
Pixtral 12B



Abstract

We introduce Pixtral 12B, a 12-billion-parameter multimodal language model. Pixtral 12B is trained to understand both natural images and documents, achieving leading performance on various multimodal benchmarks, surpassing a number of larger models. Unlike many open-source models, Pixtral is also a cutting-edge text model for its size, and does not compromise on natural language performance to excel in multimodal tasks. Pixtral uses a new vision encoder trained from scratch, which allows it to ingest images at their natural resolution and aspect ratio. This gives users flexibility on the number of tokens used to process an image. Pixtral is also able to process any number of images in its long context window of 128K tokens. Pixtral 12B substantially outperforms other open models of similar sizes (Llama-3.2 11B & Qwen-2-VL 7B). It also outperforms much larger open models like Llama-3.2 90B while being 7x smaller. We further contribute an open-source benchmark, MM-MT-Bench, for evaluating vision-language models in practical scenarios, and provide detailed analysis and code for standardized evaluation protocols for multimodal LLMs. Pixtral 12B is released under Apache 2.0 license.

Webpage: <https://mistral.ai/news/pixtral-12b/>

Inference code: <https://github.com/mistralai/mistral-inference/>

Evaluation code: <https://github.com/mistralai/mistral-evals/>

1 Introduction

This paper describes Pixtral 12B, a multimodal language model trained to understand both images and text, released with open weights under an Apache 2.0 license. Pixtral is an instruction tuned model which is pretrained on large scale interleaved image and text documents, and hence is capable of multi-turn, multi-image conversation.

Pixtral comes with a new vision encoder which is trained with a novel ROPE-2D implementation, allowing it to process images at their native resolution and aspect ratio. In this way, the model can flexibly process images at low resolution in latency-constrained settings, while processing images at high resolution when fine-grained reasoning is required.

When compared against models of a similar size in the same evaluation setting, we find that Pixtral delivers strong multimodal reasoning capabilities without sacrificing text-only reasoning performance.

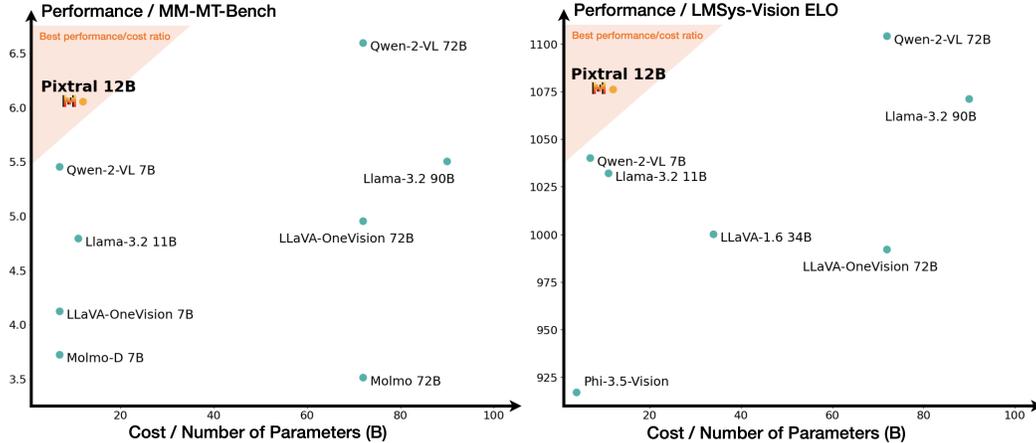


Figure 1: Pixtral Performance. Pixtral outperforms all open-models within its weight class on multimodal tasks by a substantial margin. *Left:* Performance on MM-MT-Bench, a new multimodal, multiturn, instruction following benchmark designed to reflect real world usage of multimodal language models. *Right:* Performance on the public LMSys leaderboard (Vision arena, October 2024).

For instance, our model matches or exceeds the performance of models like Qwen2-VL 7B [23] and Llama-3.2 11B [6] on popular multimodal benchmarks like MMMU [24] and MathVista [14], while outperforming most open-source models on popular text-only tasks like MATH [7] and HumanEval [26]. Pixtral even outperforms much larger models like Llama-3.2 90B [6], as well as closed models such as Claude-3 Haiku [1] and Gemini-1.5 Flash 8B [18], on multimodal benchmarks.

During evaluation of Pixtral and the baselines, we found that evaluation protocols for multimodal language models is not standardized, and that small changes in the setup can dramatically change the performance of some models. We provide thorough analysis of our experience in re-evaluating vision-language models under a common evaluation protocol.

Specifically, we identify two issues with evaluation:

- **Prompts:** Several benchmarks have default prompts which are under-specified, and dramatically reduce the performance of leading closed source models [16, 1] compared to reported figures.
- **Evaluation Metrics:** The official metrics typically require *exact match*, which score model generations as correct only if they exactly match the reference answer. However, this metric penalizes answers which are substantively correct but in a slightly different format (e.g., "6.0" vs "6").

To alleviate these issues, we propose ‘Explicit’ prompts that explicitly specify the format required by the reference answer. We further analyze the impact of flexible parsing for various models, releasing the evaluation code and prompts in an effort to establish fair and standardized evaluation protocols¹.

Moreover, while current multimodal benchmarks mostly evaluate short-form or multiple-choice question answering given an input image, they do not fully capture a model’s utility for practical use cases (e.g. in a multi-turn, long-form assistant setting). To address this, we open-source a novel multimodal, multi-turn evaluation: MM-MT-Bench². We find that performance on MM-MT-Bench correlates highly with ELO rankings on the LMSys Vision Leaderboard.

Pixtral excels at multimodal instruction following, surpassing comparable open-source models on the MM-MT-Bench benchmark (see Figure 1). Based on human preferences on the LMSys Vision Leaderboard, Pixtral 12B is currently the highest ranked Apache 2.0 model, substantially outperforming other open-models such Llama-3.2 11B [6] and Qwen2-VL 7B [23]. It even ranks higher than several closed models such as Claude-3 Opus & Claude-3 Sonnet [1], and several larger models such as Llama-3.2 90B [6].

¹<https://github.com/mistralai/mistral-evals/>

²<https://huggingface.co/datasets/mistralai/MM-MT-Bench>

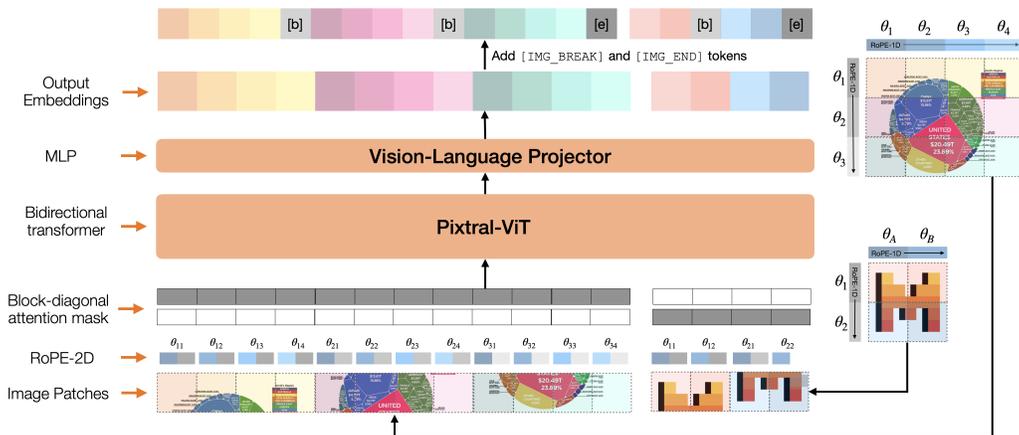


Figure 2: Pixtral Vision Encoder. Pixtral uses a new vision encoder, which is trained from scratch to natively support variable image sizes and aspect ratios. Block-diagonal attention masks enable sequence packing for batching, while RoPE-2D encodings facilitate variable image sizes. Note that the attention mask and position encodings are fed to the vision transformer as additional input, and utilized only in the self-attention layers.

2 Architectural details

Pixtral 12B is based on the transformer architecture [22], and consists of a *multimodal decoder* to perform high-level reasoning, and a *vision encoder* to allow the model to ingest images. The main parameters of the model are summarized in Table 1.

2.1 Multimodal Decoder

Pixtral 12B is built on top of Mistral Nemo 12B [15], a 12-billion parameter decoder-only language model that achieves strong performance across a range of knowledge and reasoning tasks.

2.2 Vision Encoder

In order for Pixtral 12B to ingest images, we train a new vision encoder from scratch, named Pixtral-ViT. Here, our goal is to instantiate a simple architecture which is capable of processing images across a wide range of resolutions and aspect ratios. To do this, we build a 400 million parameter vision transformer [5] (see Table 1) and make four key changes over the standard architectures [17]:

Break tokens: In order to assist the model in distinguishing between images with the same number of patches (same area) but different aspect ratios, we include [IMAGE BREAK] tokens between image rows [2]. We further include an [IMAGE END] token at the end of an image sequence.

Gating in FFN: Instead of standard feedforward layer in the attention block, we use gating in the hidden layer [19].

Sequence packing: In order to efficiently process images within a single batch, we flatten the images along the sequence dimension and concatenate them [3]. We construct a block-diagonal mask to ensure no attention leakage between patches from different images.

RoPE-2D: We replace traditional *learned* and *absolute* position embeddings for image patches with *relative, rotary* position encodings [11, 20] in the self-attention layers. While learned position embeddings must be interpolated to deal with new image sizes (often at the cost of performance), relative position encodings lend themselves naturally to variable image sizes.

Parameters	Decoder	Encoder
dim	5120	1024
n_layers	40	24
head_dim	128	64
hidden_dim	14336	4096
n_heads	32	16
n_kv_heads	8	16
context_len	131072	4096
vocab_size	131072	-
patch_size	-	16

Table 1: Decoder and encoder parameters.

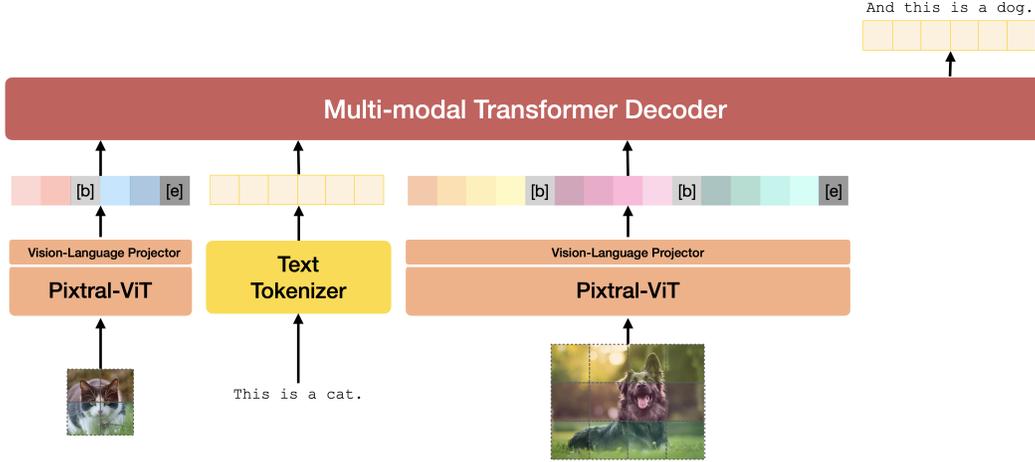


Figure 3: Complete Pixtral Architecture. Pixtral has two components: a *vision encoder*, which tokenizes images, and a *multimodal decoder*, which predicts the next text token given a sequence of text and images. Pixtral can take an arbitrary number of images as input, provided they fit within its 128K context window.

Particularly, let x be a d -dimensional patch vector (either a key or query feature). We denote this feature as $x^{(i,j)}$ when it appears at position (i, j) in the image. Then, the ROPE-2D transform of $x^{(i,j)}$ is expressed as:

$$\text{ROPE-2D} \left(x^{(i,j)}, \Theta \right) = M_{\Theta}^{(i,j)} x^{(i,j)}, \quad (1)$$

$$\text{where } M_{\Theta}^{(i,j)} = \begin{pmatrix} \cos i\theta_1 & -\sin i\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin i\theta_1 & \cos i\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos j\theta_2 & -\sin j\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin j\theta_2 & \cos j\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos j\theta_{\frac{d}{2}} & -\sin j\theta_{\frac{d}{2}} \\ 0 & 0 & 0 & 0 & \cdots & \sin j\theta_{\frac{d}{2}} & \cos j\theta_{\frac{d}{2}} \end{pmatrix}.$$

Here, sub-matrices $M_{\Theta}^{(i,j)} [k : k + 2, k : k + 2]$ capture the height position of the feature (i) for odd values of dimension k , and capture the width position (j) for even values of k (1-based indexing). Furthermore, $\Theta = [\theta_1 \dots \theta_{d/2}]$ is a vector of frequencies for the various dimensions of x , where θ_m is defined following standard practice for ROPE-1D [20].

Critically, our simple implementation of the ROPE-2D transform satisfies the “relative” property: that inner products between two vectors are dependent only on their relative difference in height and width position, rather than their absolute position (see more details in Appendix B).

Discussion: Our vision encoder is specifically designed for multimodal modeling. Traditional encoders are typically optimized for ImageNet performance at a resolution of, for example, 224×224 or 336×336 pixels. When incorporated into multimodal language models – which flexibly perform tasks from standard classification to optical character recognition – prior works typically break an image into smaller (square) tiles before independently feeding tiles to the vision encoder. Instead, our vision encoder can naturally adapt to both high and low resolution images at their native aspect ratio, providing substantially improved performance for multi-modal tasks (see Section 4.4).

2.3 Complete architecture

The Pixtral vision encoder is linked to the multimodal decoder via a two-layer fully connected network. This network transforms the output of the vision encoder into the input embedding size required by the decoder via an intermediate hidden layer of the same size, employing the GeLU activation [8]. The image tokens are treated identically to the text tokens by the multimodal decoder, including RoPE-1D [20] positional encodings for all tokens. Particularly, our decoder uses a causal self-attention mechanism, smoothly facilitating capabilities such as multi-image conversations. The architecture is illustrated in Figure 3.

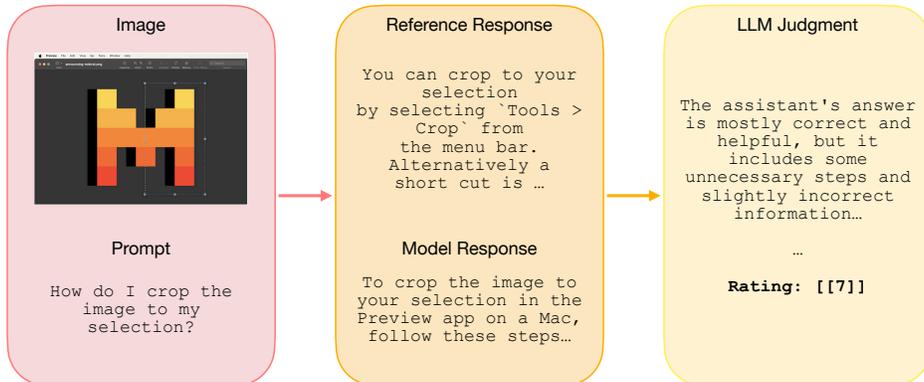


Figure 4: MM-MT-Bench: We open-source a new instruction following benchmark for multimodal models, which correlates highly with LMSys ELO ratings. Given an input image, reference answer and model response, an independent LLM judge is instructed to grade the model’s response on a scale of 1 through 10.

3 MM-MT-Bench: A benchmark for multi-modal instruction following

Most existing multimodal benchmarks measure the ability of a model to perform some form of multiple-choice question answering given an input image. While this is a useful signal for the model’s ability to understand the image, it does not capture the extent of the model’s utility to a user (for instance as a multimodal assistant or chatbot). In order to measure this quality, instruction-tuned text-only models are typically evaluated on MT-Bench [25], wherein an independent LLM *judge* grades a model’s output with respect to a reference answer. We construct and release a new benchmark named Multimodal MT-Bench (MM-MT-Bench) in a similar vein to the text-only variant, to evaluate the performance of instruction-tuned multimodal models.

Design. MM-MT-Bench contains 92 conversations in total. It covers a breadth of practical use cases, covering five categories of images: charts (21), tables (19), PDF pages (24) diagrams (20) and miscellaneous (8). There are 69 single-turn conversations, 18 conversations with 2 turns, 4 of them with 3 turns and 1 conversation with 4 turns. To evaluate a model, we query the model in parallel over all turns of a conversation, providing reference answers for the past turns as history. Each turn is rated independently by the judge with the entire conversation history provided. The judge is prompted to rate the conversation on a scale of 1 to 10 based on correctness (*i.e.* was the extracted information correct) and completeness (*i.e.* does the model answer cover all the points raised in the reference). The evaluation process is illustrated in Figure 4. The judge prompt is provided in Appendix A.5. The results shown in Table 2 show that MM-MT-Bench has a **0.91 Pearson Correlation Coefficient** with LMSys-Vision ELO ratings.

Examples. MM-MT-Bench was designed to mimic real world usage of vision-language models, for extraction, summarization and reasoning over the contents of an image. Representative images from each category are provided in Figure 12 and an example of rated model responses from vision-language models are provided in Figure 11. We manually curated the images, prompts and answers and verified the answers from a second group of labelers. We ensure that all prompts require reference to the image input to be answered correctly.

4 Results

In this section, we provide evaluations of Pixtral 12B against closed and open-source models across a range of model sizes, re-evaluating all models through the same evaluation harness. Particularly, for each dataset, we design the prompt such that we can reproduce the results of leading multimodal models (GPT-4o [16] and Claude-3.5 Sonnet [1]). These prompts are ‘Explicit’ and fully specify the output format (see Section 4.2), allowing models which follow the prompt instructions to be marked accurately at test-time. All models were evaluated with the same prompts, which are specified in Appendix A. We provide additional analysis on re-evaluating models under various prompts and metrics in Sections 4.2 and 4.3, as well as in Appendices D and E.

	Mathvista	MMMU	ChartQA	DocVQA	VQAv2	MM-MT-Bench	LMSys-Vision
	CoT	CoT	CoT	ANLS	VQA Match	GPT-4o Judge	(Oct '24)
Pixtral 12B	58.3	52.0	81.8	90.7	78.6	6.05	1076
Qwen-2-VL 7B [23]	53.7	48.1	41.2	94.5	75.9	5.45	1040
→ w/ Flexible Parsing	55.2	48.7	77.5	–	–	–	–
Llama-3.2 11B [6]	24.3	23.0	14.8	91.1	67.1	4.79	1032
→ w/ Flexible Parsing	47.9	45.3	78.5	–	–	–	–
Molmo-D 7B [4]	12.3	24.3	27.0	72.2	57.1	3.72	–
LLaVA-OneVision 7B [9]	36.1	45.1	67.2	90.5	78.4	4.12	–
Claude-3 Haiku [1]	44.8	50.4	69.6	74.6	68.4	5.46	1000
Gemini-1.5-Flash 8B ₍₀₈₂₇₎ [18]	56.9	50.7	78.0	79.5	65.5	5.93	1111
Molmo 72B [4]	52.2	52.7	75.6	86.5	75.2	3.51	–
LLaVA-OneVision 72B [9]	57.2	54.4	66.9	91.6	83.8	4.95	992
Qwen-2-VL 72B [23]	68.2	60.3	66.6	96.3	81.6	6.59	1104
Llama-3.2 90B [6]	49.1	53.7	33.8	85.7	67.0	5.50	1071
GPT-4o ₍₀₅₁₃₎ [16]	64.6	68.6	85.1	88.9	77.8	7.72	1208
Claude-3.5 Sonnet [1]	64.4	68.0	87.6	90.3	70.7	7.50	1189

Table 2: Multimodal Benchmarks. Pixtral substantially outperforms open models of a similar size, as well as several closed-source models. We re-evaluate all models with the same prompt and evaluation metric (see Section 4.2). For transparent comparison against Qwen2-VL 7B [23] and Llama-3.2 11B [6], we additionally report their performance under relaxed evaluation constraints in (gray) (see Section 4.3). To further investigate the gap with reported figures for some open-source models, we provide analysis in Section E.

	MT-Bench	MMLU	Math	HumanEval
		5-shot	Maj@1	Pass@1
Pixtral 12B	7.68	69.2	48.1	72.0
LLaVA-OneVision 7B [9]	6.94	67.9	38.6	65.9
Molmo-D 7B [4]	4.53	61.2	10.2	3.7
Qwen-2-VL 7B [23]	6.41	68.5	27.9	62.2
Llama-3.2 11B [6]	7.51	68.5	48.3	62.8

Table 3: Language benchmarks. Pixtral 12B consistently outperforms open-source models of a comparable size on text-only benchmarks, making it a drop-in multimodal replacement for existing text-only deployments.

4.1 Main Results

Multimodal performance: Table 2 shows that Pixtral substantially outperforms all open models around its scale on multimodal benchmarks, as well as closed source models such as Claude-3 Haiku [1] and Gemini-1.5 Flash 8B [18]. Particularly, Pixtral outperforms all models of comparable size on MM-MT-Bench, which targets real world use cases, a finding corroborated by strong performance on LMSys Vision Arena. On this public leaderboard, Pixtral 12B approaches the performance of the largest open-weights models, such as Qwen2-VL 72B [23] and Llama-3.2 90B [6].

We highlight that, with our ‘Explicit’ prompts, the performance of some open-source models is substantially lower than their reported figures. For the closest open-source models – Qwen2-VL 7B [23] and Llama-3.2 11B [6] – this is mainly due to models not following instructions on answer formatting (e.g. generating "The answer is 6." instead of "Final answer: 6"). For transparent comparison against these models, we further report their evaluations using relaxed metrics, with more flexible parsing, in gray (see Section 4.3). We analyze the performance of these models under various prompts in Appendix D. In Appendix E, we customize the evaluation to each model in turn, describing the changes required to bridge the gaps to reported performance.

Language performance: Table 3 evaluates Pixtral 12B against open-source models of comparable size on common text-only benchmarks (again, with common prompting and evaluation protocols). Pixtral does not compromise text understanding in pursuit of multimodal capabilities, making it a suitable drop-in replacement for both text and vision tasks.

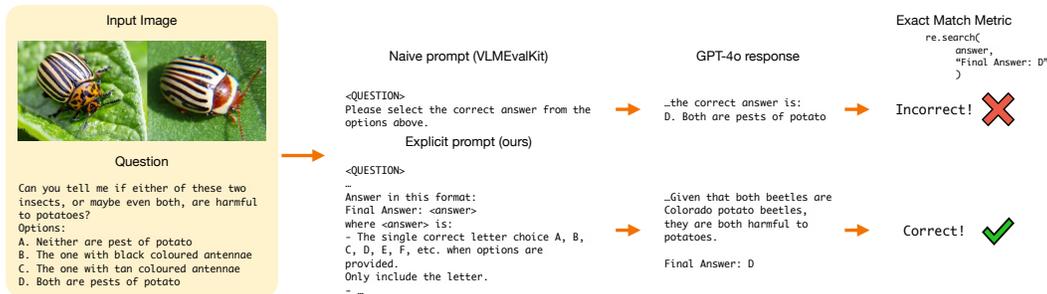


Figure 5: Effect of ‘Naive’ vs. ‘Explicit’ prompts on leading models. Leading models benefit greatly from ‘Explicit’ prompts which provide details about the output format. This makes sense, as otherwise substantively correct responses are marked as incorrect during evaluation (top row, right).

Prompt →	VQAv2		ChartQA		MMMU	
	Naive	Explicit	Naive	Explicit	Naive	Explicit
GPT-4o (0513) [16]	64.2	77.8	58.0	85.1	55.0	68.6
Sonnet-3.5 [1]	50.2	70.7	39.6	87.6	48.6	68.0
Qwen-2-VL 7B [23]	82.1	75.9	83.4	41.2	46.7	48.1
Llama-3.2 11B [21]	29.5	67.1	0.0	14.8	20.7	23.0
Llama-3.2 90B [21]	52.6	67.0	3.9	33.8	27.0	53.7
Pixtral 12B	78.9	78.6	84.3	81.8	45.8	52.0

Table 4: Prompt ablations. Leading models require prompts which explicitly specify the output format to perform well. Pixtral 12B performs well with both ‘Explicit’ and ‘Naive’ prompts, with only a minor regression on ChartQA.

4.2 Prompt selection

Here we discuss our methodology for designing the evaluation prompts. In our evaluation harness, we choose prompts which allow for reproduction of the reported results of leading closed-source models: GPT-4o [16] and Claude-3.5-Sonnet [1]. These prompts are provided in Appendix A, and we report results averaged over 10 prompts in Appendix D.

We find that commonly used prompts do not properly specify the output format. For instance, for a multiple choice question, we find open-source prompts include vague instructions like "Select the correct answer from the options above". In this case, it is impossible for models to know whether answers should be presented as an index ("Option A", "Option B" etc.) or with a natural language response. Models are then penalized for incorrect formatting. As such, leading models require prompts which *explicitly* specify the required output format. We illustrate this with a real example from MMMU in Figure 5.

In Table 4, we demonstrate that our ‘Explicit’ prompts substantially improve the performance of leading models over ‘Naive’ prompts. We also note that in a number of cases, the performance of smaller models *reduces* with the Explicit prompt format, perhaps due to a discrepancy with the prompt-style in the training set of these benchmarks. Pixtral 12B generally performs better with Explicit prompts, with only a minor regression on ChartQA.

4.3 Sensitivity to evaluation metrics

In Section 4.2, we discuss the importance of prompts which properly specify the output format. However, during evaluations, we find that even with Explicit prompts, many models still provide outputs in various formats, which are then penalized by metrics which require responses to match the reference answers exactly.

To investigate this, we take models’ generations and evaluate them under progressively looser parsing constraints. For instance, if the correct answer is "6", flexible metrics do not penalize answers such as "6.0" or "The answer is 6". We provide the details of these parsing settings in Appendix C, but

	Llama-3.2 11B [21]	Llama-3.2 90B [21]	Qwen2-VL 7B [23]	Pixtral 12B
Mathvista				
Baseline	24.3	49.1	53.7	58.3
Flexible level 1	25.9	50.3	54.3	58.3
Flexible level 2	40.2	54.7	54.3	58.3
Flexible level 3	47.9	57.3	55.2	58.5
MMMU				
Baseline	23.0	53.7	48.1	52.0
Flexible level 1	23.4	53.7	48.1	52.0
Flexible level 2	41.0	55.7	48.1	52.0
Flexible level 3	45.3	56.7	48.7	52.0
ChartQA				
Baseline	14.8	33.8	41.2	81.8
Flexible level 1	20.4	33.9	73.8	81.9
Flexible level 2	29.9	35.6	73.8	81.9
Flexible level 3	78.5	79.1	77.5	82.0

Table 5: Flexible parsing ablations. We evaluate models under progressively looser parsing constraints (see Appendix C for details). Under loose parsing constraints, the performance of some models dramatically improves. Pixtral 12B performance is stable under all parsing conditions, and continues to lead even when flexible parsing is accounted for. ‘Flexible Level 3’ is included for illustration only, as it allows some incorrect answers to be marked as correct.

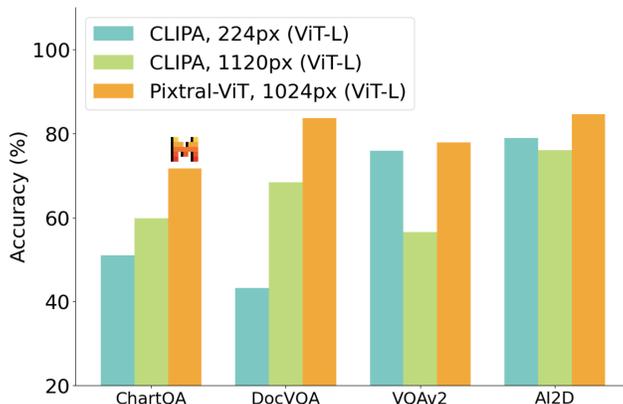


Figure 6: Vision encoder ablations: When leveraged for visual instruction tuning, our encoder substantially outperforms a strong CLIPA [10] baseline for tasks requiring fine-grained document understanding, while maintaining parity for natural images.

here note that ‘Flexible Level 3’ marks a response as correct if the reference answer occurs *anywhere* in the generation. This is an overly generous metric which is included only to illustrate an upper bound, as it permits answers like "6000" for a reference answer of "6".

We provide the results of our analysis in Table 5. We find that the performance of some models dramatically improves with more flexible parsing metrics, indicating that the lower scores can be attributed to the inability of models to properly follow prompt instructions. We further note that Pixtral 12B benefits very little from flexible parsing (substantiating its ability to follow instructions), and furthermore can generally outperform other models even after flexible metrics are used.

4.4 Vision Encoder Ablations

In order to verify the design choices for our vision encoder, we conduct small-scale ablations with Visual Instruction Tuning [13]. We conduct short-horizon multimodal instruction-tuning runs, both with our vision encoder (Pixtral-ViT), as well as a CLIPA [10] backbone as a baseline. For both vision encoders, we use Mistral-Nemo 12B-Instruct [15] to initialize the multimodal decoder.

Like many open-source vision encoders, CLIPA is trained at a fixed resolution of 224×224 pixels. In order to upscale the resolution in vision-language models, existing methods [12] construct several tiled crops from the image, and pass each crop independently through the vision encoder at its pretraining resolution. We conduct two ablations with CLIPA: (a) we resize the entire image to 224×224 ; (b) we construct 25 crops of the input image, for a total resolution of 1120×1120 . These models are also evaluated at 224 pixels and 1120 pixels respectively, while our flexible encoder is evaluated at variable image resolutions, with a maximum resolution of 1024 pixels.

In Figure 6, we find that our model substantially outperforms CLIPA in settings which require fine-grained understanding, such as chart and document understanding, while matching its performance on natural language benchmarks such as VQAv2.

5 Qualitative examples

We discuss real world application of Pixtral by looking at some qualitative examples. Specifically, Pixtral can be used for reasoning over complex figures (eg. Fig. 7), multi-image instruction following (eg. Fig. 8), chart understanding and analysis (eg. Fig. 9) and converting image to code (eg. Fig. 10).

In Fig. 11, we compare Pixtral 12B to QwenVL-7B and Gemini-1.5 Flash-8B (0827) on an example from MM-MT-Bench. The example consists of a complex chart on job jitters in the US with an instruction requiring accurate understanding, reasoning and analysis of the chart. Pixtral’s response is complete and accurate, hence getting a rating of 8, while Gemini-Flash-8B extracts wrong information, and QwenVL does not elaborate on trends.

6 Conclusion

This paper introduced Pixtral 12B, a state-of-the-art multimodal model that excels in both text-only and multimodal tasks. With a novel architecture featuring a 400M-parameter vision encoder and a 12B-parameter multimodal decoder, Pixtral 12B demonstrates strong performance across various benchmarks, outperforming other open models and matching larger models. Its superior instruction following abilities, support for variable image sizes, and long context window make it highly versatile for complex multimodal applications. Pixtral 12B is released under the Apache 2.0 license.

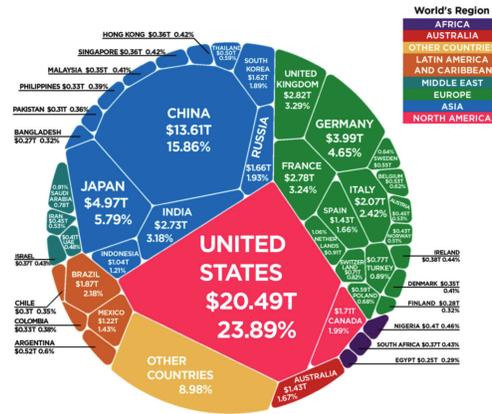
7 Contributors

Mistral AI Science team (listed in alphabetical order by last name):

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick Von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, Sophia Yang.

Acknowledgements

We extend our thanks to the LMSys team for their assistance in deploying our model in the LLM arena, and the vLLM team for their help in integrating Pixtral 12B into their inference library.



DC List the top 5 countries in Europe with the highest GDP



The top 5 countries in Europe with the highest GDP, as represented in the diagram, are:

1. **Germany**
 - GDP: \$3.99 trillion
 - Percentage: 4.65%
2. **United Kingdom**
 - GDP: \$2.82 trillion
 - Percentage: 3.29%
3. **France**
 - GDP: \$2.78 trillion
 - Percentage: 3.24%
4. **Italy**
 - GDP: \$2.07 trillion
 - Percentage: 2.42%
5. **Spain**
 - GDP: \$1.43 trillion
 - Percentage: 1.66%

These countries are highlighted in green on the diagram.

Figure 7: Reasoning over complex figures. An example showcasing Pixtral’s capabilities to understand and reason over complex figures. Pixtral correctly identifies that the green boxes represent the European countries and then reads and sorts the GDP of all the European countries to list the top 5 with accurate GDP numbers.

DC Combine both the tables into a single markdown table

Model	Mathvista (CoT)	MMMU (CoT)	ChartQA (CoT)	DocVQA (ANLS)	VQAv2 (VQA Match)
Pixtral 12B	58.3	52.0	81.8	90.7	78.6
Claude-3 Haiku	44.8	50.4	69.6	74.6	68.4
Gemini-1.5 Flash 8B (0827)	56.9	50.7	78.0	79.5	65.5
LLaVA-OV 72B	57.2	54.4	66.9	91.6	83.8
GPT-4o	64.6	68.6	85.1	88.9	77.8
Claude-3.5 Sonnet	64.4	68.0	87.6	90.3	70.7

Model	Mathvista (CoT)	MMMU (CoT)	ChartQA (CoT)	DocVQA (ANLS)	VQAv2 (VQA Match)
Pixtral 12B	58.3	52.0	81.8	90.7	78.6
Qwen2-VL 7B	53.7	48.1	41.2	94.5	75.9
LLaVA-OV 7B	36.1	45.1	67.2	90.5	78.4
Phi-3 Vision	36.4	40.3	72.0	84.9	42.4
Phi-3.5 Vision	39.3	38.3	67.7	74.4	56.1

Multimodal Benchmarks

Certainly! Below is the combined markdown table from the two provided tables:

```
## Multimodal Benchmarks
| Model | Mathvista (CoT) | MMMU (CoT) | ChartQA (CoT) | DocVQA (ANLS) |
|-----|-----|-----|-----|-----|
| Pixtral 12B | 58.3 | 52.0 | 81.8 | 90.7 |
| Claude-3 Haiku | 44.8 | 50.4 | 69.6 | 74.6 |
| Gemini-1.5 Flash 8B (0827) | 56.9 | 50.7 | 78.0 | 79.5 |
| LLaVA-OV 72B | 57.2 | 54.4 | 66.9 | 91.6 |
| GPT-4o | 64.6 | 68.6 | 85.1 | 88.9 |
| Claude-3.5 Sonnet | 64.4 | 68.0 | 87.6 | 90.3 |
| Qwen2-VL 7B | 53.7 | 48.1 | 41.2 | 94.5 |
| LLaVA-OV 7B | 36.1 | 45.1 | 67.2 | 90.5 |
| Phi-3 Vision | 36.4 | 40.3 | 72.0 | 84.9 |
| Phi-3.5 Vision | 39.3 | 38.3 | 67.7 | 74.4 |
```

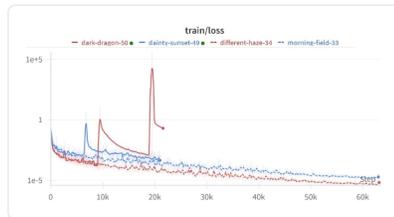
This table combines the performance metrics of various models across multiple benchmarks.



Figure 8: Multi-image instruction following. Pixtral can process arbitrary number of images in its context window. The example shows that Pixtral can successfully combine the information from both images into a single markdown table.

DC

When did things start to go wrong for dark dragon?



The image is a line graph titled 'train/loss' that shows the training loss over steps for four different models: dark-dragon-50, dainty-sunset-49, different-haze-34, and morning-field-33. The x-axis represents the steps, ranging from 0 to 60k, and the y-axis represents the loss on a logarithmic scale from 1e-5 to 1e+5.

To determine when things started to go wrong for the dark-dragon-50 model, we need to look at the red line representing this model. Initially, the loss decreases, indicating that the model is learning effectively. However, around the 10k step mark, the loss starts to spike significantly. This spike indicates that the model's performance is deteriorating. The loss continues to spike and fluctuate, reaching very high values, which suggests that the model is not learning effectively and is likely overfitting or encountering other issues.

Therefore, the critical point where things started to go wrong for the dark-dragon-50 model is around the 10k step mark.

Pixtral ▾

Ask anything!



Figure 9: Chart Understanding and Analysis. Pixtral demonstrates the capability to interpret and analyze intricate charts with high accuracy. In this instance, Pixtral correctly identifies that "dark-dragon" corresponds to the red line. Furthermore, it recognizes that the training loss is expected to decrease smoothly and notes that the training run became unstable around the 10K step mark due to a significant spike in loss.



DC Write HTML code to create a website like this



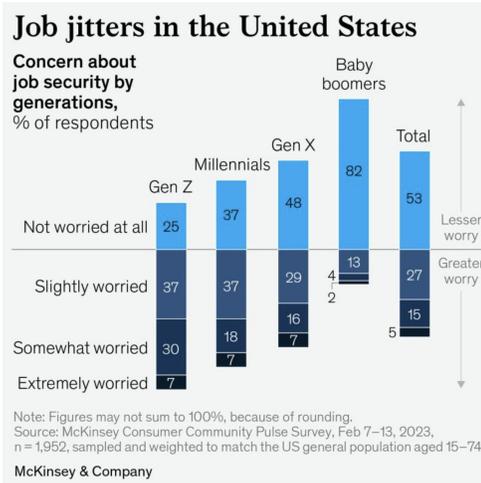
Certainly! Below is an example of HTML code to create a simple webpage that allows users to pick an ice cream flavor. This example includes a dropdown menu for selecting the flavor and a "Next" button.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pick an Ice Cream Flavor</title>
</head>
<body>
  <div class="container">
    <h1>Pick An Ice Cream Flavor</h1>
    <select id="flavorSelect">
      <option value="vanilla">Vanilla</option>
      <option value="chocolate">Chocolate</option>
      <option value="strawberry">Strawberry</option>
      <option value="cookie_dough">Cookie Dough</option>
      <option value="mint_chocolate">Mint Chocolate</option>
    </select>
    <button onclick="next()">Next</button>
  </div>
  <div style="position: fixed; bottom: 0; width: 100%; text-align: center; font-size: 0.8em; font-family: sans-serif; color: #ccc; background-color: #333; padding: 5px 0 0 0;">
    &copy; copyright MistralAI
  </div>
  <script>
    function next() {
      const selectedFlavor = document.getElementById('flavorSelect').value;
      alert('You selected: ' + selectedFlavor);
    }
  </script>
</body>
</html>

```

Figure 10: Image to Code. This demonstration illustrates Pixtral’s capability to convert hand-drawn website interfaces into executable HTML code, bringing hand-drawn designs to life as fully functional websites.



USER: In aggregate, how are different generations feeling about their work prospects? Use precise numbers to back up your claims. Explain smaller and broader trends

Reference

Analysis of Generational Work Prospect Concerns

Data Breakdown by Generation

- Gen Z**
 - 71% worried
 - 37% Slightly worried
 - 39% Somewhat worried
 - 7% Extremely worried
 - 29% Not worried at all
- Millennials**
 - 63% worried
 - 31% Slightly worried
 - 18% Somewhat worried
 - 7% Extremely worried
 - 37% Not worried at all
- Gen X**
 - 52% worried
 - 29% Slightly worried
 - 16% Somewhat worried
 - 7% Extremely worried
 - 48% Not worried at all
- Baby Boomers**
 - 18% worried
 - 13% Slightly worried
 - 4% Somewhat worried
 - 2% Extremely worried
 - 82% Not worried at all

Broader Trends

- Clear generational divide in job security concerns. Younger generations are significantly more worried than older ones.
- Worry decreases steadily as age increases, with Gen Z being the most concerned and Baby Boomers being the least concerned.
- The majority of Gen Z, Millennials, and Gen X are worried to some degree, while the vast majority of Baby Boomers are not worried.

Smaller Trends

- Level of extreme worry is relatively consistent (around 7%) for Gen Z, Millennials, and Gen X, but drops dramatically for Baby Boomers (2%).
- Slight worry is the most common level of concern for Gen Z and Millennials (both at 37%), while it's less prevalent in older generations.
- Gen Z has the highest percentage of "Somewhat worried" respondents at 39%, indicating a deeper level of concern compared to other generations.

Summary

There's a strong inverse relationship between age and job security concerns, with younger generations feeling much more uncertain about their work prospects than older ones. This could reflect factors such as job market experience, career stability, proximity to retirement, or differing perceptions of the current economic climate across generations.

To determine how different generations feel about their work prospects, we need to analyze the percentages of respondents from each generation who are worried about job security. The generations included in the survey are Gen Z, Millennials, Gen X, Baby Boomers, and the total average.

- Gen Z:**
 - Not worried at all: 25%
 - Slightly worried: 37%
 - Somewhat worried: 30%
 - Extremely worried: 7%
 - Total worried (slightly, somewhat, extremely): 37% + 30% + 7% = 74%
- Millennials:**
 - Not worried at all: 37%
 - Slightly worried: 37%
 - Somewhat worried: 18%
 - Extremely worried: 7%
 - Total worried (slightly, somewhat, extremely): 37% + 18% + 7% = 62%
- Gen X:**
 - Not worried at all: 48%
 - Slightly worried: 29%
 - Somewhat worried: 16%
 - Extremely worried: 7%
 - Total worried (slightly, somewhat, extremely): 29% + 16% + 7% = 52%
- Baby Boomers:**
 - Not worried at all: 82%
 - Slightly worried: 13%
 - Somewhat worried: 4%
 - Extremely worried: 2%
 - Total worried (slightly, somewhat, extremely): 13% + 4% + 2% = 19%
- Total Average:**
 - Not worried at all: 53%
 - Slightly worried: 27%
 - Somewhat worried: 15%
 - Extremely worried: 5%
 - Total worried (slightly, somewhat, extremely): 27% + 15% + 5% = 47%

Analysis:

- Gen Z** has the highest percentage of respondents worried about job security, with 74% expressing some level of worry.
- Millennials** follow with 62% worried.
- Gen X** has 52% worried.
- Baby Boomers** have the lowest percentage of worry, with only 19% expressing concern.
- The **total average** shows that 47% of respondents are worried about job security.

Trends:

- Younger generations (Gen Z and Millennials) show higher levels of concern about job security compared to older generations.
- Baby Boomers are the least worried, indicating a trend where older generations may feel more secure in their job prospects.
- The overall trend suggests that younger workers are more anxious about job security than their older counterparts. Younger generations (Gen Z and Millennials) are more worried about job security, with 74% and 62% expressing concern, respectively. Gen X follows with 52%, while Baby Boomers are the least worried at 19%. The total average shows that 47% of respondents are worried about job security. This indicates that younger workers are more anxious about job security compared to older generations.

Judge: Rating = 8

The chart shows the concern about job security by generations in the United States, based on a survey conducted by McKinsey Consumer Community Pulse Survey from February 7 to 13, 2023. Here are the key findings:

- Total Concern (53%):**
 - The overall concern about job security is 53%.
 - This includes a mix of not worried at all (27%), slightly worried (15%), somewhat worried (5%), and extremely worried (2%).
- Generational Breakdown:**
 - Baby Boomers:** 82% are not worried at all, 13% are slightly worried, and 2% are somewhat worried.
 - Gen X:** 48% are not worried at all, 29% are slightly worried, 16% are somewhat worried, and 7% are extremely worried.
 - Millennials:** 37% are not worried at all, 37% are slightly worried, 18% are somewhat worried, and 7% are extremely worried.
 - Gen Z:** 25% are not worried at all, 37% are slightly worried, 30% are somewhat worried, and 7% are extremely worried.
- Trends:**
 - Baby Boomers:** The highest concern (82%) is among Baby Boomers, indicating a significant level of worry about job security.
 - Gen X:** Gen X has a moderate level of concern, with 48% not worried and 29% slightly worried.
 - Millennials:** Millennials have the lowest level of concern, with 37% not worried at all and 37% slightly worried.
 - Gen Z:** Gen Z falls in between, with 25% not worried at all and 37% slightly worried.
- Overall Concern:**
 - The total concern about job security is highest among Baby Boomers (82%), followed by Gen X (48%), Millennials (37%), and Gen Z (25%).
- Worry Levels:**
 - The majority of respondents (53%) are either somewhat or extremely worried about job security.
 - Only a small portion (27%) are not worried at all.
 - In summary, the chart shows that Baby Boomers are the most concerned about job security, followed by Gen X, with Millennials and Gen Z having lower levels of concern. The overall concern is significant, with a majority of respondents expressing some level of worry.

Judge: Rating = 3

Based on the provided chart, here's a breakdown of how different generations feel about their job security:

- Baby Boomers** are the most concerned about job security, with 82% expressing some level of worry.
- Gen X** follows closely behind with 48% expressing concern.
- Millennials** show a lower level of concern at 37%.
- Gen Z** has the lowest level of concern, with only 37% expressing some level of worry.

Smaller Trends:

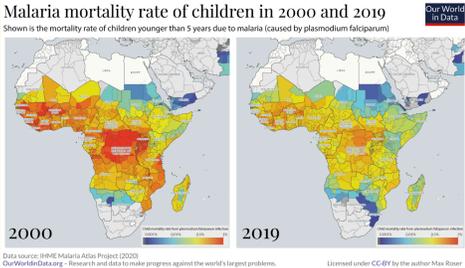
- The overall trend shows a decreasing level of concern as generations get younger.
- Baby Boomers** have the highest level of concern, while **Gen Z** has the lowest.

Broader Trends:

- Despite the decreasing trend, the percentage of people who feel some level of worry is still significant across all generations. This indicates that job security is a concern for a large portion of the US population, regardless of age.
- The data suggests that the younger generations are generally less worried about job security than older generations. This might be influenced by factors such as a more dynamic job market and a more flexible approach to work.
- It's important to remember that this data is based on a survey, so it's not exhaustive, and there could be other factors influencing these results. Nonetheless, the chart provides a clear picture of the perceived job security concerns across different generations in the United States.

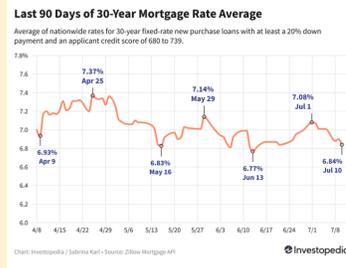
Judge: Rating = 2

Figure 11: Examples of model responses from Pixtral-12B, QwenVL-7B and Gemini-1.5 Flash-8B (0827) LLM-as-a-judge scores. Pixtral's response is complete and accurate, hence getting a rating of 8, while Gemini-Flash-8B extracts wrong information, and QwenVL does not elaborate on trends.



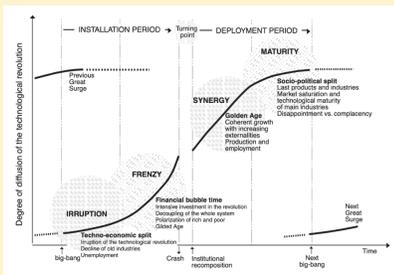
USER: Based on the heatmap, what can we observe about malaria trends from 2000 to 2019?

Category: CHARTS



USER: Identify the three largest drops in the 30-year mortgage rate shown in this chart. For each drop, state the starting and ending dates, calculate the total basis point decrease and the rate of change. Which of these drops was the largest in magnitude and which had the largest rate of change?

Category: CHARTS



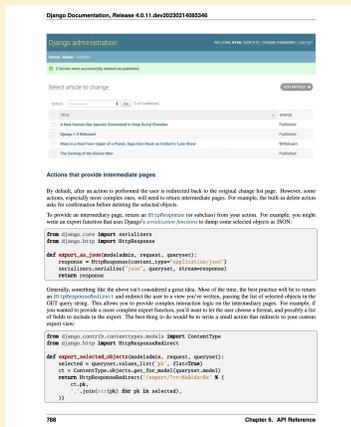
USER: Can you explain to me the mechanism for the evolution of technology in this diagram?

Category: DIAGRAMS

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
5:00 AM	Wake Up/Get Ready				Wake Up/Get Ready		
6:00 AM	Commuter	Wake Up/Com		Wake Up/Get Ready			
7:00 AM	Commuter	Get Ready	Wake Up/Com	Commuter	Commuter		
8:00 AM	Work 8:00			Work 8:30	Work 8:00	Cym	Cym
9:00 AM	Commuter						
10:00 AM	Work 10:00	Commuter				Homework	Errands
11:00 AM			Work 11:00			Study	
12:00 PM							Homework
1:00 PM							Study
2:00 PM				Unit 2:30			
3:00 PM				Class 3:10			GO OUTSIDE
4:00 PM							HAVE FUN
5:00 PM	Unit 4:00	Unit 5:30	Unit 5:30	Unit 4:00	Unit 4:00		Meal Prep for the week
6:00 PM	Commuter	Class 6:10 AM before	Class 6:10 AM before	Commuter			
7:00 PM	Driver and Prep	Unit 8:00		Driver and Prep	Go Out		Relax
8:00 PM		Commuter	Unit 9:15	Dr. Stay In			
9:00 PM	Sleep	Prep	Commuter	Sleep	Have Fun		Sleep
10:00 PM		Sleep					
11:00 PM			Sleep		Sleep		

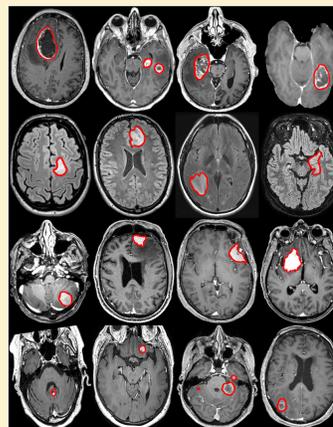
USER (turn 1): How long do I spend commuting a week?
USER (turn 2): How long do I spend sleeping each week?

Category: TABLES



USER: What are the main points discussed in this doc?

Category: PDF PAGES



USER (turn 1): Is there tumor present in these scans?
USER (turn 2): In which images do the tumors appear largest? Provide details.

Category: MISCELLANEOUS

Figure 12: Example images from MM-MT-Bench

References

- [1] Anthropic (2024). The Claude 3 Model Family: Opus, Sonnet, Haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.
- [2] Bavishi, R., Elsen, E., Hawthorne, C., Nye, M., Odena, A., Somani, A., and Taşırlar, S. (2023). Fuyu-8b: A multimodal architecture for ai agents.
- [3] Dehghani, M., Mustafa, B., Djolonga, J., Heek, J., Minderer, M., Caron, M., Steiner, A., Puigcerver, J., Geirhos, R., Alabdulmohsin, I. M., et al. (2024). Patch n’pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36.
- [4] Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J. S., Salehi, M., Muennighoff, N., Lo, K., Soldaini, L., et al. (2024). Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*.
- [5] Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- [6] Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- [7] Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. (2021). Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- [8] Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.
- [9] Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Li, Y., Liu, Z., and Li, C. (2024). Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.
- [10] Li, X., Wang, Z., and Xie, C. (2023). An inverse scaling law for clip training. In *NeurIPS*.
- [11] Li, Y. and Harada, T. (2022). Leopard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5554–5564.
- [12] Liu, H., Li, C., Li, Y., and Lee, Y. J. (2024a). Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.
- [13] Liu, H., Li, C., Wu, Q., and Lee, Y. J. (2024b). Visual instruction tuning. *Advances in neural information processing systems*, 36.
- [14] Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.-W., Galley, M., and Gao, J. (2023). Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.
- [15] MistralAI (2024). Mistral NeMo 12B. <https://mistral.ai/news/mistral-nemo/>.
- [16] OpenAI, R. et al. (2023). Gpt-4 technical report. *ArXiv*, 2303:08774.
- [17] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- [18] Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.-b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- [19] Shazeer, N. (2020). Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

- [20] Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. (2024). Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.
- [21] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- [22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [23] Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., and Lin, J. (2024). Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution.
- [24] Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al. (2023). Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arxiv*.
- [25] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.
- [26] Zhong, W., Cui, R., Guo, Y., Liang, Y., Lu, S., Wang, Y., Saied, A., Chen, W., and Duan, N. (2023). Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

Appendix

Table of Contents

A Prompts	19
A.1 MMMU and Mathvista	19
A.2 ChartQA	19
A.3 VQAv2	19
A.4 DocVQA	19
A.5 MM-MT-Bench Judge Prompt	19
B Relative Position Encoding Property of ROPE-2D	20
C Flexible Parsing Settings	21
D Robustness to prompting	21
D.1 Llama-Specific Prompts	21
D.2 Average performance across prompts	22
E Reproducing Reported Numbers	22
E.1 Summary	22
E.2 Closed models: Claude-3 Haiku and Gemini-Flash-8B	22
E.3 Qwen2-VL 7B	22
E.4 Llama-3.2	23
E.5 Llava-OneVision 72B	23
E.6 Molmo	23

A Prompts

Here we open-source the prompts used for evaluations in the main paper. As discussed in Section 4.2, prompts are selected to reproduce reported performance of GPT-4o [16] and Claude-3.5 Sonnet [1].

A.1 MMMU and Mathvista

Analyze the image and question carefully, using step-by-step reasoning. First, describe any image provided in detail. Then, present your reasoning. And finally your final answer in this format:
Final Answer: <answer>
where <answer> is:

- The single correct letter choice A, B, C, D, E, F, etc. when options are provided. Only include the letter.
- Your direct answer if no options are given, as a single phrase or number.
- If your answer is a number, only include the number without any unit.
- If your answer is a word or phrase, do not paraphrase or reformat the text you see in the image.
- You cannot answer that the question is unanswerable. You must either pick an option or provide a direct answer.

IMPORTANT: Remember, to end your answer with Final Answer: <answer>.

A.2 ChartQA

Analyze the image and question carefully, using step-by-step reasoning. First, describe any image provided in detail. Then, present your reasoning. And finally your final answer in this format:
Final Answer: <answer>
where <answer> follows the following instructions:

- <answer> should be a single phrase or number.
- <answer> should not paraphrase or reformat the text in the image.
- If <answer> is a ratio, it should be a decimal value like 0.25 instead of 1:4.
- If the question is a Yes/No question, <answer> should be Yes/No.
- If <answer> is a number, it should not contain any units.
- If <answer> is a percentage, it should include a % sign.
- If <answer> is an entity, it should include the full label from the graph.

IMPORTANT: Remember, to end your answer with Final Answer: <answer>.

A.3 VQAv2

- Answer the question using a single word, number, or short phrase. Use as few words as possible.
- If the answer is a number, report it as a number, i.e. 2, not Two, and only include the number without any unit.
- If the question is Yes/No, answer with Yes/No, and nothing else (no likely, unknown, etc.).
- You cannot answer that the question is unanswerable. You must answer.

A.4 DocVQA

Answer the question using a single word or phrase.

A.5 MM-MT-Bench Judge Prompt

SYSTEM: Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the most recent question given the previous conversation as context. Your evaluation should consider correctness and helpfulness. You will be given a reference answer and the assistant's answer. Begin your evaluation by comparing the assistant's answer with the reference answer. Identify and correct any mistakes. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[rating]", for example: "Rating: [[5]]".

<|The Start of Conversation with User|>

```

### User:
<image> Analyze this image.

### Reference answer:
The image consists of ...

### Assistant:
This is an image of...

<|The End of Conversation with User|>\n\n\n

```

The history of the conversation is passed to the judge with reference answers as assistant answer (teacher-forcing).

B Relative Position Encoding Property of ROPE-2D

In this section, we show the relative position encoding property of ROPE-2D. The goal is prove that:

$$\langle \text{ROPE-2D}(x^{(p,q)}, \Theta), \text{ROPE-2D}(y^{(r,s)}, \Theta) \rangle = \langle \text{ROPE-2D}(x^{(p-r, q-s)}, \Theta), \text{ROPE-2D}(y^{(0,0)}, \Theta) \rangle$$

for any feature $x, y \in \mathbb{R}^d$ for all positions $p, r \in \{0 \dots H\}$ and $q, s \in \{0 \dots W\}$. To keep the discussion simple, we will illustrate this property for $d = 4$ (the extension to higher dimension is straightforward).

$$\text{ROPE-2D}(x^{(p,q)}, \Theta) = \begin{pmatrix} \cos p\theta_1 & -\sin p\theta_1 & 0 & 0 \\ \sin p\theta_1 & \cos p\theta_1 & 0 & 0 \\ 0 & 0 & \cos q\theta_2 & -\sin q\theta_2 \\ 0 & 0 & \sin q\theta_2 & \cos q\theta_2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\text{ROPE-2D}(y^{(r,s)}, \Theta) = \begin{pmatrix} \cos r\theta_1 & -\sin r\theta_1 & 0 & 0 \\ \sin r\theta_1 & \cos r\theta_1 & 0 & 0 \\ 0 & 0 & \cos s\theta_2 & -\sin s\theta_2 \\ 0 & 0 & \sin s\theta_2 & \cos s\theta_2 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

Now, we compute

$$\begin{aligned} & \langle \text{ROPE-2D}(x^{(p,q)}, \Theta), \text{ROPE-2D}(y^{(r,s)}, \Theta) \rangle \\ &= (x_1 \ x_2) \cdot \begin{pmatrix} \cos p\theta_1 & -\sin p\theta_1 \\ \sin p\theta_1 & \cos p\theta_1 \end{pmatrix}^T \begin{pmatrix} \cos r\theta_1 & -\sin r\theta_1 \\ \sin r\theta_1 & \cos r\theta_1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ & \quad + (x_3 \ x_4) \cdot \begin{pmatrix} \cos q\theta_2 & -\sin q\theta_2 \\ \sin q\theta_2 & \cos q\theta_2 \end{pmatrix}^T \begin{pmatrix} \cos s\theta_2 & -\sin s\theta_2 \\ \sin s\theta_2 & \cos s\theta_2 \end{pmatrix} \cdot \begin{pmatrix} y_3 \\ y_4 \end{pmatrix} \\ &= (x_1 \ x_2) \cdot \begin{pmatrix} \cos p\theta_1 \cos r\theta_1 + \sin p\theta_1 \sin r\theta_1 & -\sin r\theta_1 \cos p\theta_1 + \sin p\theta_1 \cos r\theta_1 \\ \sin r\theta_1 \cos p\theta_1 - \sin p\theta_1 \cos r\theta_1 & \cos p\theta_1 \cos r\theta_1 + \sin p\theta_1 \sin r\theta_1 \end{pmatrix}^T \cdot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ & \quad + (x_3 \ x_4) \cdot \begin{pmatrix} \cos q\theta_2 \cos s\theta_2 + \sin q\theta_2 \sin s\theta_2 & -\sin q\theta_2 \cos s\theta_2 + \sin q\theta_2 \cos s\theta_2 \\ \sin q\theta_2 \cos s\theta_2 - \sin q\theta_2 \cos s\theta_2 & \cos q\theta_2 \cos s\theta_2 + \sin q\theta_2 \sin s\theta_2 \end{pmatrix} \cdot \begin{pmatrix} y_3 \\ y_4 \end{pmatrix} \\ &= (x_1 \ x_2) \cdot \begin{pmatrix} \cos((p-r) \cdot \theta_1) & -\sin((p-r) \cdot \theta_1) \\ \sin((p-r) \cdot \theta_1) & \cos((p-r) \cdot \theta_1) \end{pmatrix}^T \cdot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ & \quad + (x_3 \ x_4) \cdot \begin{pmatrix} \cos((q-s) \cdot \theta_2) & -\sin((q-s) \cdot \theta_2) \\ \sin((q-s) \cdot \theta_2) & \cos((q-s) \cdot \theta_2) \end{pmatrix}^T \cdot \begin{pmatrix} y_3 \\ y_4 \end{pmatrix} \\ &= \langle \text{ROPE-2D}(y^{(p-r, q-s)}, \Theta), \text{ROPE-2D}(y^{(0,0)}, \Theta) \rangle \end{aligned}$$

C Flexible Parsing Settings

In Section 4.3, we introduce three ‘parsing levels’ which evaluate models under progressively looser constraints. While common evaluation metrics reward only exactly the answer format in the ground truth annotation, we seek to relax these requirements and investigate how model performance varies.

Baseline: This setting requires exact following of prompt instructions, with model responses ending in the string "Final Answer: <ANSWER>".

Flexible Parsing Level 1: This setting also catches cases where the model ends responses with "Answer: <ANSWER>".

Flexible Parsing Level 2: Here we additionally catch cases where the model has added extra mark-down formatting. We strip markdown such as: "**Answer**", "**Answer:**", "*Answer: <ANSWER>*". We find such formatting to be particularly prevalent in Llama-3.2 models [6].

Flexible Parsing Level 3: This is the most generous evaluation setting. Here we mark a response as correct if the ground truth answer appears *anywhere* in the model’s response. For single letter answers, we search the response for "is <A>", "are <A>", "<A>". For single number responses, we search the response for the number both with and without commas.

We highlight that Flexible Parsing Level 3 is intended to serve as an upper bound, as it may mark incorrect answers as correct.

D Robustness to prompting

D.1 Llama-Specific Prompts

In Section 4.1, we evaluate all models with a common prompt, which allowed us to reproduce the reported figures of GPT-4o [16] and Claude-3.5 Sonnet [1]. This prompt requires models to end responses with "Final Answer: <ANSWER>" (see Appendix A for full prompts).

However, when evaluating Llama-3.2 models [6], we found that this model family defaults to responding with "**Answer:** <ANSWER>" (*i.e.*, with markdown formatting and omission of ‘Final’, despite the explicit instruction). While the performance degradation due to regex mismatches is mitigated through our flexible parsing strategy (see Section 4.3), we found that Llama-3.2 models performed substantially better when the *prompt* specifically asks for "**Answer:** <ANSWER>" (*i.e.*, respecting its default output format).

In Table 6, we show the results for models both with the default prompts from Appendix A, and with the Llama-specific prompts (all evaluated under the *Exact Match* metric). We show that the Llama-specific prompt substantially improves the performance of Llama-3.2 models, particularly for the 11B variant, with over 15% jumps on both Mathvista and MMMU. We further note that Pixtral performance is stable across prompts, and leads the 11B variant by a substantial margin.

		Mathvista	MMMU	ChartQA
		Exact Match	Exact Match	Exact Match
Llama-3.2 11B [6]	Default prompt	24.3	23.0	14.8
	Llama-specific prompt	41.6	41.9	33.7
Llama-3.2 90B [6]	Default prompt	49.1	53.7	33.8
	Llama-specific prompt	57.6	58.6	34.8
Qwen2-VL 7B [23]	Default prompt	53.7	48.1	41.2
	Llama-specific prompt	52.6	47.4	74.0
Pixtral 12B	Default prompt	58.3	52.0	81.8
	Llama-specific prompt	57.7	50.8	83.8

Table 6: Evaluation with Llama-specific prompts. We re-evaluate models with a prompt tailored towards the Llama-3.2 model family [6]. We find that this substantially improves the performance of the 11B variant of the model. Pixtral 12B reports stable performance across both prompts, and maintains a substantial lead over Llama-3.2 11B and Qwen2-VL 7B.

D.2 Average performance across prompts

Here we report average results across a number of prompts. We task Mistral Large v2 with creating 10 versions of the prompt used in the main paper (see Appendix A), with varied wording while keeping instructions explicit. As prior works suffer under stricter parsing constraints, all models are evaluated under ‘Flexible Parsing Level 3’ for this experiment (see Section 4.3 and Appendix C).

We find that the trends follow those from the main paper, with Pixtral outperforming models of comparable size, and surpassing Llama-3.2 90B [6] on Mathvista and ChartQA. Pixtral also typically displays lower variance in performance between prompts (shown in gray).

	Mathvista	MMMU	ChartQA
	Flexible Level 3	Flexible Level 3	Flexible Level 3
Llama-3.2 11B [6]	42.1 (± 1.9)	45.3 (± 1.0)	77.2 (± 0.8)
Llama-3.2 90B [6]	56.0 (± 1.5)	56.7 (± 0.5)	80.1 (± 0.5)
Qwen2-VL 7B [23]	53.7 (± 2.1)	46.9 (± 1.9)	77.0 (± 0.8)
Pixtral 12B	56.4 (± 1.0)	49.5 (± 1.5)	83.8 (± 0.4)

Table 7: Average multimodal performance across prompts. We evaluate models with 10 different prompts, reporting the mean performance, and standard deviations in gray. The trends follow those in the main paper, with Pixtral outperforming open-source models of a comparable size. All models are evaluated with ‘Flexible Level 3’ parsing (see Section 4.3)

E Reproducing Reported Numbers

In Section 4.1 we re-evaluate all models under a common and rigorous protocol. All models are evaluated under the same evaluation metric and with the same prompt, in such a way that frontier models achieve their reported performance.

Under this common protocol, we found some models substantially underperformed their reported figures. Here, we document the steps required to recover the reported figures of open models, by tuning the evaluation prompt and metric to each model in turn. All results are shown in Table 8.

E.1 Summary

Our analysis indicates that frontier models, and even smaller closed-source models, are able to recover or exceed their reported figures under the common protocol discussed in Section 4.1. This is achieved through precise following of instructions in the ‘Explicit’ prompts (see Appendix A).

Smaller, open-source models typically require some degree of prompt tuning and/or adjustment of the evaluation metric, targeted towards the model, to recover reported performance. With such interventions, we are generally able to recover or exceed reported figures.

Pixtral 12B, like closed and leading models, is able to follow prompt instructions to report strong performance without targeted interventions. This is substantiated in robust performance across prompts (see Appendix D), as well as strong performance in both LMSys Vision Arena and MM-MT-Bench (see Section 4.1).

E.2 Closed models: Claude-3 Haiku and Gemini-Flash-8B

We find we the standardized evaluation protocol roughly matches or exceeds reported figures, with a small gain achieved through flexible parsing. The only exception is for Claude Haiku [1] on ChartQA, where Flexible Parsing Level 3 is required to approach reported performance.

E.3 Qwen2-VL 7B

We first simplify the prompt into a one-line instruction, similar to the training set of ChartQA. Next, we provide different prompts depending on the answer format expected. For instance, if the answer is a floating point number, we specify "Answer with a two decimal place floating point",

with analogous prompts for integer and multiple-choice questions. We found that providing a single, unified prompt with all format specifications (as in the prompts in Appendix A) reduces performance.

E.4 Llama-3.2

We find that these models default to responses with markdown formatting such as: `***Answer***`, `***Answer:**`, `*Answer: <ANSWER>*`. We find substantial improvement by changing the ‘Explicit’ prompt to request this format (see Appendix D). These models then recover their reported performance after evaluating with Flexible Level 3.

When evaluating Llama-3.2 90B on DocVQA, many generations are of the form ‘The answer is <ANSWER>’, which is penalized by the ANLS metric. We strip such prefixes, and this improves DocVQA by +4.8.

E.5 Llava-OneVision 72B

Similarly to Qwen2-7B [23], we first simplify the prompt into a one-line instruction and provide different prompts depending on the answer format expected. We found that providing a single, unified prompt with all format specifications reduces performance.

E.6 Molmo

Similarly to Qwen2-7B [23] and Llava-Onevision 7B [9], we first simplify the prompt into a one-line instruction, and provide different prompts depending on the answer format expected. Furthermore, similarly to the intervention for Llama-3.2 [6], we reformat the prompt and relax the evaluation metrics. Molmo models default to ending long responses with `\n\n<ANSWER>`. In long-answer cases, we adjust the evaluation metric to capture this.

For VQAv2, we apply custom post-processing filters, such as remapping textual output of numerical answers to the integer digits (*e.g.* Two to 2).

	Mathvista CoT	MMMU CoT	ChartQA CoT	DocVQA ANLS	VQAv2 VQA Match	MM-MT-Bench GPT-4o Judge	LMSys-Vision (Oct '24)
Pixtral 12B	58.3	52.0	81.8	90.7	78.6	6.05	1076
Qwen-2-VL 7B [23]							
Measured (Exact Match)	53.7	48.1	41.2	94.5	75.9	5.45	
Measured (Custom evaluation, see Section E.3)	63.7	50.6	83.4	94.5	82.1	-	1040
Reported	58.2	54.1	83.0	94.5	-	-	
Llama-3.2 11B [6]							
Measured (Exact Match)	24.3	23.0	14.8	91.1	67.1	4.79	
Measured (Custom evaluation, see Section E.4)	47.9	46.6	78.5	91.1	67.1	-	1032
Reported	51.5	50.7	83.4	88.4	75.2	-	
Molmo-D 7B [4]							
Measured (Exact Match)	12.3	24.3	27.0	72.2	57.1	3.72	
Measured (Custom evaluation, see Section E.6)	43.2	47.0	76.7	72.2	70.0	-	-
Reported	51.6	45.3	84.1	92.2	85.6	-	
LLaVA-OneVision 7B [9]							
Measured (Exact Match)	36.1	45.1	67.2	90.5	78.4	4.12	
Measured (Custom evaluation, see Section E.5)	63.1	48.1	80.2	90.5	83.7	-	-
Reported	63.2	48.8	80.0	87.5	-	-	
Molmo 72B [4]							
Measured (Exact Match)	52.2	52.7	75.6	86.5	75.2	3.51	
Measured (Custom evaluation, see Section E.6)	61.3	52.9	82.3	86.5	75.5	-	-
Reported	58.6	54.1	87.3	93.5	86.5	-	
Llama-3.2 90B [6]							
Measured (Exact Match)	49.1	53.7	33.8	85.7	67.0	5.50	
Measured (Custom evaluation, see Section E.4)	57.5	60.2	91.7	91.5	67.0	-	1071
Reported	57.3	60.3	85.5	90.1	78.1	-	
Claude-3 Haiku [1]							
Measured (Exact Match)	44.8	50.4	69.6	74.6	68.4	5.46	
Measured (Custom evaluation, see Section E.2)	44.8	51.3	79.8	74.6	68.4	-	1000
Reported	46.4	50.2	81.7	88.8	-	-	
Gemini-1.5-Flash 8B₍₀₈₂₇₎ [18]							
Measured (Exact Match)	56.9	50.7	78.0	79.5	65.5	5.93	
Measured (Custom evaluation, see Section E.2)	57.1	50.7	78.2	79.5	69.2	-	1111
Reported	-	50.3	-	73.6	-	-	

Table 8: Reproducing the reported performance of prior models. In Table 2 we conduct fair re-evaluation of all models through the same evaluation harness, with the same prompt and metric. Here, we endeavour to recover the reported performance of all models by tuning evaluation settings towards individual models. We highlight that Pixtral 12B, like strong closed-source models (e.g. Gemini-1.5-Flash 8B [18] and Claude-3 Haiku [1]) is able reports strong performance without such interventions.