.

[47] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning. In *CVPR*, 2019.

[48] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. Solving geometry problems: Combining text and diagram interpretation. In *EMNLP*, 2015.

[49] Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression. *arXiv:2212.02746*, 2022.

[50] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.

[51] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *NeurIPS*, 2022.

[52] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi.

Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*, 2017.

[53] Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. Pathvqa: 30000+ questions for medical visual question answering. *arXiv:2003.10286*, 2020.

[54] Sören Auer, Dante AC Barone, Cassiano Bartz, Eduardo G Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, et al. The sciqa scientific question answering benchmark for scholarly knowledge. *Scientific Reports*, 13(1):7240, 2023.

[55] Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):1–10, 2018.

[56] TIGER-Lab. Visualwebinstruct dataset. `https://huggingface.co/datasets/TIGER-Lab/VisualWebInstruct`, 2024.

[57] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *ACL*, 2022.

[58] Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. MMC: Advancing multimodal chart understanding with large-scale instruction tuning. *arXiv:2311.10774*, 2023.

[59] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *CVPR*, 2018.

[60] Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. PlotQA: Reasoning over scientific plots. In *WACV*, 2020.

[61] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv:2306.14565*, 2023.

[62] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv:2209.14610*, 2022.

[63] Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. UniChart: A universal vision-language pretrained model for chart comprehension and reasoning. *arXiv:2305.14761*, 2023.

[64] Benny J Tang, Angie Boggust, and Arvind Satyanarayan. Vistext: A benchmark for semantically rich chart captioning. *arXiv:2307.05356*, 2023.

[65] Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. Towards complex document understanding by discrete reasoning. In *ACMMM*, 2022.

[66] VQAonDB. Vqaondb dataset. `https://ilocr.iiit.ac.in/vqabd/`.

[67] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning. *arXiv:1710.07300*, 2017.

[68] Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv:2203.06486*, 2022.

[69] Yilun Zhao, Chen Zhao, Linyong Nan, Zhenting Qi, Wenlin Zhang, Xiangru Tang, Boyu Mi, and Dragomir Radev. Robut: A systematic study of table qa robustness against human-annotated adversarial perturbations. *arXiv:2306.14321*, 2023.

[70] Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. *arXiv:2206.01347*, 2022.

[71] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. OCR-Free document understanding transformer. In *ECCV*, 2022.

[72] Mengchao He, Yuliang Liu, Zhibo Yang, Sheng Zhang, Canjie Luo, Feiyu Gao, Qi Zheng, Yongpan Wang, Xin Zhang, and Lianwen Jin. Icpr2018 contest on robust reading for multi-type web images. In *ICPR*, 2018.

[73] Yipeng Sun, Zihan Ni, Chee-Kheng Chng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, et al. Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In *ICDAR*, 2019.

[74] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. Icdar 2019 robust reading challenge on scanned receipts ocr and information extraction. In *International conference on document analysis recognition*, 2019.

[75] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *ICDAR Workshops*, 2019.

[76] OleehyO. Latex formulas dataset. `https://huggingface.co/datasets/OleehyO/latex-formulas`, 2024.

[77] U-V Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International journal on document analysis and recognition*, 5:39–46, 2002.

[78] aidapearson. Aida calculus math handwriting recognition dataset. `https://www.kaggle.com/datasets/aidapearson/ocr-data`, 2023.

[79] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, et al. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In *ICDAR*, 2019.

[80] Tai-Ling Yuan, Zhe Zhu, Kun Xu, Cheng-Jun Li, Tai-Jiang Mu, and Shi-Min Hu. A large chinese text dataset in the wild. *Journal of Computer Science and Technology*, 34:509–521, 2019.

[81] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *ICDAR*, 2019.

[82] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. COCO-Text: Dataset and benchmark for text detection and recognition in natural images. *arXiv:1601.07140*, 2016.

[83] Wenwen Yu, Chengquan Zhang, Haoyu Cao, Wei Hua, Bohan Li, Huang Chen, Mingyu Liu, Mingrui Chen, Jianfeng Kuang, Mengjun Cheng, et al. Icdar 2023 competition on structured text extraction from visually-rich document images. In *ICDAR*, 2023.

[84] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. ICDAR 2023 competition on hierarchical text detection and recognition. In *ICDAR*, 2023.

[85] George Tom, Minesh Mathew, Sergi Garcia-Bordils, Dimosthenis Karatzas, and CV Jawahar. ICDAR 2023 competition on roadtext video text detection, tracking and recognition. In *ICDAR*, 2023.

[86] Zekun Li, Yijun Lin, Yao-Yi Chiang, Jerod Weinman, Solenn Tual, Joseph Chazalon, Julien Perret, Bertrand Duménieu, and Nathalie Abadie. ICDAR 2024 competition on historical map text detection, recognition, and linking. In *ICDAR*, 2024.

[87] parasam. Captcha dataset. `https://www.kaggle.com/datasets/parsasam/captcha-dataset`, 2024.

[88] Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *CVPR*, 2020.

[89] TAL. Tal open dataset. `https://ai.100tal.com/dataset`, 2023.

[90] Praveen Krishnan, Rama Kovvuri, Guan Pang, Boris Vassilev, and Tal Hassner. Textstylebrush: transfer of text aesthetics from a single example. *IEEE Trans. PAMI*, 45(7):9122–9134, 2023.

[91] Markus Diem, Stefan Fiel, Florian Kleber, Robert Sablatnig, Jose M Saavedra, David Contreras, Juan Manuel Barrios, and Luiz S Oliveira. Icfhr 2014 competition on handwritten digit string recognition in challenging datasets (hdsrc 2014). In *International Conference on Frontiers in Handwriting Recognition*, 2014.

[92] mychen76. Invoices and receipts ocr v1 dataset. `https://huggingface.co/datasets/mychen76/invoices-and-receipts_ocr_v1`, 2024.

[93] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions. In *International Conference on Frontiers in Handwriting Recognition*, 2016.

[94] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012.

[95] Sathyanarayanan Ramamoorthy, Nethra Gunti, Shreyash Mishra, S Suryavardan, Aishwarya Reganti, Parth Patwa, Amitava DaS, Tanmoy Chakraborty, Amit Sheth, Asif Ekbal, et al. Memotion 2: Dataset on sentiment and emotion analysis of memes. In *De-Factify: Workshop on Multimodal Fact Checking and Hate Speech Detection, CEUR*, 2022.

[96] Azu. Handwritten-mathematical-expression-convert-latex. `https://huggingface.co/datasets/Azu/Handwritten-Mathematical-Expression-Convert-LaTeX`, 2023.

[97] Xudong Xie, Ling Fu, Zhifei Zhang, Zhaowen Wang, and Xiang Bai. Toward understanding wordart: Corner-guided transformer for scene text recognition. In *ECCV*, 2022.

[98] wendlerc. Renderedtext dataset. `https://huggingface.co/datasets/wendlerc/RenderedText`, 2024.

[99] ift. Handwriting forms dataset. `https://huggingface.co/datasets/ift/handwriting_forms`, 2024.

[100] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehension. In *ACL*, 2018.

[101] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. InfographicVQA. In *WACV*, 2022.

[102] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019.

[103] Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. *arXiv:2403.00231*, 2024.

[104] Yu-Chung Hsiao, Fedir Zubach, Gilles Baechler, Victor Carbune, Jason Lin, Maria Wang, Srinivas Sunkara, Yun Zhu, and Jindong Chen. Screenqa: Large-scale question-answer pairs over mobile app screenshots. *arXiv:2209.08199*, 2022.

[105] Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv:2403.12895*, 2024.

[106] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *arXiv:2310.05126*, 2023.

[107] Hamed Rahimi Sujet AI, Allaa Boutaleb. Sujet-finance-qa-vision-100k: A large-scale dataset for financial document vqa. https://huggingface.co/datasets/sujet-ai/Sujet-Finance-QA-Vision-100k, 2024.

[108] Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions. *arXiv:2408.12637*, 2024.

[109] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge. In *ECCV*, 2022.

[110] Kamizuru00. Diagram image to text dataset. https://huggingface.co/datasets/Kamizuru00/diagram_image_to_text, 2024.

[111] Shuaichen Chang, David Palzer, Jialin Li, Eric Fosler-Lussier, and Ningchuan Xiao. Mapqa: A dataset for question answering on choropleth maps. *arXiv:2211.08545*, 2022.

[112] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. OCR-VQA: Visual question answering by reading text in images. In *ICDAR*, 2019.

[113] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, 2019.

[114] Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images. In *AAAI*, 2023.

[115] Yihao Ding, Siwen Luo, Hyunsuk Chung, and Soyeon Caren Han. Vqa: A new dataset for real-world vqa on pdf documents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2023.

[116] Ibrahim Souleiman Mahamoud, Mickaël Coustaty, Aurélie Joseph, Vincent Poulain d'Andecy, and Jean-Marc Ogier. Chic: Corporate document for visual question answering. In *ICDAR*, 2024.

[117] shreyanshu09. Block diagram dataset. https://huggingface.co/datasets/shreyanshu09/Block_Diagram, 2024.

[118] Jingqun Tang, Qi Liu, Yongjie Ye, Jinghui Lu, Shu Wei, Chunhui Lin, Wanqing Li, Mohamad Fitri Faiz Bin Mahmood, Hao Feng, Zhen Zhao, et al. Mtvqa: Benchmarking multilingual text-centric visual question answering. *arXiv:2405.11985*, 2024.

[119] Manuel Faysse, Hugues Sibille, Tony Wu, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. *arXiv:2407.01449*, 2024.

[120] Minesh Mathew, Lluis Gomez, Dimosthenis Karatzas, and CV Jawahar. Asking questions on handwritten document collections. *International Journal on Document Analysis and Recognition*, 24(3):235–249, 2021.

[121] Manoj Acharya, Kushal Kafle, and Christopher Kanan. TallyQA: Answering complex counting questions. In *AAAI*, 2019.

[122] Haoqin Tu, Chenhang Cui, Zijun Wang, Yiyang Zhou, Bingchen Zhao, Junlin Han, Wangchunshu Zhou, Huaxiu Yao, and Cihang Xie. How many unicorns are in this image? a safety evaluation benchmark for vision llms. *arXiv:2311.16101*, 2023.

[123] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016.

[124] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.

[125] Longtao Zheng, Zhiyuan Huang, Zhenghai Xue, Xinrun Wang, Bo An, and Shuicheng Yan. AgentStudio: A toolkit for building general virtual agents. *arXiv:2403.17918*, 2024.

[126] Junke Wang, Lingchen Meng, Zejia Weng, Bo He, Zuxuan Wu, and Yu-Gang Jiang. To see is to believe: Prompting gpt-4v for better visual instruction tuning. *arXiv:2311.07574*, 2023.

[127] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv:2402.11684*, 2024.

[128] LAION. Gpt-4v dataset. https://huggingface.co/datasets/laion/gpt4v-dataset, 2023.

[129] Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. LLaVAR: Enhanced visual instruction tuning for text-rich image understanding. *arXiv:2306.17107*, 2023.

[130] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. VizWiz Grand Challenge: Answering visual questions from blind people. In *CVPR*, 2018.

[131] Sungguk Cha, Jusung Lee, Younghyun Lee, and Cheoljong Yang. Visually dehallucinative instruction generation. In *ICASSP*, 2024.

[132] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *ECCV*, 2020.

[133] FastJobs. Visual emotional analysis dataset. https://huggingface.co/datasets/FastJobs/Visual_Emotional_Analysis, 2024.

[134] Kaiyu Yang, Olga Russakovsky, and Jia Deng. Spatialsense: An adversarially crowdsourced benchmark for spatial relation recognition. In *ICCV*, 2019.

[135] keremberke. Indoor scene classification dataset. https://huggingface.co/datasets/keremberke/indoor-scene-classification, 2024.

[136] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. PAMI*, 40(6):1452–1464, 2017.

[137] Yangzhou Liu, Yue Cao, Zhangwei Gao, Weiyun Wang, Zhe Chen, Wenhai Wang, Hao Tian, Lewei Lu, Xizhou Zhu, Tong Lu, et al. Mminstruct: A high-quality multimodal instruction tuning dataset with extensive diversity. *arXiv:2407.15838*, 2024.

[138] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv:2312.14150*, 2023.

[139] Abhilash Nandy, Yash Agarwal, Ashish Patwa, Millon Madhur Das, Aman Bansal, Ankit Raj, Pawan Goyal, and Niloy Ganguly. Yesbut: A high-quality annotated multimodal dataset for evaluating satire comprehension capability of vision-language models. *arXiv:2409.13592*, 2024.

[140] Yujie Lu, Dongfu Jiang, Wenhu Chen, William Yang Wang, Yejin Choi, and Bill Yuchen Lin. Wildvision: Evaluating vision-language models in the wild with human preferences. *arXiv:2406.11069*, 2024.

[141] Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. Llava-critic: Learning to evaluate multimodal models. *arXiv:2410.02712*, 2024.

[142] Tianyu Yu, Haoye Zhang, Yuan Yao, Yunkai Dang, Da Chen, Xiaoman Lu, Ganqu Cui, Taiwen He, Zhiyuan Liu, Tat-Seng Chua, et al. Rlaif-v: Aligning mllms through open-source ai feedback for super gpt-4v trustworthiness. *arXiv:2405.17220*, 2024.

[143] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

[144] Siwei Wu, Kang Zhu, Yu Bai, Yiming Liang, Yizhi Li, Haoning Wu, Jiaheng Liu, Ruibo Liu, Xingwei Qu, Xuxin Cheng, et al. Mmra: A benchmark for multi-granularity multi-image relational association. *arXiv:2407.17379*, 2024.

[145] Vlad Hosu, Hanhe Lin, Tamas Sziranyi, and Dietmar Saupe. KonIQ-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Trans. Image Processing*, 29:4041–4056, 2020.

[146] Ziyu Liu, Tao Chu, Yuhang Zang, Xilin Wei, Xiaoyi Dong, Pan Zhang, Zijian Liang, Yuanjun Xiong, Yu Qiao, Dahua Lin, et al. Mmdu: A multi-turn multi-image dialog understanding benchmark and instruction-tuning dataset for lvlms. *arXiv:2406.11833*, 2024.

[147] Harsh Jhamtani and Taylor Berg-Kirkpatrick. Learning to describe differences between pairs of similar images. *arXiv:1808.10584*, 2018.

[148] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *NeurIPS*, 2020.

[149] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. *NeurIPS*, 2015.

[150] Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *ACL*, 2017.

[151] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv:2405.02246*, 2024.

[152] Jonas Belouadi, Anne Lauscher, and Steffen Eger. Automatikz: Text-guided synthesis of scientific vector graphics with tikz. *arXiv:2310.00367*, 2023.

[153] LLM-Red-Team Contributors. emo-visual-data: Emotion and visual data analysis project. `https://github.com/LLM-Red-Team/emo-visual-data`, 2024.

[154] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. IconQA: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv:2110.13214*, 2021.

[155] Hugo Laurençon, Léo Tronchon, and Victor Sanh. Unlocking the conversion of web screenshots into html code with the websight dataset. *arXiv:2403.09029*, 2024.

[156] W Lian, B Goodson, E Pentland, et al. OpenOrca: An open dataset of gpt augmented flan reasoning traces, 2023.

[157] Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-Math: Unlocking the potential of slms in grade school math. *arXiv:2402.14830*, 2024.

[158] Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. OpenCodeInterpreter: Integrating code generation with execution and refinement. *arXiv:2402.14658*, 2024.

[159] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv:2309.05653*, 2023.

[160] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv:2304.12244*, 2023.

[161] Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. In *EMNLP*, 2023.

[162] Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants. `https://huggingface.co/datasets/teknium/OpenHermes-2.5`, 2023.

[163] Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. `https://huggingface.co/datasets/AI-MO/NuminaMath-CoT`, 2024.

[164] FLOCK4H. Python codes 25k dataset. `https://huggingface.co/datasets/flytech/python-codes-25k`, 2024.

[165] BAAI. Infinity-instruct dataset. `https://huggingface.co/datasets/BAAI/Infinity-Instruct`, 2024.

[166] Tarun Bisht. Python code instructions 18k alpaca dataset. `https://huggingface.co/datasets/iamtarun/python_code_instructions_18k_alpaca`, 2024.

[167] LooksJuicy. Ruozhiba dataset. `https://huggingface.co/datasets/LooksJuicy/ruozhiba`, 2024.

[168] Bo-Wen Zhang, Yan Yan, Lin Li, and Guang Liu. Infinitymath: A scalable instruction tuning dataset in programmatic mathematical reasoning. In *ACM International Conference on Information and Knowledge Management*, 2024.

[169] Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-DPO: Step-wise preference optimization for long-chain reasoning of llms. *arXiv:2406.18629*, 2024.

[170] Xiaokang Zhang, Jing Zhang, Zeyao Ma, Yang Li, Bohan Zhang, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, et al. TableLLM: Enabling tabular data manipulation by llms in real office usage scenarios. *arXiv:2403.19318*, 2024.

[171] Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. Advancing llm reasoning generalists with preference trees. *arXiv:2404.02078*, 2024.

[172] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[173] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *ECCV*, 2020.

[174] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv:2311.12793*, 2023.

[175] Xiaotong Li, Fan Zhang, Haiwen Diao, Yueze Wang, Xinlong Wang, and Ling-Yu Duan. Densefusion-1m: Merging vision experts for comprehensive multimodal perception. *arXiv:2407.08303*, 2024.

[176] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019.

[177] Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *arXiv:2402.10176*, 2024.

[178] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *CVPR*, 2022.

[179] Sentence-Transformers. All-minilm-l6-v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2, 2023.

[180] Yuliang Liu, Zhang Li, Hongliang Li, Wenwen Yu, Mingxin Huang, Dezhi Peng, Mingyu Liu, Mingrui Chen, Chunyuan Li, Lianwen Jin, et al. On the hidden mystery of ocr in large multimodal models. *arXiv:2305.07895*, 2023.

[181] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. LlamaFactory: Unified efficient fine-tuning of 100+ language models. *arXiv:2403.13372*, 2024.

[182] Mario Michael Krell, Matej Kosec, Sergio P Perez, and Andrew Fitzgibbon. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance. *arXiv:2107.02027*, 2021.

[183] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023.

[184] OpenAI. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/, 2023. Accessed: 2024-11-12.

[185] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024.

[186] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv:2404.06395*, 2024.

[187] OpenAI. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt, 2023. Accessed: 2023.

[188] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022.

[189] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[190] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*, 2024.

[191] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. CogVLM: Visual expert for pretrained language models. *arXiv:2311.03079*, 2023.

[192] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv:2305.03726*, 2023.

[193] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[194] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. MiniCPM-V: A gpt-4v level mllm on your phone. *arXiv:2408.01800*, 2024.

[195] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim

Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv:2407.07726*, 2024.

[196] Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuoling Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NVLM: Open frontier-class multimodal llms. *arXiv:2409.11402*, 2024.

[197] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. VILA: On pre-training for visual language models, 2023.

[198] Yunhao Fang, Ligeng Zhu, Yao Lu, Yan Wang, Pavlo Molchanov, Jan Kautz, Jang Hyun Cho, Marco Pavone, Song Han, and Hongxu Yin. VILA2: Vila augmented vila. *arXiv:2407.17453*, 2024.

[199] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. NVILA: Efficient frontier visual language models. *arXiv:2412.04468*, 2024.

[200] IDEFICS. Introducing idefics: An open reproduction of state-of-the-art visual language model. https://huggingface.co/blog/idefics, 2023.

[201] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[202] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip. Zenodo. Version 0.1. https://doi.org/10.5281/zenodo.5143773, 2021. DOI: 10.5281/zenodo.5143773.

[203] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-Clip: Improved training techniques for clip at scale. *arXiv:2303.15389*, 2023.

[204] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv:2312.14238*, 2023.

[205] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *CVPR*, 2024.

[206] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. AM-RADIO: Agglomerative vision foundation model reduce all domains into one. In *CVPR*, 2024.

[207] Greg Heinrich, Mike Ranzinger, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, Pavlo Molchanov, et al. RADIO Amplified: Improved baselines for agglomerative vision foundation models. *arXiv:2412.07679*, 2024.

[208] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv:2311.07575*, 2023.

[209] Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. Prismer: A vision-language model with an ensemble of experts. *TMLR*, 2024.

[210] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: Investigating the design space of visually-conditioned language models. *arXiv:2402.07865*, 2024.

[211] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv:2403.18814*, 2024.

[212] Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Feast your eyes: Mixture-of-resolution adaptation for multimodal large language models. *arXiv:2403.03003*, 2024.

[213] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. PaLI-X: On scaling up a multilingual vision and language model. *arXiv:2305.18565*, 2023.

[214] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. PaLI-3 vision language models: Smaller, faster, stronger. *arXiv:2310.09199*, 2023.

[215] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv:2311.06607*, 2023.

[216] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *arXiv:2403.11703*, 2024.

[217] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. InternLM-XComposer2-4KHD: A pioneering large vision-language model handling resolutions from 336 pixels to 4K HD. *arXiv:2404.06512*, 2024.

[218] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv:2310.03744*, 2023.

[219] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *NeurIPS*, 36, 2024.

[220] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. MIMIC-IT: Multi-modal in-context instruction tuning. 2023.

[221] Qingyun Li, Zhe Chen, Weiyun Wang, Wenhai Wang, Shenglong Ye, Zhenjiang Jin, et al. OmniCorpus: A unified multimodal corpus of 10 billion-level images interleaved with text. *arXiv:2406.08418*, 2024.

[222] Anas Awadalla, Le Xue, Oscar Lo, Manli Shu, Hannah Lee, Etash Kumar Guha, Matt Jordan, Sheng Shen, Mohamed Awadalla, Silvio Savarese, Caiming Xiong, Ran Xu, Yejin Choi, and Ludwig Schmidt. MINT-1T: Scaling open-source multimodal data by 10x: A multimodal dataset with one trillion tokens. 2024.

[223] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv:2409.17146*, 2024.

[224] Tianyi Bai, Hao Liang, Binwang Wan, Yanran Xu, Xi Li, Shiyu Li, Ling Yang, Bozhou Li, Yifan Wang, Bin Cui, et al. A survey of multimodal large language model from a data-centric perspective. *arXiv:2405.16640*, 2024.

[225] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv:2306.13549*, 2023.

[226] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and S Yu Philip. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data*, 2023.

A. Computing

We show our training resource for Eagle2-9B in Tab. A. In actual development, we rarely iterate the Stage-1 model. Usually, we iterate Stage-1.5 once after iterating Stage-2 $>$ 10 times.

		Stage-1	Stage-1.5	Stage-2
Qwen-2-9B	**GPUs**	H100×128	H100×256	H100×256
	Hours	2.5	28	6

Table A | Training Information of Eagle2-9B.

Dataset	Category	#Stage-1.5	#Stage-2
LLaVa-150K-EN	General	158K	57K
LLaVa-150K-CN	General	158K	50K
LVIS-Instruct4V	General	223K	12K
ALLaVa-laion	General	505K	20K
ALLaVa-vflan	General	202K	26K
Laion-GPT4V	General	11K	11K
LLAVAR	General	20K	20K
SketchyVQA	General	4K	4K
IDK	General	11K	11K
AlfworldGPT	General	45K	9K
LNQA	General	303K	23K
Face-Emotion	General	1K	1K
SpatialSense	General	10K	10K
Indoor-QA	General	3K	3K
Place365	General	19K	19K
MMInsturct-QA	General	167K	23K
DriveLM	General	4K	4K
YesBut	General	1K	1K
WildVision	General	6K	6K
LLaVa-Critic-113K	General	113K	56K
RLAIF-V	General	83K	14K
VQAv2	General	83K	18K
MMRA	General	1K	1K
KONIQ	General	30K	30K
MMDU	General	45K	23K
Spot-The-Diff	General	9K	9K
Hatefull-Memes	General	9K	9K
COCO-QA	General	46K	23K
NLVR2	General	50K	25K
Mimic-CGD	General	71K	7K
Datikz	General	44K	8K
Chinese-Meme	General	5K	5K
IconQA	General	27K	27K
Websight	General	10K	10K

Table B | General VQA Data.

B. Dataset

We show the detailed used number of samples of every data source in Tab. B, Tab. C, Tab. D, Tab. E, Tab. F, Tab. G, Tab. H, Tab. J, and Tab. I.

In addition to these existing data, we will also provide the augmented data information later.

Dataset	Category	#Stage-1.5	#Stage-2
SynthDog	Naive OCR	100K	400
MTWI	Naive OCR	10K	10K
LVST	Naive OCR	30K	30K
SROIE	Naive OCR	34K	1K
FUNSD	Naive OCR	199	199
Latex-Formula	Naive OCR	110K	6K
IAM	Naive OCR	58K	16K
Handwriting-Latex	Naive OCR	100K	3K
ArT	Naive OCR	55K	14K
CTW	Naive OCR	26K	26K
ReCTs	Naive OCR	20K	20K
COCO-Text	Naive OCR	16K	16K
SVRD	Naive OCR	2K	2K
Hiertext	Naive OCR	10K	10K
RoadText	Naive OCR	200	200
MapText	Naive OCR	240	240
CAPTCHA	Naive OCR	10K	10K
Est-VQA	Naive OCR	17K	17K
HME-100K	Naive OCR	75K	37K
TAL-OCR-ENG	Naive OCR	10K	10K
TAL-HW-Math	Naive OCR	22K	22K
IMGUR5K	Naive OCR	6K	6K
ORAND-CAR	Naive OCR	5K	5K
Invoices-and-Receipts	Naive OCR	2K	2K
Chrome-Writting	Naive OCR	9K	9K
IIIT5K	Naive OCR	2K	2K
K12-Printing	Naive OCR	257K	51K
Memotion	Naive OCR	6K	6K
Arxix2Markdown	Naive OCR	502K	50K
HW-Mathematicsl-Exp.	Naive OCR	12K	12K
WordArt	Naive OCR	5K	5K
Rendered Text	Naive OCR	10K	10K
Handwriting-Forms	Naive OCR	1K	1K

Table C | Naive OCR Data.

Dataset	Category	#Stage-1.5	#Stage-2
TallyQA	Counting	133K	12K
OODVQA	Counting	3K	3K
RefCOCO/+/g	Grounding	105K	25K
GroundUI	Grounding	17K	8K
Object365	Grounding	1184K	0

Table D | Counting & Grounding Data.

Dataset	Category	#Stage-1.5	#Stage-2
AI2D	Science	12K×4	12K×4
ScienceQA	Science	13K×4	13K×2
TQA	Science	7K	7K
PathVQA	Science	33K	1K
SciQA	Science	296K	7K
VQA-RAD	Science	313	313
VisualWebInsturct	Science	263K	263K
TextBooks-QA	Science	47K	47K

Table E | Science Data. $\times n$ notes repeat the data by n times.

Dataset	Category	#Stage-1.5	#Stage-2
GeoQA+	Math	177K	13K
MathQA	Math	40K	40K
CLEVR	Math	70K	3K
CLEVR-Math	Math	70K	3K
MAVIS-math-rule-geo	Math	100K	100K
MAVIS-math-mategen	Math	86K	86K
InterGPS	Math	1280	1280
Raven	Math	43K	31K
GEOS	Math	498	498
UniGeo	Math	12K	12K

Table F | Math Data.

Dataset	Category	#Stage-1.5	#Stage-2
ShareGPT4o	Captioning	57K	12K
KVQA	Knowledge	24K	24K
Movie-Posters	Knowledge	15K	15K
Google-Landmark	Knowledge	26K	26K
WikitArt	Knowledge	12K	12K
Weather-QA	Knowledge	1100	1100
Coco-colors	Captioning	44K	22K
music-sheet	Knowledge	9K	9K
SPARK	Captioning	6K	6K
SAM-caption	Captioning	78K	39K
Tmbd-Celeb-10K	Knowledge	8K	8K
CC3M	Captioning	2237K	0
Textcaps	Captioning	110K	0
ShareGPT-4V	Captioning	767K	0
DenseFusion	Captioning	1171K	0

Table G | Caption & Knowledge Data.

Dataset	Category	#Stage-1.5	#Stage-2
ChartQA	Chart	60K	60K
MMC-Inst	Chart	363K	11K
DVQA	Chart	197K	8K
PlotQA	Chart	157K	7K
LRV-Instruction	Chart	7K	7K
TamMWP	Table	23K	23K
UniChart	Chart	956K	33K
Vistext	Table	10K	10K
TAT-DQA	Table	2K	2K
VQAonDB	Table	34K	40K
FigureQA	Chart	100K	29K
Chart2Text	Chart	27K	27K
Robut	Table	111K	23K
MultiHiertt	Table	7K	7K

Table H | Chart & Table Data. We heavily use some low-quality data such as MMC-Inst, PlotQA in Stage-1.5. But in our final stage, we just sample a very small part from these sources.

Dataset	Category	#Stage-1.5	#Stage-2
DocVQA	OCR QA	39K×3	39K
InfoVQA	OCR QA	24K×4	24K×4
TextVQA	OCR QA	35K×4	35K×2
ArxivQA	OCR QA	54K	3K
ScreenQA	OCR QA	33K	1K
DocReason	OCR QA	9K	9K
Ureader	OCR QA	75K	37K
FinanceQA	OCR QA	10K	10K
DocMatrix	OCR QA	250K	7K
A-OKVQA	OCR QA	8K	8K
Diagram-Image-To-Text	OCR QA	300	300
MapQA	OCR QA	37K	37K
OCRVQA	OCR QA	166K	83K
ST-VQA	OCR QA	17K	17K
SlideVQA	OCR QA	6K	6K
PDF-VQA	OCR QA	9K	9K
SQuAD-VQA	OCR QA	87K	46K
VQA-CD	OCR QA	330	330
Block-Diagram	OCR QA	48K	1K
MTVQA	OCR QA	7K×4	7K×4
ColPali	OCR QA	46K	23K
BenthanQA	OCR QA	19K	19K

Table I | OCR QA Data. “×4" means we repeat every sample 4 times.

B.1. Dataset Collection

Internal Data To augment the existing OCR data, we used some internal PDF OCR annotated data, notated as Arxiv2Markdown in this work, which converts each page of papers from into the corresponding Markdown format. For this dataset, we use 500k in stage-1.5 and 50k in stage-2. We also use a dataset created from the textbooks as shown in Fig. A, 47K samples are used in both stage-1.5 and stage-2.

Non-QA data conversion Some of the data source was originally in a non-QA format. If it is classification data, we convert it into multiple-choice questions, as shown in Fig. B. For certain datasets with only images, we use automated annotation tools to generate image descriptions, transforming them into captioning data.

Similarity Score our designed similarity score can quickly help us assess the overlap between new data and the existing data pool. Here, we provide an example: if the current data pool is Cambrian-7M, and we aim to introduce new

Dataset	Category	#Stage-1.5	#Stage-2
Orca	Text-only	492K	49K
Orca-math	Text-only	199K	99K
MathInsturct	Text-only	279K	130K
OpenMathInsturct	Text-only	1580K	0
WizardLM	Text-only	70K	42K
TheoremQA	Text-only	796	796
OpenHermes2.5	Text-only	99K	50K
NuminaMath-CoT	Text-only	349K	140K
Python-Code-25k	Text-only	25K	25K
Infinity-Instruct	Text-only	303K	121K
Python-18k-Alpaca	Text-only	18K	18K
Ruozhiba	Text-only	1734	1734
Infinity-Math	Text-only	74K	74K
StepDPO	Text-only	11K	11K
TableLLM	Text-only	73K	36K
UltraInteract-sft	Text-only	279K	84K

Table J | Text-only Data. The quality of text-only data still matters for multi-modal LLMs. We collect a diverse collection of open-source text-only datasets. We also convert some preference datasets into SFT format.

datasets such as UReader, COCO-Colors, and Textbook-QA as shown in Tab. K. Since UReader is a data collection contains DocQA, InfoQA and ChartQA which is already included in Cambrian-7M, we can observe that it has a high similarity score 0.45. The dataset COCO-colors uses COCO images but has new instruction about image colors, so that it has a middle-level score 0.10. Textbooks-QA is our internal data, so it has a relative low score 0.02. In our practice, sources with a score below 0.3 are considered different from the existing data pool. Data with a score above 0.3 may also be retained or removed based on specific considerations. Given the relatively high quality of the data within Ureader, we chose to retain it.

Dataset	Similarity Score	Max value
Ureader	0.45	0.95
Coco-Colors	0.10	0.3
Textbooks-QA	0.02	0.1

Table K | Similarity Score of new data source to Cambrian. We can found that using similarity score can roughly reflect the overlap between new introduced dataset with the existing data pool.

B.2. Dataset Filtering

Our data filtering strategy mainly relies on manual inspection to extract the key features of erroneous data and then filter them through rules.

In addition to the common error data mentioned in the paper, there are other types of errors, but their proportion is relatively small. For example. we notice a particular type of annotated data where the responses are similar to “I cannot answer this.” For questions involving safety or ethical issues, such responses are appropriate and even necessary. However, some data, such as in pure-text form where the question is “Can you help me describe this image” and the answer is “Sorry, I cannot”, are evidently unsuitable for continued use in VLM training. Therefore, we designed a set of keyword-based filtering rules to exclude these samples.

Dataset	Category	Original Answer Type	#Samples
TQA-CoT	Science	Only Option Letter	5K
ChartQA-CoT	Science	Only Final Answer	24K
DVQA-CoT	Chart	Only Final Answer	25K
Clever-CoT	Math	Only Final Answer	13K
Clever-Math-CoT	Math	Only Final Answer	59K
SketchyVQA-CoT	General	Yes/No Answer	8K
Tab-MWP-CoT	Table	Brief Explanation	20K
RAVEN-CoT	Math	Option Letter	9K
MAVIS-math-metagen-CoT	Math	Unformatted Long Answer	86K
UniGeo-CoT	Math	Only Final Answer	12K

Table L | Dataset for CoT data augmentation.

training w/ CoT Data	Evaluating w/ CoT Prompt	MathVista-Mini
✗	✓	61.0
✗	✓	60.5
✓	✓	63.2
✓	✓	63.5

Table M | With CoT training data, adding "Solve this problem step-by-step" prompt can help to improve the performance.

B.3. Subset Selection

As shown in Tab. B, Tab. C, Tab. D, Tab. E, Tab. F, Tab. G, Tab. H, Tab. J and Tab. I, we have several general rules from subset selection.

- For datasets with fewer than 20,000 samples, we do not perform subset selection.
- If we perform subset selection, we remove at least half of the data. For datasets originally exceeding 100,000 samples, in most cases, we limit the subset to no more than 50,000 samples.

B.4. Data Augmentation

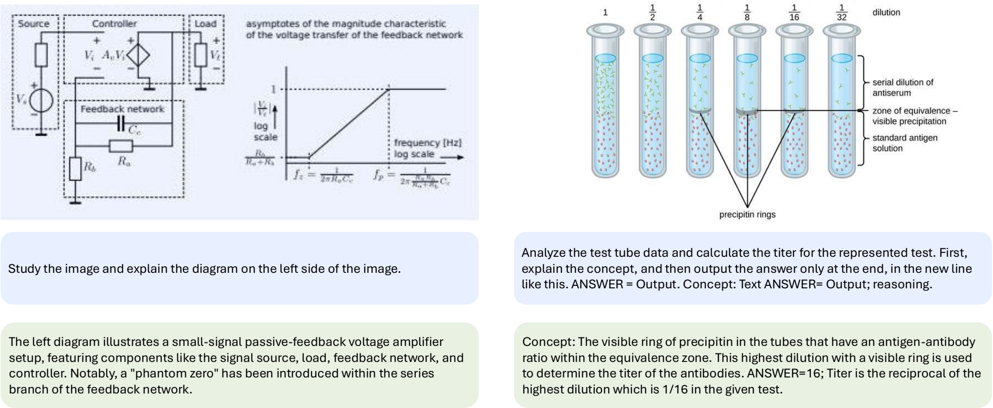
CoT Augmentation. We use existing SOTA VLMs to help us re-write some Science, Math or Chart data to generate detailed chain-of-thought answer. We list datasets used to augment here in Tab. L. We show our prompt used for CoT augmentation in List 1.

In fact, using this automated augmentation method can easily generate incorrect answers or solution steps, even when the correct answer is provided in the prompt. To address this, we employed another LLM to compare the generated answers with the original answers, filtering out some erroneous sample, the prompt is shown in List 2.

To verify the effectiveness of the constructed CoT data,

Model	packing method	DocVQA Test	ChartQA Test	InfoVQA Test	TextVQA Val	OCRBench Test	MMstar Test	RWQA Test	AI2D Test	MMMU Val	MMB$_{1.1}$ EN-Val	MMVet Test	HallB Test	MathVista Test-Mini	Average Score
Eagle2-9B	Greedy	92.6	84.7	76.5	83.8	855	62.7	67.8	84.0	54.7	81.7	63.0	47.9	61.6	72.8
Eagle2-9B	Blanced	92.6	86.4	77.2	83.0	868	62.6	69.3	83.9	56.1	81.9	62.2	49.3	63.8	73.5

Table N | The overall performance of using naive greedy packing is inferior to that of using balanced-aware packing.


Source
Controller
Load
Feedback network
V_s
V_i
$A_v V_i$
V_ℓ
C_c
R_a
R_b
asymptotes of the magnitude characteristic of the voltage transfer of the feedback network
1
$|\frac{V_i}{V_\ell}|$
log scale
$\frac{R_b}{R_a+R_b}$
frequency [Hz] log scale
$f_z = \frac{1}{2\pi R_a C_c}$
$f_p = \frac{1}{2\pi \frac{R_a R_b}{R_a+R_b} C_c}$
1
$\frac{1}{2}$
$\frac{1}{4}$
$\frac{1}{8}$
$\frac{1}{16}$
$\frac{1}{32}$
dilution
serial dilution of antiserum
zone of equivalence – visible precipitation
standard antigen solution
precipitin rings

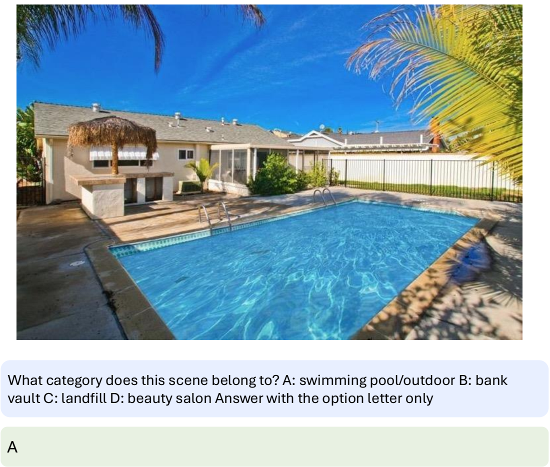
Study the image and explain the diagram on the left side of the image.

The left diagram illustrates a small-signal passive-feedback voltage amplifier setup, featuring components like the signal source, load, feedback network, and controller. Notably, a "phantom zero" has been introduced within the series branch of the feedback network.

Analyze the test tube data and calculate the titer for the represented test. First, explain the concept, and then output the answer only at the end, in the new line like this. ANSWER = Output. Concept: Text ANSWER= Output; reasoning.

Concept: The visible ring of precipitin in the tubes that have an antigen-antibody ratio within the equivalence zone. This highest dilution with a visible ring is used to determine the titer of the antibodies. ANSWER=16; Titer is the reciprocal of the highest dilution which is 1/16 in the given test.

Figure A | Samples of Internal TextbookQA Dataset.



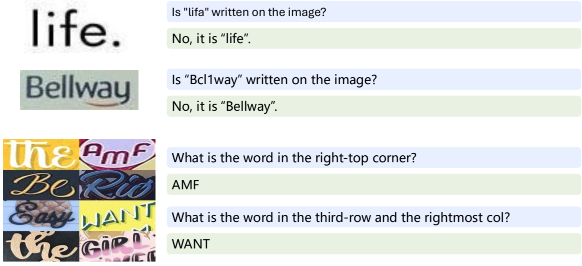
What category does this scene belong to? A: swimming pool/outdoor B: bank vault C: landfill D: beauty salon Answer with the option letter only

A

Figure B | Sample of SpatialScene Dataset.

we evaluate it on MathVista. For models that do not train on the generated CoT data, when adding the CoT prompt "Think it step-by-step" to the question, they do give answers in a CoT format, but unfortunately do not improve accuracy or even significantly reduce it. However, after incorporating our constructed CoT data, using the CoT prompt bring performance boost, as shown in Tab. M.

Rule-Based QA Generation In addition to the previous In addition to the augmentation method for table data, we also designed several other rules to expand the existing dataset as shown in Fig. C. These tasks are designed to reduce model hallucinations and enhance the model's spatial awareness capabilities. Specifically, we use IIIT5K and WordArt dataset for OCR data augmentation.



Is "lifa" written on the image?

No, it is “life”.

Is “Bcl1way” written on the image?

No, it is “Bellway”.

What is the word in the right-top corner?

AMF

What is the word in the third-row and the rightmost col?

WANT

Figure C | Rule-based data augmentation for OCR data.

Expanding Short Answers. We extending the short response in dataset VQAv2, GQA and VSR to detailed response via the prompt List 3.

C. Packing

The knapsacks generated by the naive greedy packing strategy exhibit an unnecessary length distribution bias. To address this, we designed a balanced-aware knapsack method aimed at producing knapsacks with a more uniform length distribution. We show the ablation results with different packing methods in Tab. N. Additional details on practical implementation are provided in List 4.

```
    f"""Rewrite the following answer using a **
Chain of Thought (CoT)** approach. The final
answers should adhere to the following
structure and constraints:

    1. **Problem Restatement**: Start by
restating the problem clearly to set the
context.

    2. **Step-by-Step Process**:
    - **Explicit Steps**: Break the solution
into **discrete steps**, showing all
calculations.
    - **Justifications**: Include a brief
explanation for each step (e.g., referencing
mathematical rules such as the distributive
property, derivative rules, or solving
equations).

    3. **Mathematical Principles**: Where
relevant, mention the specific mathematical
principles or theorems being applied (e.g.,
chain rule, Pythagoras' theorem, etc.).

    4. **Final Answer**: End with the final
solution, clearly boxed or highlighted.

    5. **Consistent Structure**: Ensure every
solution follows this format:
    - **Restatement of the problem**
    - **Steps and calculations with
justifications**
    - **Final answer**

    The output should be detailed but concise,
explaining each step logically while avoiding
excessive repetition. Clarity and logical flow
 are crucial.

    Here is a question and answer pair of this
image:
    Question: {question}
    Answer: {answer}
    """
```

Listing 1 | Prompt we used for CoT Augementation.

```
    f"""Please evaluate if the correctness of
my answer based on the provided question and
the correct answer.

    Question: {question}
    Correct Answer: {ori_answer}
    My Answer: {new_answer}

    Please only return "True" if my answer is
correct, or "False" if it is incorrect.
    My answer is:"""
```

Listing 2 | Prompt we used for judging the correctness of generation chain-of-thought response.

```
    f"""Given the question {question}. The
original answer is {answer}.
    Please reply with a more specific answer
based on the existing answer, as detailed as
possible."""
```

Listing 3 | Prompt we used for expanding short answers.

```
    # Our proposed greedy knapsack method
    def balanced_greedy_knapsack(samples, L,
delta=20):
    # Step 1: Sort the samples
    samples.sort(reverse=True)
    total_length = sum(samples)
    min_knapsacks = (total_length + L - 1) // L
 + delta
    # Step 2: Initialize knapsacks
    knapsacks=[[] for _ in range(min_knapsacks)
]
    knapsack_lengths = [0] * min_knapsacks
    # Step 3: Distribute samples across
knapsacks
    ks_index = 0
    sample_index = 0
    while sample_index < len(samples):
    length = samples[sample_index]
    if knapsack_lengths[ks_index]+length<=L:
    knapsacks[ks_index].append(length)
    knapsack_lengths[ks_index] += length
    sample_index += 1
    else:
    knapsacks.append([])
    knapsack_lengths.append(0)
    ks_index = argmin(knapsack_lengths)

    return knapsacks
```

Listing 4 | In practical implementation, we added an extra redundancy delta to min_knapsacks to avoid creating new knapsacks within the loop. Without this delta, knapsacks with imbalanced distributions could be introduced. The delta value is set based on the data length distribution. The size of samples in our settings is 4k.