Reading, Not Thinking: Understanding and Bridging the Modality Gap When Text Becomes Pixels in Multimodal LLMs

Kaiser Sun[*]
Johns Hopkins University
hsun74@cs.jhu.edu

Xiaochuang Yuan[*]
Amazon

Hongjun Liu[*]
New York University

Chen Zhao
New York University

Cheng Zhang
Texas A&M University

Mark Dredze
Johns Hopkins University

Fan Bai
Johns Hopkins University

Abstract

Multimodal large language models (MLLMs) can process text presented as images, yet they often perform worse than when the same content is provided as textual tokens. We systematically diagnose this "modality gap" by evaluating seven MLLMs across seven benchmarks in five input modes, spanning both synthetically rendered text and realistic document images from arXiv PDFs to Wikipedia pages. We find that the gap is highly sensitive to rendering choices such as font and resolution, and that natural document images often exhibit much smaller gaps, suggesting the performance difference partly reflects evaluation artifacts rather than fundamental limitations. Through a grounded-theory error analysis of over 5,000 examples, we identify the primary cause: image input alone suppresses reasoning effort, with most models producing 8.6–85$\times$ shorter outputs that skip step-by-step computation or reasoning. The reluctance to reason, not a failure of perception or knowledge retrieval, drives the performance gap, particularly on tasks requiring multi-step reasoning. We show that a simple lightweight self-distillation method by fine-tuning models on their own text-mode reasoning traces paired with image inputs closes this gap, raising image-mode accuracy to match or exceed text-mode performance with over 50% improvement, and the gains transfer to unseen benchmarks without catastrophic forgetting. Overall, our results and analyses provide a systematic understanding of the modality gap and suggest a practical path toward improving visual text understanding in multimodal language models. Code and data: `https://github.com/KaiserWhoLearns/ReadingNotThinking`.

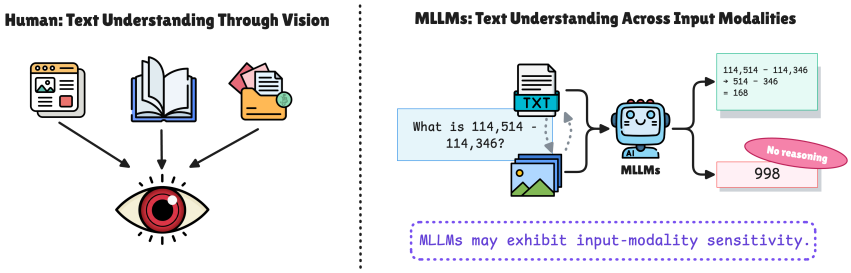


Figure 1: Humans read text visually across diverse sources (e.g., books, webpages, documents), whereas MLLMs may produce different predictions for identical content presented in different formats.

[*]Equal contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

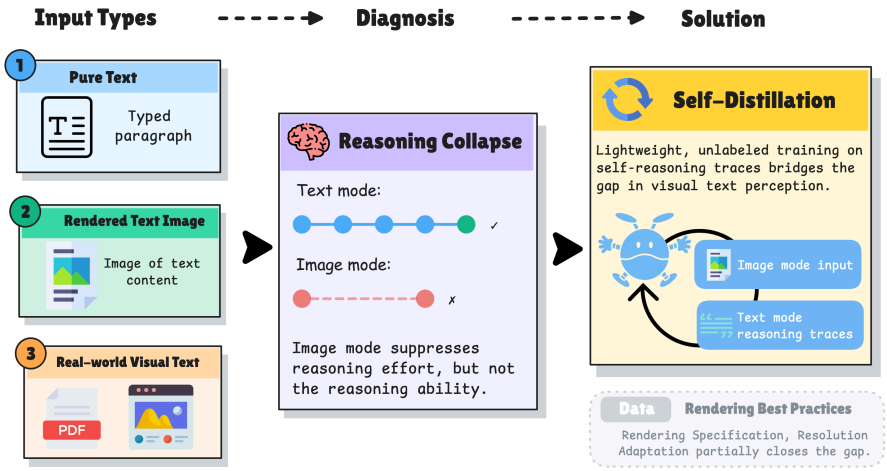


Figure 2: Diagnosing the modality gap in visual text understanding. We evaluate MLLMs across five input modes, including pure text, rendered text images, real-world visual text, and two OCR-based diagnostic settings (`OCR-1P` and `OCR-2P`). Our error taxonomy reveals that while image modality amplifies reading and calculation errors, it leaves underlying reasoning capabilities largely intact but also reduces the likelihood of triggering a reasoning chain. We further propose remedies to bridge the performance gap, including rendering specification control, resolution-aware preprocessing, and LM-only self-distillation.

1 Introduction

Humans read text through vision: we perceive words as visual patterns on pages, screens, and signs, not as sequences of abstract symbols [1, 2]. Modern multimodal large language models (MLLMs), however, process text and images through separate pathways, compressing visual input into tokens before fusing it with textual representations [3, 4, 5, 6, 7]. When the same textual content is presented as an image rather than as text tokens, these models often produce different, and frequently worse, predictions (Figure 1). This "modality gap" has been documented across math, coding, and knowledge tasks [8, 9], and motivates a growing line of work on visual text processing, including screenshot language models [10, 11], vision-centric tokenization [12, 13], and unified multimodal architectures [14].

Yet existing work predominantly focuses on either *documenting* the gap or *building new systems* to circumvent it, without systematically explaining *why* the gap arises in current MLLMs, characterizing *when* visual input helps versus hinders, or showing *how* the gap can be closed with minimal intervention. A further limitation is that existing evaluations rely almost exclusively on *synthetically rendered* text images, potentially confounding rendering artifacts with genuine perceptual deficits.

In this work, we diagnose the modality gap and show it can be substantially narrowed with a simple training recipe. We evaluate seven MLLMs across seven benchmarks in five input modes (Figure 2), spanning synthetically rendered text and realistic document images from arXiv PDFs and Wikipedia pages. Two OCR-based modes decompose the visual pipeline into reading and reasoning stages, isolating their respective contributions to the gap.

We first establish that the gap is task- and data-dependent. On synthetic benchmarks, math tasks show gaps exceeding 60 points while knowledge-intensive tasks show modest drops. On natural document images, MLLMs show much smaller performance differences and sometimes even exceed their text-mode performance. Rendering choices such as font and resolution are critical confounds, with font alone swinging accuracy by up to 38 percentage points. These findings suggest the gap partly reflects distributional mismatch rather than fundamental perceptual limitations.

To understand *why* image mode degrades performance, we conduct a grounded-theory error analysis of over 5,000 errors. We find that image mode triggers a reasoning collapse: models produce substantially shorter outputs, skipping step-by-step computation entirely. The reluctance to reason in image mode, rather than a failure of visual perception or knowledge retrieval, is the major cause of the performance gap. The error taxonomy confirms this: calculation and format errors increase under image input, while conceptual and reasoning errors remain unchanged.

Motivated by this diagnosis, we propose a self-distillation approach: training the model on its own text-mode reasoning traces paired with image inputs. This simple approach, requiring only lightweight fine-tuning without architectural changes, raises image-mode accuracy to match or exceed text-mode performance. We validate self-distillation on two open-source architectures, and show the gains transfer to unseen benchmarks without catastrophic forgetting. Overall, our contributions are:

- We provide a comprehensive evaluation across seven models, seven benchmarks, and five input modes, showing that the gap is task-dependent and that rendering choices are critical confounds.
- We identify reasoning collapse as the root cause of the modality gap: models are far less likely to produce step-by-step reasoning when the input is visual, leading to cascading errors.
- We propose a lightweight but effective solution that closes the gap by training on text-mode reasoning traces paired with image inputs, without requiring architectural changes or external data.

2 Related Work

Visual Text Processing. A growing body of work processes language through visual representations. Screenshot language models [10, 11] train on rendered text with masked patch prediction, while vision-centric tokenization [12, 13] compresses context by 3–4$\times$ via rendering. Unified architectures such as Emu3 [14] perform next-token prediction over a shared image-text space. These works focus on *building better systems*; in contrast, we diagnose *why* existing MLLMs fail on visual text and show the gap can be narrowed with minimal training.

Evaluating MLLMs on Visual Text. Unlike scene-text understanding [15, 16, 17], which tests recognition in natural images, recent work evaluates *reasoning* over visually presented text. PixelWorld [8] and VISTA-Bench [18] document task-dependent gaps, while Li et al. [19] show that rendering long contexts as images can halve token counts. Concurrent work, Sprang et al. [9], measure cross-modal consistency but do not explain *why* inconsistency arises or how to reduce it. We go beyond measurement: we evaluate on realistic documents (Wikipedia screenshots, arXiv PDFs), point out the anomaly within existing evaluations, provide an error taxonomy identifying reasoning collapse as the root cause, and propose self-distillation to close the gap.

Closing the Modality Gap. While the works above establish that a text-versus-image gap exists, relatively little work addresses *how to close* the gap in existing MLLMs. Wang et al. [20] show architecture-dependent effects of embedding text in images but propose no remedy; Hu et al. [21] achieve strong ARC performance with visual priors alone but require training from scratch. Our self-distillation trains the model on its own text-mode reasoning traces paired with image inputs. This connects to cross-modal distillation [22, 23], but uniquely uses the *same* model as both teacher (text mode) and student (image mode).

3 Evaluation Setup

3.1 Datasets

As discussed in Section 2, our focus is on *reasoning* over visually presented text rather than scene-text recognition. We include the following datasets:

Synthetic-image datasets. Five benchmarks from prior visual text evaluations exhibit clear modality gaps, allowing us to examine their source. **MMLU** [24, 25] contains multiple-choice questions across 57 knowledge domains. **ARC** [26] includes challenging grade-school science questions. **GPQA** [27] (main set) contains 448 graduate-level science questions that experts answer correctly but non-experts often fail. **GSM8K** [28] consists of grade-school math word problems. **HumanEval** [29] evaluates Python code generation from natural-language descriptions. Because these tasks are purely textual, their image counterparts are synthetically rendered.

Natural-image datasets. Synthetic renderings may not reflect the visual distribution of real-world text. We therefore include two datasets whose image inputs come from natural sources. **QASPER** [30] is a long-form QA dataset over scientific articles that often requires long context reasoning.[2] We

[2]We remove questions that depend on tables and figures for a fair comparison across modalities. Results on the full test set are in the appendix.

obtain the original PDF files from arXiv and convert relevant pages into PNG images using the *fitz* library.[3] **SQuAD (v2)** [31] is an extractive QA dataset built over Wikipedia. For each example, we capture a screenshot of the corresponding Wikipedia page to create a natural-image version of the context. Additional details on dataset processing and image construction are provided in the appendix.

Evaluation metrics. We follow the standard metric for each dataset, using pass@k for HumanEval and accuracy for the rest. For QASPER and SQuAD, zero-shot MLLM inference does not align well with span-based metrics, because MLLMs are not trained to reproduce annotation artifacts such as exact spans. Following recent information-seeking benchmarks [32, 33], we adopt an LLM-as-judge protocol, where we use GPT-5 (*gpt-5-2025-08-07*) to compare the model prediction with the gold answer and report accuracy. Results using span-based F1 are provided in the appendix.

3.2 Input Modalities

We compare five input modalities, with examples shown in the appendix. Prior work has examined `Pure Text`, `Pure Image`, and `Instr.+Image` separately; we place them in a unified comparison and additionally introduce two OCR-based settings that help isolate whether the modality gap stems from failures in *reading* text from pixels, failures in *reasoning* over the extracted content, or both.

`Pure Text.` The model receives only the original textual content (e.g., question, answer options, passages, or function signatures), preceded by a task instruction when necessary. This serves as our reference modality.

`Pure Image.` The identical textual content from `Pure Text` is rendered into a 1280×720 image (white canvas, black text) following prior work [8, 34]; the model receives only this image. For natural-image datasets, the task instruction is rendered and concatenated at the top of the document image. Section 4 examines how rendering parameters (font, resolution, compression) affect performance.

`Instr.+Image.` The model receives the example content as an image and the task instruction as text. This setting closely mirrors traditional VQA-style inputs, where language instructions are provided textually while the model must read text embedded in the image.

`OCR-1P (one pass).` The model receives the same image as in `Pure Image` (with the task instruction rendered in it), plus a text prompt that asks it to first extract all text from the image and then solve the task, all within a single inference pass. This tests whether explicitly prompting for text extraction improves downstream performance while still coupling reading and reasoning.

`OCR-2P (two passes).` In this two-stage pipeline, the model first receives the image and is prompted *only* to extract the textual content. The extracted text is then fed back into the model as a new `Pure Text` query to solve the task. By fully separating reading from reasoning, this setting isolates whether errors originate from the vision-based text extraction or from downstream task execution.

3.3 Models

We evaluate seven MLLMs selected to span different architectures, model scales, and access types. Our selection includes six open-source models: Qwen2.5-VL (7B and 32B) [35], Qwen3-VL-8B [36], InternVL3-8B [37], InternVL3.5-8B [38], and Pixtral-12B [39], as well as the proprietary GPT-5.2 (*gpt-5.2-2025-12-11*). The open-source models share a common architecture in which a ViT-based visual encoder converts input images into visual tokens that are fed into an autoregressive LLM decoder. All models receive identical inputs for each modality, and inference parameters are listed in the appendix.

4 Data Characteristics and Rendering Confounds

4.1 Modality Gap on Synthetic and Natural Images

[3] `https://pypi.org/project/fitz/`

[4] OCR-2P is omitted for QASPER due to the computational cost of running OCR on extremely long documents. Pixtral-12B is excluded from QASPER due to insufficient context length.

Dataset	Model	Pure Text (T)	Instr.+ Image (T I)	Pure Image (I)	OCR-1P (I→T)	OCR-2P (I→T)	Dataset	Model	Pure Text (T)	Instr.+ Image (T I)	Pure Image (I)	OCR-1P (I→T)	OCR-2P (I→T)
MMLU	GPT-5.2	92.33	91.06†	90.93†	86.54†	86.72†	GPQA	GPT-5.2	81.25	81.25	80.13	77.90	80.58
	InternVL3-8B	59.83	69.09†	33.93†	60.40	71.86†		InternVL3-8B	33.71	32.14	32.81	30.58	33.26
	InternVL3.5-8B	82.97	82.25	80.74†	77.65†	80.75†		InternVL3.5-8B	42.63	40.40	34.60†	40.18	46.21
	Pixtral-12B	37.89	41.60†	42.76†	49.54†	44.25†		Pixtral-12B	31.25	30.36	27.23	29.69	30.36
	Qwen2.5-7B-VL	67.80	67.97	67.36	65.46†	67.08		Qwen2.5-7B-VL	34.15	34.15	30.58	29.02	32.59
	Qwen2.5-32B-VL	78.34	75.47†	73.79†	76.55†	80.38†		Qwen2.5-32B-VL	42.41	41.52	43.97	43.08	43.30
	Qwen3-VL-8B	78.44	72.29†	71.01†	72.98†	73.61†		Qwen3-VL-8B	43.75	39.29	35.04†	37.28	36.38
ARC	GPT-5.2	96.25	92.24†	95.48	97.70	97.18	GSM8K	GPT-5.2	95.83	85.14†	96.51	77.86†	68.16†
	InternVL3-8B	92.24	91.30	90.27	86.26†	90.53		InternVL3-8B	78.01	49.51†	42.53†	73.92	87.87†
	InternVL3.5-8B	92.66	92.41	92.66	93.26	93.52		InternVL3.5-8B	95.30	95.22	95.30	95.60	94.47
	Pixtral-12B	84.64	71.25†	61.77†	72.10†	39.33†		Pixtral-12B	3.79	4.93	8.11†	75.89†	55.12†
	Qwen2.5-7B-VL	87.20	85.92	89.42	87.71	86.35		Qwen2.5-7B-VL	19.48	20.24	24.79†	20.09	20.17
	Qwen2.5-32B-VL	93.09	91.30	90.02†	94.62	94.37		Qwen2.5-32B-VL	95.07	4.47†	94.92	66.41†	96.13
	Qwen3-VL-8B	91.72	91.38	91.04	92.92	92.75		Qwen3-VL-8B	93.56	39.58†	30.71†	56.94†	94.77

Table 1: Performance of different input modalities on four synthetic-image datasets (MMLU, ARC, GPQA, GSM8K). Cell shading indicates the performance difference relative to `Pure Text`: red indicates the modality outperforms text, blue indicates it underperforms text, and stronger shading reflects larger gaps. All HumanEval results throughout this paper are reported in Section E due to small dataset size and high pass@1 variance [29]. †: significant difference from Pure Text (Holm-adjusted $p < 0.05$; Appendix R).

Dataset	Model	Pure Text (T)	Instr.+Image (T I)	Pure Image (I)	OCR-2P (I→T)	OCR-1P (I)
QASPER	GPT-5.2	51.92	75.10	77.25	-	74.27
	Qwen2.5-7B-VL	30.49	66.32	64.38	-	57.05
	Qwen3-VL-8B	53.27	76.25	70.75	-	56.89
	InternVL3-8B	44.02	69.36	54.43	-	56.15
	InternVL3.5-8B	49.38	57.66	61.07	-	55.68
SQuAD v2	GPT-5.2	97.50	85.50	94.00	85.50	85.50
	Qwen2.5-7B-VL	80.00	82.50	83.50	77.00	81.50
	Qwen3-VL-8B	95.00	85.50	84.50	85.50	84.00
	InternVL3-8B	82.50	84.00	73.50	76.50	69.50
	InternVL3.5-8B	98.00	83.50	84.00	86.00	79.50
	Pixtral-12B	91.50	81.00	83.50	82.50	84.50

Table 2: Performance of different input modalities on two real-image datasets QASPER and SQuAD v2. Cell shading indicates the performance difference relative to `Pure Text`: red indicates the modality outperforms text, blue indicates it underperforms text. Image-based inputs such as `Pure Image` and `Instr.+Image` outperform `Pure Text` on QASPER, indicating that MLLMs can benefit from naturally occurring visual text.[4]

Performance on synthetic images. Table 1 reports results on four synthetic-image datasets, with cell shading indicating the performance difference relative to `Pure Text` (red = better than text, blue = worse). On MMLU, `Pure Text` outperforms `Pure Image` for six of seven models. GPQA shows the same direction for six of seven models (Appendix R). On ARC, models still perform better on text, but the gap narrows: Qwen2.5-VL-7B achieves 89.42% under `Pure Image`, and InternVL3.5-8B ties its text baseline. GSM8K reveals the largest modality gaps overall, with drops exceeding 60 points for several models. HumanEval results are reported separately in Section E due to the high variance inherent in pass@1 estimation from only 164 instances [29].

Performance on natural images. Table 2 reports results on QASPER and SQuAD, where image inputs come from naturally occurring documents rather than synthetic renderings. On QASPER, the trend reverses: almost all models experience a performance gain when switching from text to image. These gains likely reflect the alignment between real PDF images and the document-heavy pretraining data of modern MLLMs. On SQuAD, the behavior is more model-dependent: Qwen2.5-VL-7B benefits from images, whereas GPT-5.2 sees a much larger drop. On SQuAD, rendering the original context under `Instr.+Image` changes accuracy by $+5.1$ percentage points on average across seven models relative to `Pure Text`, so an image input need not incur a performance penaltys.

Divergent robustness across models. As shown by the cell shading in Tab. 1 and Tab. 2, MLLMs exhibit sharply different responses to modality changes. InternVL3.5-8B is the most modality-robust model, maintaining near-zero or positive differences on nearly all benchmarks. Qwen2.5-VL-7B also shows strong robustness, with image often being its better-performing modality. These discrepancies may stem from differences in vision encoder capability; we therefore use the OCR-1P and OCR-2P settings to measure recognition quality directly.

4.2 Image Recognition Cannot Fully Explain the Modality Gap

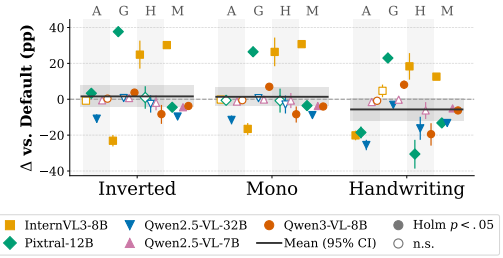


Figure 3: Accuracy change of each rendering relative to the default rendering. Each point is a dataset–model pair (A/G/H/M = ARC/GSM8K/HumanEval/MMLU) with its 95% paired bootstrap CI; filled points are significant (Holm-corrected $p < 0.05$). Handwriting is the only rendering with a negative mean change.

The two OCR settings in Table 1 reveal an asymmetric pattern across tasks. `OCR-1P` matches or exceeds `Pure Image` in 16 of 28 cells, with the largest gains where `Pure Image` collapses (Pixtral-12B on GSM8K: 8.11 to 75.89; InternVL3-8B on MMLU: 33.93 to 60.40), although it hurts the strongest models on GSM8K (GPT-5.2: 96.51 to 77.86). `OCR-2P` likewise recovers a portion of the performance: InternVL3-8B increases from 42.53 to 87.87 on GSM8K when switching from `Pure Image` to `OCR-2P`, and Pixtral-12B sees a 42-point improvement. To quantify this relationship, we compute the correlation between `OCR-2P` word error rate and `OCR-2P` accuracy across all models and tasks, obtaining only $r = 0.163$. This suggests that overall OCR quality is a poor predictor of image-mode performance; errors on a small number of task-critical tokens (e.g., operators, variable names) may matter more than aggregate recognition accuracy.

4.3 Rendering Choices Confound Visual Evaluation

The results above show that MLLMs struggle with synthetic renderings yet perform well on natural document images, suggesting a distributional mismatch between evaluation images and pretraining data.

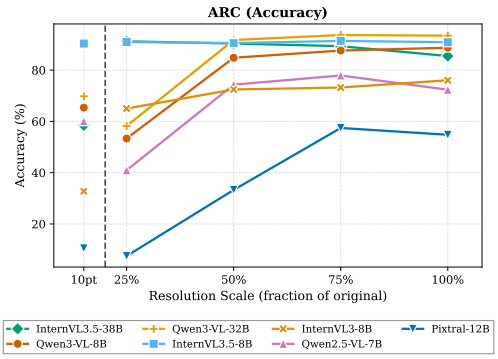


Figure 4: Performance vs. image resolution scale on ARC. The black vertical dashed line indicates the point where `Pure Image` consumes the same FLOPs as `Pure Text`. Other datasets are in Section G.

Font choice matters. We render text in four styles: default (NotoSansMath), inverted colors, monospaced (DejaVuSans Mono), and handwriting (Priestacy). Figure 3 shows that font choice can swing accuracy by up to 38 percentage points, with handwriting causing the largest degradation. Paired comparisons against the default rendering are significant in 38 of 60 cases after Holm correction across all 60 tests ($p_{\text{adj}} < 0.05$): 12/20 for inverted colors, 11/20 for monospaced text, and 15/20 for handwriting (Appendix F). This directly contradicts Sprang et al. [9], who report no effect of font on cross-modal consistency. The discrepancy likely reflects our focus on *task accuracy* rather than response consistency: fonts common in pretraining (inverted, monospaced) cause minimal harm, while out-of-distribution fonts (handwriting) severely degrade performance.

To assess whether the handwriting degradation reflects difficulty reading the text, we separately test transcription on 50 instances per model–dataset pair for two models and three datasets (Appendix L.1). WER is only 0.9–6.1% and CER 0.5–1.7%, with no significant correlation between per-item WER and image-mode answer correctness in any pair (all $p \geq 0.13$). These results suggest that recognition errors alone do not explain the handwriting degradation.

Resolution sensitivity. Figure 4 shows that most models maintain accuracy at 0.50×–1.0× resolution but degrade sharply below this threshold. InternVL3.5, with its Visual Resolution Router [38], is uniquely resolution-invariant. Notably, a compact 10pt rendering outperforms higher-resolution counterparts, suggesting that smart compression preserving critical features can mitigate resolution sensitivity. Overall, font, color, and resolution are strong confounders that benchmarks should report alongside results.

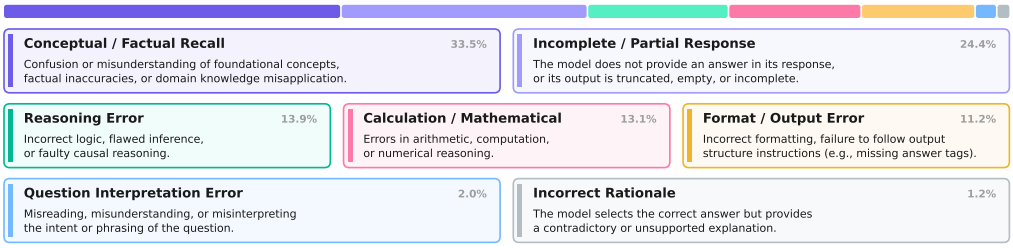


Figure 5: Error taxonomy derived from grounded-theory coding of 5,245 errors across all models and datasets. Percentages indicate overall frequency.

5 Error Analysis

The aggregate performance numbers in Section 4 show that image mode degrades accuracy, but not *why*. To move beyond surface-level accuracy comparisons, we conduct a systematic error analysis following *grounded theory* [40]. We classify over 5,000 individual errors across all models and six datasets under three input modes (`Pure Text`, `Pure Image`, `Instr.+Image`).

5.1 Method

Grounded theory [41, 40] is a widely adopted qualitative research methodology that has been a standard approach in sociology, healthcare, and education for over five decades. Unlike hypothesis-driven methods, it derives theoretical frameworks inductively from data through iterative coding phases: *open coding*, which segments raw observations into initial descriptive labels; *constant comparison*, which refines codes by classifying new data against existing categories; and *axial coding*, which groups related codes into higher-level categories. Following Nelson's [42] computational grounded theory framework, which combines human interpretive skills with computational pattern recognition for rigorous and reproducible analysis, we adopt a human-in-the-loop variant. Recent work has validated that LLM-assisted coding achieves high agreement with human experts when humans review model outputs [43, 44, 45, 46]. In our implementation, GPT-5.2 performs the open coding and constant comparison while human annotators review *every* proposed change and conduct the final axial coding to ensure coherence and validity.

Phase 1: Open coding. We sample 1,000 errors uniformly across models, datasets, and modalities. For each error, the model receives the original input (question and answer choices where applicable), ground-truth answer, and the model's raw response, and proposes a short failure description along with a candidate error label. The model maintains a running set of codes and can either assign an instance to an existing code or propose a new one. This phase produced 415 initial codes.

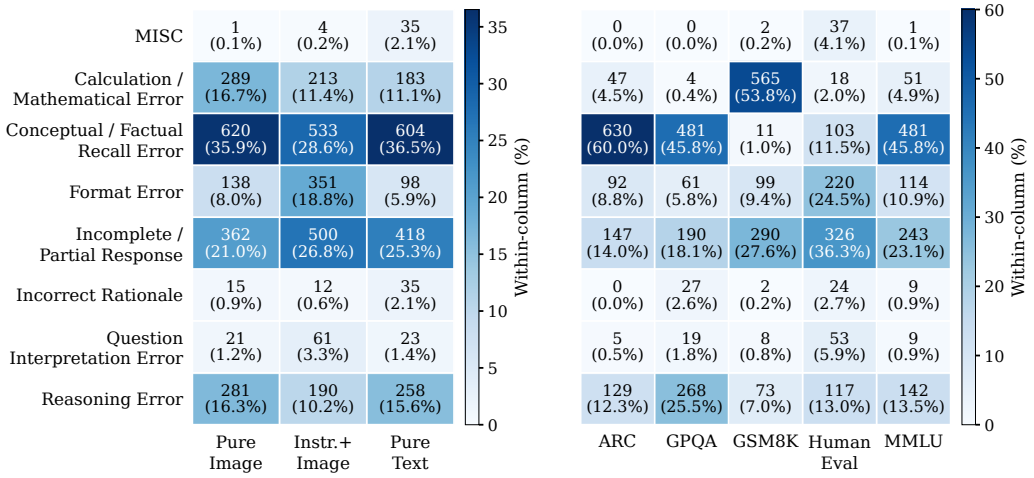
Phase 2: Constant comparison. New errors are randomly sampled and classified against existing codes. For each instance, the model determines whether the error matches an existing code, requires a new code, or warrants updating an existing description. A human annotator reviews every proposed new code, merge, or description update, approving or rejecting changes. Saturation is reached when no new codes are added for 50 consecutive instances. Codes occurring only once are pruned unless conceptually critical, and the corresponding errors are assigned to a miscellaneous category.

Phase 3: Axial coding. A human expert reviews all codes and groups them into organizing dimensions, producing the final taxonomy of primary error categories along with a miscellaneous category.

5.2 Error Taxonomy and Findings

The coding process above yields the taxonomy shown in Figure 5; an inter-annotator agreement study validates that this taxonomy can be reliably applied with moderate to substantial agreement ($\kappa = 0.57$–0.69, Appendix O). We then classify up to 150 errors per model–dataset pair, yielding 5,245 errors across 40 pairs.

Shortened reasoning under image input. The key finding is that models are far less likely to produce long-form reasoning in image mode. On GSM8K, Qwen3-VL-8B averages 617 characters per response in text mode but only 32 in image mode; Pixtral averages 115 characters in text mode but only 22 in image mode. Rather than working through problems step-by-step, models in image mode



(a) Error distribution by input mode.

(b) Error distribution by dataset.

Figure 6: Distribution of error categories across input modes (left) and datasets (right). Cell values show raw counts and within-column percentages.

jump directly to answers or exit reasoning early, often resulting in incorrect answers (see Figure S11 in Appendix for an example). This reluctance to reason explicitly has cascading consequences: without intermediate computation steps, arithmetic mistakes go undetected and uncorrected, and the model loses the self-verification that chain-of-thought reasoning provides.

Comparing error distributions between image mode and text mode (Figure 6a), we observe a consistent pattern: *Calculation errors* make up a larger share of errors under image input (16.7% vs. 11.1%), and *format errors* rise from 5.9% to 8.0%. These are precisely the error types that reasoning helps prevent: step-by-step computation catches arithmetic mistakes, and longer outputs provide more opportunity to follow formatting instructions. In contrast, *conceptual/factual recall errors* and *reasoning errors* remain virtually unchanged. This asymmetry indicates that image mode does not impair the model's knowledge or logical capabilities; rather, it suppresses the reasoning process that would normally be applied.

Dataset-specific error profiles. Output lengths before and after self-distillation are reported in Appendix Q; benchmark-matched error comparisons are in Appendix R. The error distribution varies across tasks (Figure 6b), reflecting their different requirements. GSM8K errors are dominated by calculation failures, consistent with reasoning collapse having the largest impact on multi-step arithmetic. ARC, GPQA, and MMLU show high rates of conceptual/factual recall errors, reflecting the knowledge-intensive nature of these benchmarks. HumanEval exhibits a distinct profile with high rates of incomplete responses (36.3%) and format errors (24.5%), suggesting code generation is potentially sensitive to the output truncation that accompanies the reasoning collapse.

These findings establish that the modality gap is primarily driven by the reluctance to reason when only images are provided: image mode suppresses step-by-step reasoning, leading to cascading errors in calculation and formatting. In Section 6, we show that training models to restore reasoning traces from image inputs substantially closes the gap.

6 Bridging the Modality Gap via Self-Distillation

Our error analysis (Section 5.1) identified reasoning collapse as the root cause of the modality gap: models produce dramatically shorter outputs in image mode, skipping step-by-step computation. Our primary focus is training-time alignment with text-mode reasoning, which directly addresses the model's tendency to suppress reasoning under visual input. For users with limited training resources, adding a chain-of-thought (CoT) prompt to the image offers a simple inference-time alternative, which we evaluate on a subset of model–benchmark pairs in Appendix M. To restore reasoning under visual input, we propose *self-distillation*: training the model on its own text-mode reasoning traces paired with image inputs, teaching it to produce the same multi-step reasoning regardless of input modality.

We select GSM8K and MMLU because they exhibit large modality gaps and provide standard training splits. For each model, we run inference in `Pure Text` mode to collect reasoning traces, retaining only those with correct final answers. These traces serve as supervised targets, each paired with the corresponding `Pure Image` input. Unlike standard distillation, which transfers soft labels, our setting requires transferring multi-step reasoning traces; ground-truth answers alone are insufficient since they lack the intermediate steps. All experiments apply LoRA [47] to the language model (freezing the vision encoder), with rank r=64, trained for 2 epochs with a learning rate of 2×10^{-4} and an effective batch size of 16.

		Base		Distilled					Base		Distilled		
Train	**Eval**	T	I	T	I	$\lvert\Delta\rvert$	**Train**	**Eval**	T	I	T	I	$\lvert\Delta\rvert$
		Qwen3-VL-8B							**InternVL3-8B**				
GSM8K	GSM8K	93.56	30.71	**94.09**	**92.72**	$62.85^{\dagger}\rightarrow$**1.37**‡	GSM8K	GSM8K	78.01	42.53	**90.30**	**90.14**	$35.48^{\dagger}\rightarrow$**0.16**‡
	ARC	91.72	91.04	91.47	91.89	0.68→0.42		ARC	92.24	90.27	91.38	89.33	$1.97\rightarrow2.05^{\dagger}$
	MMLU	78.44	71.01	75.49	72.70	$7.43^{\dagger}\rightarrow2.79^{\dagger\ddagger}$		MMLU	59.83	33.93	**75.16**	68.00	$25.90^{\dagger}\rightarrow7.16^{\dagger\ddagger}$
	MMMU-Pro*	60.46	45.14	58.55	44.51	–		MMMU-Pro*	48.09	33.35	**54.51**†	32.54	–
MMLU	GSM8K	93.56	30.71	93.63	85.06	$62.85^{\dagger}\rightarrow8.57^{\dagger\ddagger}$	MMLU	GSM8K	78.01	42.53	87.95	85.75	$35.48^{\dagger}\rightarrow2.20$
	ARC	91.72	91.04	91.21	91.55	0.68→**0.34**		ARC	92.24	90.27	91.21	**90.61**	$1.97\rightarrow$**0.60**
	MMLU	78.44	71.01	72.20	71.73	$7.43^{\dagger}\rightarrow$**0.47**‡		MMLU	59.83	33.93	73.60	**72.45**	$25.90^{\dagger}\rightarrow$**1.15**†‡
	MMMU-Pro*	60.46	45.14	57.98	43.87	–		MMMU-Pro*	48.09	33.35	**55.26**†	34.68	–

Table 3: Self-distillation with Pure-Image inputs, Text-mode teachers, and LM-only LoRA. $|\Delta|$: absolute text–image gap (baseline→distilled). *MMMU-Pro is a forgetting control: T/I denote Standard/Vision, with no text–image gap. † marks significant baseline/residual gaps or MMMU-Pro changes from baseline; ‡ marks significant signed gap closure (Holm-adjusted $p < 0.05$; Appendix R).

6.1 Results

Self-distillation closes the modality gap. Table 3 presents results across two architectures (Qwen3-VL-8B and InternVL3-8B) and two training datasets (GSM8K and MMLU) with the highest modality gaps. With a small amount of self-distillation, all four configurations substantially close the gap. For both models and both training sets, self-distillation reduces the modality gap by an order of magnitude or more. The largest improvement appears for Qwen3-VL-8B + GSM8K, where image-mode accuracy jumps from 30.71% to 92.72%, matching text-mode performance. Text-mode performance is largely preserved, with one exception (Qwen3-VL-8B trained on MMLU drops from 78.44% to 72.20% on MMLU text), indicating that self-distillation strengthens the visual-text alignment while largely preserving the original textual reasoning capability.

Transfer to unseen benchmarks. We evaluate each trained model on benchmarks not used during training to test whether self-distillation causes catastrophic forgetting. Across all four configurations, models preserve or improve performance on held-out tasks, with the text-image gap narrowing on most benchmarks. For instance, InternVL3-8B trained on GSM8K improves MMLU image-mode from 33.93% to 68.00%; Qwen3-VL-8B trained on GSM8K maintains HumanEval image-mode performance (Section E). Our results above suggest that self-distillation teaches general visual-text alignment rather than task-specific shortcuts.

We additionally evaluate MMMU-Pro [48] using its official `Standard` and `Vision` configurations with unmodified input formats, greedy decoding, a maximum of 8,192 output tokens, strict matching of the extracted answer letter to the gold answer, and no additional instruction prompt. Due to the nature of the task, MMMU-Pro has no `Pure Text` mode and serves only as a held-out catastrophic-forgetting control; our gap-closure claim rests on GSM8K, MMLU, ARC, and HumanEval, where identical content is available in both text and image form. Across the four model–training-set configurations, MMMU-Pro `Vision` accuracy changes by at most 1.33 percentage points, with no statistically significant change, consistent with preserved performance on this control. In `Standard`, InternVL3-8B improves by 6.42 and 7.17 percentage points after GSM8K and MMLU self-distillation, respectively (both $p < 10^{-4}$).

Ablations: filtering and adaptation strategy. We then ablate two design choices on Qwen3-VL-8B + GSM8K (Table 4). First, we compare *filtered* traces (only correct answers) against *unfiltered* traces (all outputs regardless of correctness): the gap is only 1.29%, suggesting that, as long as

the model maintains a good level of textual reasoning capability, the reasoning *structure* matters more than perfect correctness, showing the potential to conduct self-distillation with unlabeled data.

Second, we compare three adaptation strategies: ViT+LM (Full LoRA), LM-only (freeze ViT), and ViT-only (freeze LM). ViT-only training raises image-mode accuracy to 85.29%, showing that better visual feature alignment alone yields substantial gains. LM-only training achieves 92.72%, matching ViT+LM (92.57%), indicating that adapting the language model is the key factor, reducing the effort from tuning the full model to tuning only the LM component.

Strategy	Baseline		Self-Distilled		$\lvert\Delta\rvert$
	Text (T)	Image (I)	Text (T)	Image (I)	
ViT+LM (unfiltered)	93.56	30.71	93.71	91.28	2.43
ViT+LM (filtered)			93.03	92.57	**0.46**
ViT-only			93.25	85.29	7.96
LM-only			**94.09**	**92.72**	1.37

Table 4: Ablation on Qwen3-VL-8B + GSM8K: filtered vs. unfiltered traces, and ViT+LM / ViT-only / LM-only adaptation. LM-only with filtered traces is the canonical recipe. $|\Delta|$: text–image gap.

7 Conclusion

We move beyond reporting the text-vs-image accuracy gap to diagnosing why it arises and how it can be closed. Across seven benchmarks, seven models, and five input modes, three findings reshape how the modality gap should be understood. First, MLLMs already perform strongly on natural document images and struggle primarily on synthetic renderings whose visual properties diverge from pretraining distributions, with font, resolution, and compression alone shifting accuracy by tens of percentage points. Second, an error taxonomy built from over 5,000 failures reveals that image mode does not impair perception or knowledge retrieval but triggers a reasoning collapse in which models produce shorter outputs and skip step-by-step reasoning. The reluctance to reason, not the inability to read, drives the gap. Third, this diagnosis suggests a lightweight remedy: self-distillation on the model's own text-mode reasoning traces, paired with image inputs, closes the gap without architectural changes, external data, or catastrophic forgetting. These findings translate into concrete guidance for two audiences: practitioners using existing MLLMs should add explicit chain-of-thought prompts to visual inputs and attend to rendering choices. Model builders, in turn, should treat long-form reasoning over visual inputs as a major training objective rather than relying on short VQA-style answers. Together, our results suggest that the reported modality gaps stem less from a fundamental limit of pixel-based language understanding than from a fixable misalignment between how MLLMs are trained to behave on visual inputs and how they are capable of reasoning over them.

Acknowledgments and Disclosure of Funding

Part of this study is supported by the Lambda Research Grant Program by lambda.ai. K.S. was supported in part by DARPA (SciFY). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government.

References

[1] Stanislas Dehaene and Laurent Cohen. Cultural recycling of cortical maps. *Neuron*, 56(2):384–398, 2007.

[2] Ghislaine Dehaene-Lambertz, Karla Monzalvo, and Stanislas Dehaene. The emergence of the visual word form: Longitudinal evolution of category-specific ventral visual areas during reading acquisition. *PLoS biology*, 16(3):e2004103, 2018.

[3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

[5] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning

for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[6] Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, et al. Pixtral 12b. *arXiv preprint arXiv:2410.07073*, 2024.

[7] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[8] Zhiheng Lyu, Xueguang Ma, and Wenhu Chen. Pixelworld: How far are we from perceiving everything as pixels?, 2025.

[9] Angela van Sprang, Laurens Samson, Ana Lucic, Erman Acar, Sennay Ghebreab, and Yuki M Asano. Same content, different answers: Cross-modal inconsistency in mllms. *arXiv preprint arXiv:2512.08923*, 2025.

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

[11] Tianyu Gao, Zirui Wang, Adithya Bhaskar, and Danqi Chen. Improving language understanding from screenshots, 2024.

[12] Ling Xing, Alex Jinpeng Wang, Rui Yan, Hongyu Qu, Zechao Li, and Jinhui Tang. See the text: From tokenization to visual reading. *arXiv preprint arXiv:2510.18840*, 2025.

[13] Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, et al. Glyph: Scaling context windows via visual-text compression. *arXiv preprint arXiv:2510.17800*, 2025.

[14] Xinlong Wang, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Zhen Li, Yuqi Wang, et al. Multimodal learning with next-token prediction for large multimodal models. *Nature*, pages 1–7, 2026.

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

[16] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusiñol, Ernest Valveny, C. V. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering, 2019.

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

[18] Qing'an Liu, Juntong Feng, Yuhao Wang, Xinzhe Han, Yujie Cheng, Yue Zhu, Haiwen Diao, Yunzhi Zhuge, and Huchuan Lu. Vista-bench: Do vision-language models really understand visualized text as well as pure text? *arXiv preprint arXiv:2602.04802*, 2026.

[19] Yanhong Li, Zixuan Lan, and Jiawei Zhou. Text or pixels? evaluating efficiency and understanding of LLMs with visual text inputs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10564–10578, Suzhou, China, November 2025. Association for Computational Linguistics.

[20] Zhaochen Wang, Yiwei Wang, and Yujun Cai. Cure or poison? embedding instructions visually alters hallucination in vision-language models. *arXiv preprint arXiv:2508.01678*, 2025.

[21] Keya Hu, Ali Cy, Linlu Qiu, Xiaoman Delores Ding, Runqian Wang, Yeyin Eva Zhu, Jacob Andreas, and Kaiming He. Arc is a vision problem! *arXiv preprint arXiv:2511.14761*, 2025.

[22] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer, 2015.

[23] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation, 2022.

[24] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[25] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[26] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

[27] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. GPQA: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[28] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[29] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

[30] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. 2021.

[31] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[32] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025.

[33] Thinh Pham, Nguyen Nguyen, Pratibha Zunjare, Weiyuan Chen, Yu-Min Tseng, and Tu Vu. Sealqa: Raising the bar for reasoning in search-augmented language models, 2025.

[34] Benjamin Clavié and Florian Brand. Readbench: Measuring the dense text visual reading ability of vision-language models, 2025.

[35] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[36] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[37] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[38] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[39] 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, and Sophia Yang. Pixtral 12b, 2024.

[40] Juliet Corbin and Anselm Strauss. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage publications, 2014.

[41] Barney G Glaser, Anselm L Strauss, and Elizabeth Strutzel. The discovery of grounded theory; strategies for qualitative research. *Nursing research*, 17(4):364, 1968.

[42] Laura K Nelson. Computational grounded theory: A methodological framework. *Sociological methods & research*, 49(1):3–42, 2020.

[43] Mian Zhong, Pristina Wang, and Anjalie Field. Hicode: Hierarchical inductive coding with llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31048–31066, 2025.

[44] Ruyuan Wan, Haonan Wang, Ting-Hao Huang, and Jie Gao. From noise to nuance: Enriching subjective data annotation through qualitative analysis. In *Proceedings of the Fourth Workshop on Bridging Human-Computer Interaction and Natural Language Processing (HCI+ NLP)*, pages 240–254, 2025.

[45] Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent llm systems fail?, 2025.

[46] Ziang Xiao, Xingdi Yuan, Q Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. Supporting qualitative analysis with large language models: Combining codebook with gpt-3 for deductive coding. In *Companion proceedings of the 28th international conference on intelligent user interfaces*, pages 75–78, 2023.

[47] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[48] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186, 2025.

A Realistic Data Preparation

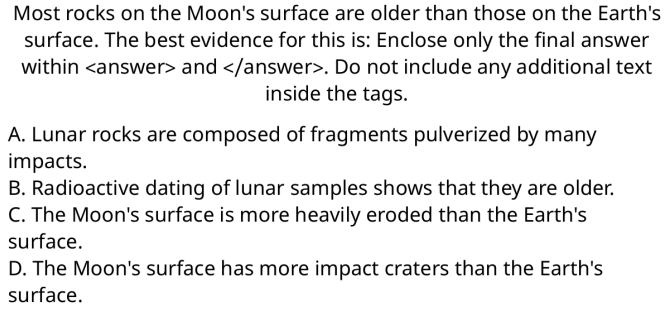
QASPER [30] is a long-form QA dataset over scientific articles that often requires multi-sentence reasoning. All instances that require access to tables and figures are removed to ensure a fair comparison between the `Pure Text` format and `Pure Image` format. We download the original PDF files from arXiv and convert all pages into PNG images using the *fitz* library, excluding the

reference section, which is also not included in the `Pure Text` format of QASPER. For each example in **SQuAD** v2, we capture a screenshot of the corresponding Wikipedia page to create a natural-image version of the context. The corresponding screenshot is manually aligned with the passages given in the original SQuAD v2 data. Due to the nature of webpages, there would be unavoidable truncations of images or tables in the `Pure Image` input that is not included by the `Pure Text` setting. For evaluation, zero-shot inference on QA tasks does not align well with the original span-based metrics, because MLLMs are not trained to reproduce annotation artifacts such as exact spans. Instead, following recent information-seeking benchmarks [32, 33], we adopt an LLM-as-judge protocol, where use GPT-5 (*gpt-5-2025-08-07*) to compare the model prediction with the gold answer and report accuracy over all examples. Results on span-based metrics can be found in Section I.

B Detailed Input Settings

This section provides the full input formats used across all configurations, along with examples for each dataset. Unless otherwise specified, models are prompted to produce outputs wrapped in `<answer>...</answer>`. For datasets that require structured or non–free-form outputs, we apply a permissive parsing strategy to extract the final predictions. All evaluations are conducted in a zero-shot setting. Accuracy is used as the primary metric for all benchmarks, with the exception of HumanEval, which is evaluated using pass@k, where $k = 1$.

MMLU MMLU evaluates a model's knowledge and reasoning ability across 57 subjects spanning the humanities, social sciences, STEM fields, and professional domains. Each question is multiple-choice and often requires domain-specific expertise. In our setting, we adopt its original tasks and targets. The example input for MMLU is shown in Figure S1.



(a) `Image` format.

`Text`	`Answer the following multiple choice question. Respond with just the letter of the correct answer (A, B, C, or D).`
	`Most rocks on the Moon's surface are older than those on the Earth's surface. The best evidence for this is:`
	`A. Lunar rocks are composed of fragments pulverized by many impacts.`
	`B. Radioactive dating of lunar samples shows that they are older.`
	`C. The Moon's surface is more heavily eroded than the Earth's surface.`
	`D. The Moon's surface has more impact craters than the Earth's surface.`
	`Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags.`

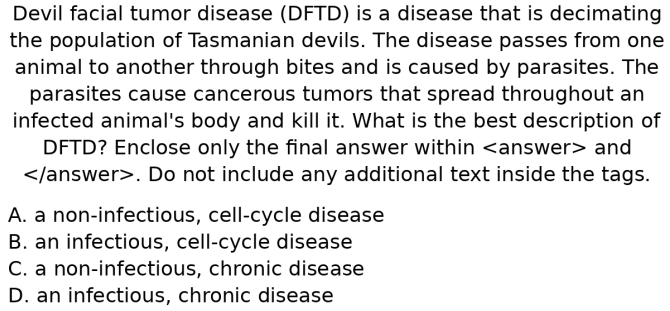
`Instr. + Image`	`Please follow the instruction in the image. Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags. [Image provided in (a) without the format instructions.]`

(b) `Text` and `Instr. + Image` format.

Figure S1: An example input of MMLU.

ARC ARC-Challenge (ARC) assesses a model's ability to perform complex, non-trivial reasoning on grade-school science questions. Unlike its easier counterpart (ARC-Easy), ARC is intentionally

adversarial: questions cannot be solved by surface heuristics and typically require multi-step inference, commonsense knowledge, or integration of scientific facts. We adopt its original tasks and targets, and the example input for ARC-C is shown in Figure S2.



(a) `Image` format.

`Text`	`Answer the following multiple choice question. Respond with just the letter of the correct answer (A, B, C, or D).` `Devil facial tumor disease (DFTD) is a disease that is decimating the population of Tasmanian devils. The disease passes from one animal to another through bites and is caused by parasites. The parasites cause cancerous tumors that spread throughout an infected animal's body and kill it. What is the best description of DFTD?` `A. a non-infectious, cell-cycle disease", "an infectious, cell-cycle disease` `B. an infectious, cell-cycle disease` `C. a non-infectious, chronic disease` `D. an infectious, chronic disease` `Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags.`
`Instr. + Image`	`Please follow the instruction in the image. Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags. [Image provided in (a) without the format instructions.]`

(b) `Text` and `Instr. + Image` format.

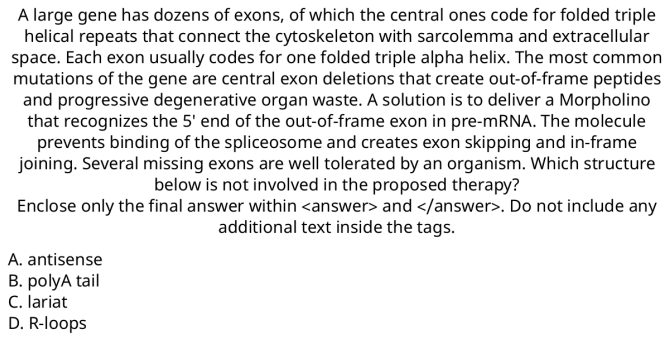
Figure S2: An example input of ARC.

GPQA GPQA [27] is a graduate-level question answering benchmark consisting of 448 challenging multiple-choice questions in physics, chemistry, and biology. The questions are crafted by domain experts and are designed to be difficult even for specialists outside their own subfield, requiring deep domain knowledge and multi-step scientific reasoning. We adopt its original tasks and targets, and the example input for GPQA is shown in Figure S3.

HumanEval HumanEval [29] measures a model's ability to synthesize correct and executable code from natural language descriptions. Each task provides a short docstring specifying a function's intended behavior, and models must generate a complete Python implementation that passes hidden unit tests. The example input of HumanEval is shown in Figure S4.

GSM8K GSM8K evaluates a model's ability to solve grade-school mathematical word problems that require multi-step reasoning. Each question is free-form and demands 2 to 8 steps to arrive at the final answer. The solution of GSM8K is provided in natural language format, and a final answer is given after the reasoning steps. In our setting, we adopt its original data and consider primarily the correctness of the answer (extracting the answer from the reasoning chain). The example input for GSM8K is shown in Figure S5. To avoid confounding factors introduced by in-context examples, all the experiments are run on the zero-shot condition.

SQuAD v2. SQuAD v2 is an extractive question answering dataset that tests a model's ability to extract the answers from a given passage and to recognize unanswerable queries. Each instance



(a) `Image` format.

`Text`	`A large gene has dozens of exons, of which the central ones code for folded triple helical repeats that connect the cytoskeleton with sarcolemma and extracellular space. Each exon usually codes for one folded triple alpha helix. The most common mutations of the gene are central exon deletions that create out-of-frame peptides and progressive degenerative organ waste. A solution is to deliver a Morpholino that recognizes the 5' end of the out-of-frame exon in pre-mRNA. The molecule prevents binding of the spliceosome and creates exon skipping and in-frame joining. Several missing exons are well tolerated by an organism. Which structure below is not involved in the proposed therapy? Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags. A. antisense B. polyA tail C. lariat D. R-loops`
`Instr. + Image`	`Please follow the instruction in the image. Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags. [Image provided in (a) without the format instructions.]`

(b) `Text` and `Instr. + Image` format.

Figure S3: An example input of GPQA.

presents a question and a supporting passage from Wikipedia, and models must either extract a precise span or correctly abstain when the passage lacks sufficient information.

QASPER. QASPER is a long-context reading comprehension task that evaluates a model's ability to answer questions about academic research papers. Each question targets specific information regarding methods, findings, assumptions, or limitations of the paper, requiring models to reason over evidence within full-length scientific documents. We feed the entire paper on a given question as the input. The example input for QASPER is shown in Figure S7.

B.1 OCR Prompt

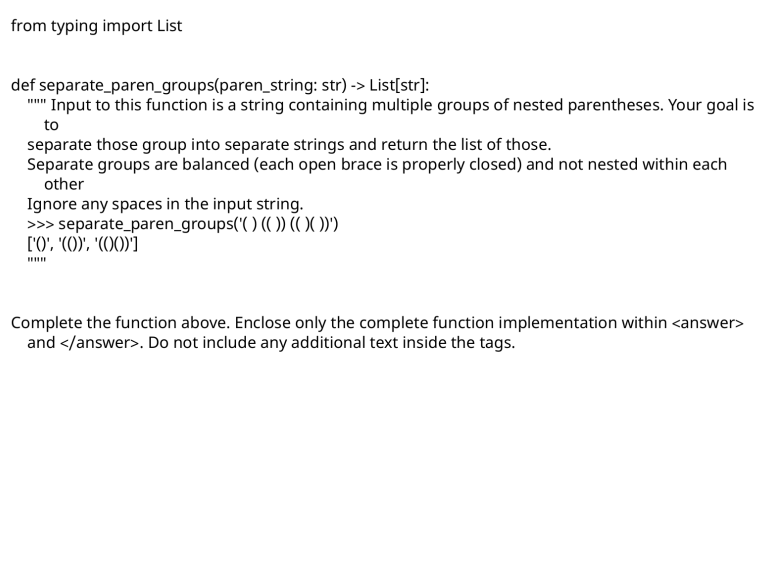
The prompt used in OCR-P2 is shown in Figure S8. It is also used in OCR-P1 as the guidelines for performing extractions.

C Dataset Statistics

Table S1 summarizes the composition of our evaluation suite. The five synthetic-image benchmarks (MMLU, ARC, GPQA, GSM8K, and HumanEval) comprise 17,145 instances in total, with MMLU accounting for the largest share (81.9%). QASPER and SQuAD v2 use natural images and are listed separately.

D Experiment Setting Details

We use a unified decoding setup across models and datasets. Unless otherwise noted, the temperature is set to 0.1 with a batch size of 8 and a maximum generation length of 1024 tokens. For code generation tasks, the temperature is set to 0.2 to ensure diversity on the generation results.



(a) `Image` format.

`Text`	`from typing import List def separate_paren_groups(paren_string: str) -> List[str]: """ Input to this function is a string containing multiple groups of nested parentheses. Your goal is to separate those group into separate strings and return the list of those. Separate groups are balanced (each open brace is properly closed) and not nested within each other Ignore any spaces in the input string. >> separate_paren_groups('( ) (( )) (( )( ))') ['()', '(())', '(()())'] """`  `Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags.`
`Instr. + Image`	`Please follow the instruction in the image. Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags. [Image provided in (a) without the format instructions.]`

(b) `Text` and `Instr. + Image` format.

Figure S4: An example input of HumanEval.

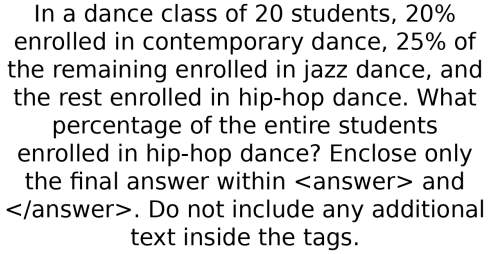
Dataset	Domain	Metric	#Samples	Proportion (%)
MMLU	Knowledge (57 subjects)	Acc.	14,042	81.9
ARC	Science reasoning	Acc.	1,172	6.8
GPQA	Graduate-level QA	Acc.	448	2.6
GSM8K	Math reasoning	Acc.	1,319	7.7
HumanEval	Code generation	Pass@1	164	1.0
QASPER	Document QA	F1 / Acc.	1,445	—
SQuAD v2	Reading comprehension	F1 / Acc.	100	—
Total (synthetic-image benchmarks)			**17,145**	**100.0**

Table S1: Dataset statistics and composition.

For OCR-1P, the maximum generation length is set to 4096, as the model is asked to perform OCR and solve the task in one inference. The generation length is increased to allow full inclusion of OCR results and task reasoning in one output.

E HumanEval Results

We report HumanEval results separately due to the high variance inherent in this benchmark. With only 164 instances, pass@1 estimates are statistically unstable; Chen et al. [29] recommend generating



(a) `Image` format.

`Text`	`In a dance class of 20 students, 20% enrolled in contemporary dance, 25% of the remaining enrolled in jazz dance, and the rest enrolled in hip-hop dance. What percentage of the entire students enrolled in hip-hop dance? Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags.`
`Instr. + Image`	`Please follow the instruction in the image. Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags. [Image provided in (a) without the format instructions.]`

(b) `Text` and `Instr. + Image` format.

Figure S5: An example input of GSM8K.

many samples per problem and using an unbiased estimator for reliable evaluation. Additionally, code generation is particularly sensitive to formatting failures (Section 5.1).

Model	Pure Text (T)	Instr.+ Image (T I)	Pure Image (I)	OCR-1P (I → T)	OCR-2P (I → T)
GPT-5.2	100.00	71.34†	100.00	98.17	100.00
InternVL3-8B	56.71	71.95	45.12	71.34	78.05†
InternVL3.5-8B	90.24	90.24	89.02	84.76	86.59
Pixtral-12B	39.02	79.27†	47.56	52.44	0.00†
Qwen2.5-7B-VL	59.15	83.54†	65.24	52.44	64.02
Qwen2.5-32B-VL	69.51	22.56	85.98	84.76	82.32
Qwen3-VL-8B	76.22	82.93	82.93	85.37	87.20

Table S2: HumanEval (pass@1) results across input modalities. Cell shading indicates performance difference relative to `Pure Text`: red = outperforms text, blue = underperforms text. Results exhibit high variance due to the small dataset size (164 instances) and the inherent instability of pass@1 estimation from limited samples [29]. †: significant difference from Pure Text (Holm-adjusted $p < 0.05$; Appendix R).

F Rendering Significance Tests

For the rendering comparisons in Figure 3, we pair each alternative rendering with the default on the same benchmark items. We use two-sided exact McNemar tests for binary correctness outcomes and paired bootstrap tests for HumanEval, where the ten generated samples per problem in each alternative rendering are averaged before pairing with the single-sample default. We use 10,000 paired bootstrap resamples for the confidence intervals and bootstrap tests, and apply Holm correction jointly across the 60 condition–default comparisons.

G Resolution Sensitivity

Figure S9 presents the full resolution sensitivity analysis on both HumanEval and ARC. Most models maintain stable performance until resolution reaches a lower bound, while InternVL3.5-8B maintains stable performance across all resolutions due to its Visual Resolution Router [38].



(a) Example `Image` input from a Wikipedia screenshot, preserving the webpage's typography, layout, and visual elements. The captured page may differ from the original SQuAD passage.

`Text`	`When was King George's war? The conflict is known by multiple names. In British America, wars were often named after the sitting British monarch, such as King William's War or Queen Anne's War. As there had already been a King George's War in the 1740s, British colonists named the second war in King George's reign after their opponents, and it became known as the French and Indian War. This traditional name continues as the standard in the United States, but it obscures the fact that Indians fought on both sides of the conflict, and that this was part of the Seven Years' War, a much larger conflict between France and Great Britain. American historians generally use the traditional name or sometimes the Seven Years' War. Other, less frequently used names for the war include the Fourth Intercolonial War and the Great War for the Empire. Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags.`
`Instr. + Image`	`Please follow the instructions in the image. Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags. [Image provided in (a) without the format instructions.]`

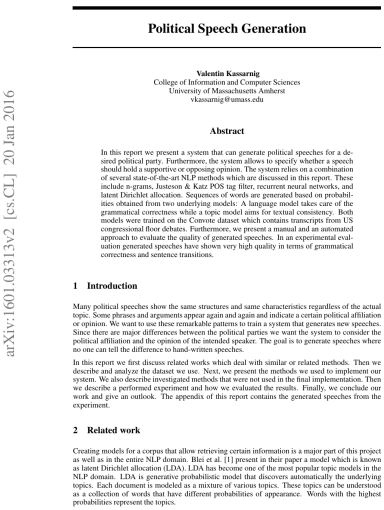
(b) `Text` and `Instr. + Image` format.

Figure S6: An example input of SQUAD v2.

H Full Results Table

We report the complete performance of all models under all input configurations in Table S3. The table covers *PureText*, *PureImage*, *Inst.+Image* (Real vs. Rendered), and both OCR-based variants (OCR-1P and OCR-2P), evaluated across seven benchmarks: MMLU, ARC, GPQA, GSM8K, QASPER, SQuAD v2, and HumanEval. All results are reported as Accuracy or F1, except for **HumanEval**, which follows the standard **pass@1** metric. Accuracy values of QASPER and SQuAD v2 are computed using the model-based evaluator described in Section I, ensuring consistent comparison across heterogeneous answer formats.

Question:
how did they measure grammatical correctness?



arXiv:1601.03313v2 [cs.CL] 20 Jan 2016

Political Speech Generation

Valentin Kassarnig
College of Information and Computer Sciences
University of Massachusetts Amherst
vkassarnig@umass.edu

Abstract

In this report we present a system that can generate political speeches for a desired political party. Furthermore, the system allows to specify whether a speech should hold a supportive or opposing opinion. The system relies on a combination of several state-of-the-art NLP methods which are discussed in this report. These include n-grams, Justeson & Katz POS tag filter, recurrent neural networks, and latent Dirichlet allocation. Sequences of words are generated based on probabilities obtained from two underlying models: A language model takes care of the grammatical correctness while a topic model aims for textual consistency. Both models were trained on the Convote dataset which contains transcripts from US congressional floor debates. Furthermore, we present a manual and an automated approach to evaluate the quality of generated speeches. In an experimental evaluation generated speeches have shown very high quality in terms of grammatical correctness and sentence transitions.

1 Introduction

Many political speeches show the same structures and same characteristics regardless of the actual topic. Some phrases and arguments appear again and again and indicate a certain political affiliation or opinion. We want to use these remarkable patterns to train a system that generates new speeches. Since there are major differences between the political parties we want the system to consider the political affiliation and the opinion of the intended speaker. The goal is to generate speeches where no one can tell the difference to hand-written speeches.

In this report we first discuss related works which deal with similar or related methods. Then we describe and analyze the dataset we use. Next, we present the methods we used to implement our system. We also describe investigated methods that were not used in the final implementation. Then we describe a performed experiment and how we evaluated the results. Finally, we conclude our work and give an outlook. The appendix of this report contains the generated speeches from the experiment.

2 Related work

Creating models for a corpus that allow retrieving certain information is a major part of this project as well as in the entire NLP domain. Blei et al. [1] present in their paper a model which is known as latent Dirichlet allocation (LDA). LDA has become one of the most popular topic models in the NLP domain. LDA is generative probabilistic model that discovers automatically the underlying topics. Each document is modeled as a mixture of various topics. These topics can be understood as a collection of words that have different probabilities of appearance. Words with the highest probabilities represent the topics.

1

(a) The first image of `Image` format. The remaining pages are omitted for length considerations. All pages of the paper are fed into the model.

`Text`	`Question: how did they measure grammatical correctness?` `Political Speech Generation` `In this report we present a system that can generate political speeches for a desired political party. Furthermore, the system allows to specify whether a speech should hold a supportive or opposing opinion. The system relies on a combination of several state-of-the-art NLP methods which are discussed in this report. These include n-grams, ...[Full Paper Text]...` `Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags.`
`Instr. + Image`	`Please follow the instructions in the image. Enclose only the final answer within <answer> and </answer>. Do not include any additional text inside the tags. [Image provided in (a) without the format instructions.]`

(b) `Text` and `Instr. + Image` format.

Figure S7: An example input of QASPER.

OCR Extraction Prompt.

You are an Optical Character Recognition (OCR) model. Your task is to extract all readable text from the provided image.
Instructions:
1. Identify and transcribe all visible text exactly as it appears, including numbers, punctuation, and symbols.
2. Preserve line breaks and spacing whenever possible.
3. For structured content (tables, forms, equations, lists), format using simple text notation:
- Tables: use tabs or | separators between columns
- Lists: use bullet points (-) or numbers
- Equations: reproduce mathematical symbols faithfully
4. If a word or section is unclear, write [illegible] in its place.
5. Do not interpret, summarize, or translate the text—transcribe it exactly as shown.

Figure S8: Prompt used for OCR-style transcription from images.

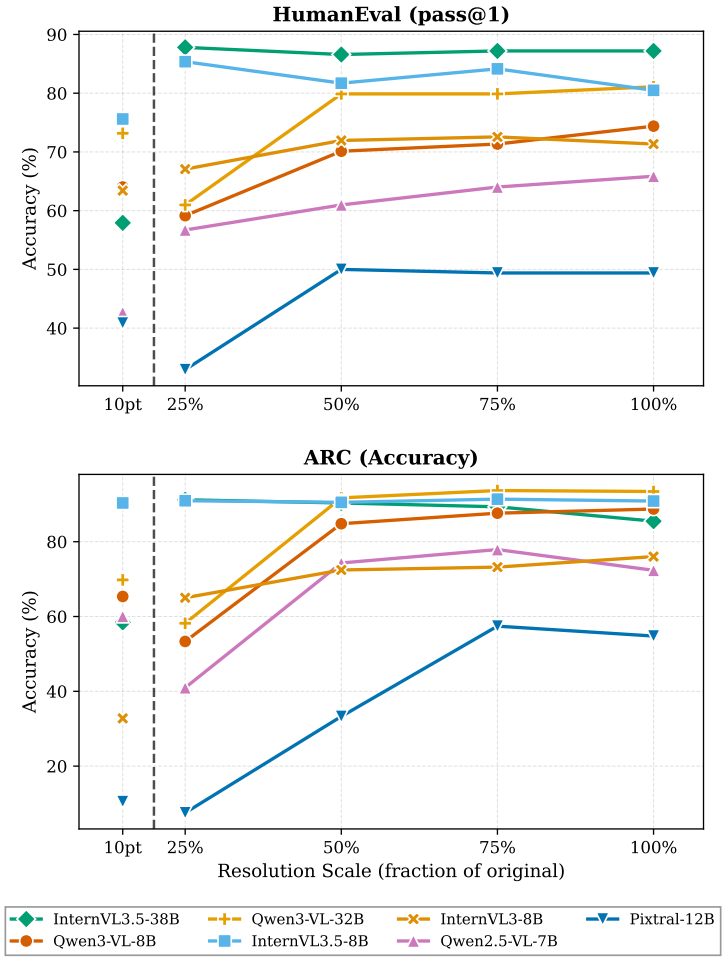


Figure S9: Performance vs. image resolution scale on HumanEval (top) and ARC (bottom). The black vertical dashed line indicates the point where `Pure Image` consumes the same FLOPs as `Pure Text`.

Table S3: Performance across input modalities on all seven datasets. Values are percentages; definitions and provenance are discussed in Appendix H. For the five diagnostic benchmarks, † marks a significant difference from Pure Text (Holm-adjusted $p < 0.05$; Appendix R).

Model	Text	Natural Instr.+Img.	Natural Pure Img.	OCR-1P	OCR-2P	Rendered Pure Img.	Rendered Instr.+Img.
MMLU (accuracy)							
GPT-5.2	92.33	–	–	86.54†	86.72†	90.93†	91.06†
InternVL3-8B	59.83	–	–	60.40	71.86†	33.93†	69.09†
InternVL3.5-8B	82.97	–	–	77.65†	80.75†	80.74†	82.25
Pixtral-12B	37.89	–	–	49.54†	44.25†	42.76†	41.60†
Qwen2.5-7B-VL	67.80	–	–	65.46†	67.08	67.36	67.97
Qwen2.5-32B-VL	78.34	–	–	76.55†	80.38†	73.79†	75.47†
Qwen3-VL-8B	78.44	–	–	72.98†	73.61†	71.01†	72.29†
ARC (accuracy)							
GPT-5.2	96.25	–	–	97.70	97.18	95.48	92.24†
InternVL3-8B	92.24	–	–	86.26†	90.53	90.27	91.30
InternVL3.5-8B	92.66	–	–	93.26	93.52	92.66	92.41
Pixtral-12B	84.64	–	–	72.10†	39.33†	61.77†	71.25†
Qwen2.5-7B-VL	87.20	–	–	87.71	86.35	89.42	85.92
Qwen2.5-32B-VL	93.09	–	–	94.62	94.37	90.02†	91.30
Qwen3-VL-8B	91.72	–	–	92.92	92.75	91.04	91.38
GPQA (accuracy)							
GPT-5.2	81.25	–	–	77.90	80.58	80.13	81.25
InternVL3-8B	33.71	–	–	30.58	33.26	32.81	32.14
InternVL3.5-8B	42.63	–	–	40.18	46.21	34.60†	40.40
Pixtral-12B	31.25	–	–	29.69	30.36	27.23	30.36
Qwen2.5-7B-VL	34.15	–	–	29.02	32.59	30.58	34.15
Qwen2.5-32B-VL	42.41	–	–	43.08	43.30	43.97	41.52
Qwen3-VL-8B	43.75	–	–	37.28	36.38	35.04†	39.29
GSM8K (accuracy)							
GPT-5.2	95.83	–	–	77.86†	68.16†	96.51	85.14†
InternVL3-8B	78.01	–	–	73.92	87.87†	42.53†	49.51†
InternVL3.5-8B	95.30	–	–	95.60	94.47	95.30	95.22
Pixtral-12B	3.79	–	–	75.89†	55.12†	8.11†	4.93
Qwen2.5-7B-VL	19.48	–	–	20.09	20.17	24.79†	20.24
Qwen2.5-32B-VL	95.07	–	–	66.41†	96.13	94.92	4.47†
Qwen3-VL-8B	93.56	–	–	56.94†	94.77	30.71†	39.58†
HumanEval (pass@1)							
GPT-5.2	100.00	–	–	98.17	100.00	100.00	71.34†
InternVL3-8B	56.71	–	–	71.34	78.05†	45.12	71.95
InternVL3.5-8B	90.24	–	–	84.76	86.59	89.02	90.24
Pixtral-12B	39.02	–	–	52.44	0.00†	47.56	79.27†
Qwen2.5-7B-VL	59.15	–	–	52.44	64.02	65.24	83.54†
Qwen2.5-32B-VL	69.51	–	–	84.76	82.32	85.98	22.56
Qwen3-VL-8B	76.22	–	–	85.37	87.20	82.93	82.93
QASPER (full archive), F1							
GPT-5.2	29.18	29.36	28.28	28.89	–	27.25	29.76
InternVL3-8B	21.60	16.32	12.51	15.51	–	10.84	16.27
Pixtral-12B	–	–	–	–	–	–	–
Qwen2.5-32B-VL	11.41	11.46	12.88	12.08	–	13.40	11.46
Qwen2.5-7B-VL	10.07	24.39	10.28	19.43	–	19.26	24.31
QASPER (full archive), Acc.							
GPT-5.2	55.44	76.99	75.19	75.92	–	71.64	74.36
InternVL3-8B	48.14	69.38	55.26	56.40	–	46.28	62.97
Pixtral-12B	–	–	–	–	–	–	–
Qwen2.5-32B-VL	43.18	77.37	75.12	76.02	–	72.85	74.36
Qwen2.5-7B-VL	35.56	68.48	65.78	58.20	–	64.68	65.64
QASPER (Table 2 subset), F1							
GPT-5.2	25.56	29.64	29.61	29.19	–	27.86	30.15
InternVL3-8B	17.78	16.91	13.07	15.54	–	11.20	16.71

Continued on next page

Table S3 (continued)

Model	Text	Natural Instr.+Img.	Natural Pure Img.	OCR-1P	OCR-2P	Rendered Pure Img.	Rendered Instr.+Img.
Pixtral-12B	–	–	–	–	–	–	–
Qwen2.5-32B-VL	12.06	12.13	10.70	12.92	–	10.44	11.55
Qwen2.5-7B-VL	10.60	25.44	19.45	20.73	–	19.14	24.26
QASPER (Table 2 subset), Acc.							
GPT-5.2	51.92	75.10	77.25	74.27	–	70.88	70.47
InternVL3-8B	44.02	69.36	54.43	56.15	–	44.64	62.57
Pixtral-12B	–	–	–	–	–	–	–
Qwen2.5-32B-VL	37.71	76.35	76.14	76.56	–	71.36	70.05
Qwen2.5-7B-VL	30.49	66.32	64.38	57.05	–	62.09	–
Qwen3-VL-8B	53.27	76.25	70.75	56.89	–	–	–
InternVL3.5-8B	49.38	57.66	61.07	55.68	–	–	–
SQuAD v2 (100 matched questions), F1							
GPT-5.2	78.92	68.77	72.44	68.15	65.38	75.77	81.51
InternVL3-8B	32.59	32.17	24.98	18.63	26.12	14.83	38.04
InternVL3.5-8B	48.78	26.89	13.91	28.51	6.10	17.13	32.52
Pixtral-12B	74.17	32.72	19.24	33.07	31.58	7.06	39.02
Qwen2.5-7B-VL	6.63	25.60	27.03	30.32	29.30	29.10	35.41
Qwen2.5-32B-VL	13.85	25.95	20.40	26.09	24.27	28.34	29.65
Qwen3-VL-8B	66.46	36.47	27.56	33.12	20.38	32.42	45.26
SQuAD v2 (100 matched questions), accuracy							
GPT-5.2	97.50	85.50	94.00	85.50	85.50	95.50	98.00
InternVL3-8B	82.50	84.00	73.50	69.50	76.50	79.50	94.00
InternVL3.5-8B	98.00	83.50	84.00	79.50	86.00	91.00	96.00
Pixtral-12B	91.50	81.00	83.50	84.50	82.50	81.00	93.00
Qwen2.5-7B-VL	80.00	82.50	83.50	81.50	77.00	90.00	93.50
Qwen2.5-32B-VL	86.50	84.00	87.00	83.00	83.50	95.50	95.00
Qwen3-VL-8B	95.00	85.50	84.50	84.00	85.50	95.50	97.50

I Evaluation for QASPER and SQuAD v2.

Standard EM or span-matching metrics are insufficient as a primary measure in our setting, since many answers are free-form, paraphrastic, or require multi-sentence justification. We therefore adopt a model-based evaluation for Accuracy: GPT-5 serves as an automatic judge and assigns a score of 1.0 (fully correct), 0.5 (partially correct), or 0.0 (incorrect) by comparing the model prediction against the reference answer. Accuracy is reported as the mean of these scores across all examples. In addition, we report the F1 used in prior work on QASPER and SQuAD v2. Concretely, answers are normalized (lowercasing, removing punctuation and articles), and F1 is computed as the harmonic mean of precision and recall over overlapping tokens between the predicted and ground-truth answers, taking the maximum over multiple references when available.

J Compute Resources

We report the computational resources used for the main experiments in this paper.

Evaluation experiments. All evaluation experiments were conducted on NVIDIA A100/H100 80GB GPUs. Running inference across all 7 models, 7 benchmarks, and 5 input modes required approximately 1000 GPU-hours in total. The majority of compute was consumed by long-context datasets (QASPER and SQuAD), larger models (Qwen2.5-VL-32B and GPT-5.2 via API), and multi-staged inference. Approximtely 200 GPU-hours were used for the preliminary experiments.

Self-distillation training. Each self-distillation run (LoRA fine-tuning with rank 64) was conducted on a single H100 80GB GPUs. Training for 2 epochs on GSM8K ($\sim$7K examples) required approximately 2 hours per model, and training on MMLU ($\sim$7K examples) required approximately 3 hours per model.

Error analysis. The grounded-theory error analysis involved GPT-5.2 API calls for coding 5,245 errors. Including preliminary experiments, this consumed approximately $500 in API costs.

K FLOPs Analysis

To quantify the computational overhead of image-based inputs relative to text, we report the ratio of prefill FLOPs consumed by visual tokens to those consumed by text tokens for the same input. A higher ratio indicates that the image pathway demands proportionally more compute. As shown in Table S4, the ratio varies considerably across models and datasets. For short-context benchmarks (ARC, GSM8K, HumanEval, MMLU) with a fixed rendered-image resolution, the ratio typically ranges from $1.4\times$ to $4.7\times$, with Pixtral-12B being a notable outlier due to its higher native image-token count. For long-context datasets (SQuAD and QASPER), where the number of input images scales with document length, the ratio increases substantially, reaching up to $29\times$ for Pixtral-12B on QASPER. These results highlight that, while visual inputs can provide richer structural signals, they come at a non-trivial computational cost that grows with document length.

Table S4: Image-to-Text FLOPs ratio across models and datasets. A ratio of k indicates that processing the visual input requires $k\times$ the FLOPs of processing the equivalent text input. Missing entries indicate that the model does not support the corresponding configuration.

Dataset	InternVL3-8B	Pixtral-12B	Qwen2.5-7B-VL	Qwen2.5-32B-VL	Qwen3-VL-8B	Gemma3-4B
ARC	3.68	13.42	3.92	2.45	3.81	4.68
GSM8K	1.98	13.65	4.01	1.56	2.82	2.23
HumanEval	1.52	6.21	1.80	1.36	1.62	1.72
MMLU	3.45	12.78	–	2.28	2.74	4.11
SQuAD	8.64	20.70	12.37	7.54	8.98	8.98
QASPER	8.33	29.29	9.38	5.91	6.50	1.80

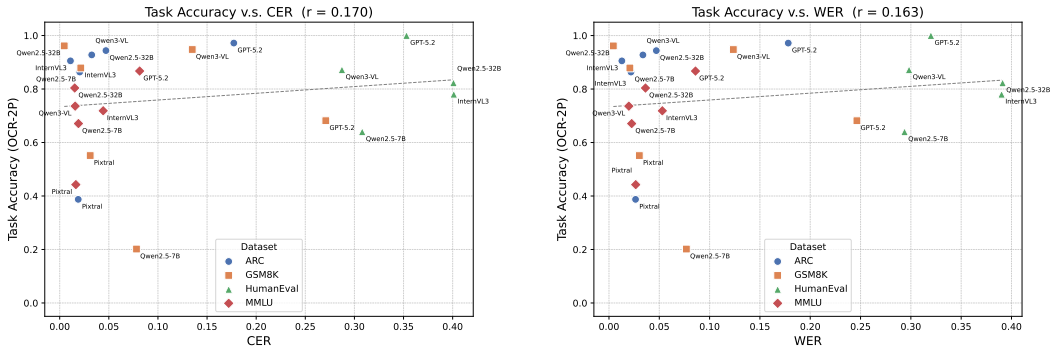
L OCR Quality and Task Accuracy

To investigate whether the visual recognition capability of each model explains the modality gap, we measure the correlation between OCR quality and task accuracy under the `OCR-2Pass` setting. For each model–dataset pair, we compute the Character Error Rate (CER) and Word Error Rate (WER) by comparing the OCR-extracted text against the gold text input, after removing the answer-format instruction sentence and normalizing whitespace. Figures S10a and S10b plot task accuracy against CER and WER, respectively.

The weak correlations ($r = 0.170$ for CER, $r = 0.163$ for WER) indicate that OCR quality alone does not account for the accuracy differences across models and datasets. The relationship is far from deterministic: several models achieve high accuracy despite moderate OCR errors, and vice versa. This suggests that the modality gap is driven by factors beyond text recognition, including reasoning collapse and sensitivity to visual formatting.

L.1 Handwriting Transcription Check

We conduct a separate transcription-only pass on Priestacy handwriting images for Qwen3-VL-8B and InternVL3-8B on MMLU, GSM8K, and ARC, sampling 50 items per model–dataset pair (300 outputs in total). We score the transcriptions against the exact text used to render each image and match each item to its handwriting image-mode answer to assess the relationship between transcription quality and task correctness. Fifteen outputs follow the answer-format instruction embedded in the image instead of the transcription request; these non-transcriptions are excluded from the reported WER/CER summaries but retained in the correlation analysis.



(a) Task accuracy vs. CER.

(b) Task accuracy vs. WER.

Figure S10: Correlation between OCR quality and task accuracy under the `OCR-2Pass` setting. Each point represents one model–dataset pair. The dashed line shows the linear trend.

Model	Dataset	WER (%)	CER (%)	Image acc. (%)	r	p
Qwen3-VL-8B	MMLU	2.7	0.6	76.0	$+0.003$	0.98
	GSM8K	1.8	1.4	48.0	-0.015	0.92
	ARC	0.9	1.5	92.0	-0.054	0.71
InternVL3-8B	MMLU	6.1	1.7	42.0	-0.215	0.13
	GSM8K	1.9	0.5	52.0	-0.144	0.32
	ARC	2.2	0.8	68.0	-0.111	0.44

Table S5: Handwriting transcription quality and task accuracy on 50 sampled items per model–dataset pair. WER and CER measure the separate transcription-only pass; image accuracy measures answers from the handwriting image-mode evaluation on the matched items. r is the point-biserial correlation between per-item WER and answer correctness, with unadjusted p-values. None of the six correlations is significant at $p < 0.05$.

As shown in Table S5, WER ranges from 0.9% to 6.1% and CER from 0.5% to 1.7%, while image-mode accuracy ranges from 42.0% to 92.0% on the matched items. None of the six point-biserial correlations between per-item WER and answer correctness reaches significance (all unadjusted $p \geq 0.13$). These results argue against widespread transcription failure as the sole explanation for the handwriting degradation, although this small-sample check cannot exclude effects from errors on individual task-critical tokens.

M Inference-Time CoT Prompting in Images

As an inference-time intervention for the reluctance to reason under image input, we append "Please think step-by-step." after the question and answer-format instruction in the rendered image. We evaluate this diagnostic on a subset of our evaluation suite: six open-weight models on all 1,319 GSM8K test questions, without additional training.

Since our primary goal is to align image-mode behavior with the model's text-mode reasoning through self-distillation, we report this prompting experiment only in the appendix.

Results. Table S6 shows higher accuracy for the CoT-in-image runs than the original `Pure Image` baselines for all six models. Qwen3-VL-8B and InternVL3-8B still fall below their text-mode baselines, whereas Qwen2.5-32B-VL and InternVL3.5-8B, whose image-mode baselines are already strong, change by less than 0.3 points. These results support explicit prompting as a useful inference-time intervention, while motivating training-time alignment that does not depend on an added reasoning cue.

Model	Pure Text	Pure Image	Image + CoT	Gain (pp)
InternVL3-8B	78.01	42.53	70.58	+28.05
InternVL3.5-8B	95.30	95.30	95.53	+0.23
Pixtral-12B	3.79	8.11	73.24	+65.13
Qwen2.5-7B-VL	19.48	24.79	40.41	+15.62
Qwen2.5-32B-VL	95.07	94.92	95.07	+0.15
Qwen3-VL-8B	93.56	30.71	86.96	+56.25

Table S6: GSM8K accuracy (%) with an explicit CoT instruction rendered inside the image, without additional training ($n = 1{,}319$ per model). Pure Text and Pure Image are the original baselines from Table 1; gain is Image + CoT minus Pure Image in percentage points. These are descriptive comparisons; decoding and rendering differences are detailed in Appendix M.

N Reasoning Collapse Example

Figure S11 illustrates the reasoning collapse phenomenon identified in Section 5.1. Given the same GSM8K problem, the model produces a detailed step-by-step solution in text mode (593 characters), working through the distance traveled, time constraints, and required speed calculation. In image mode, the model outputs only an answer tag (19 characters) without any intermediate reasoning, arriving at an incorrect answer. This example demonstrates how image mode suppresses the reasoning process that would normally catch arithmetic errors and guide the model to the correct solution.

Reasoning Collapse: Text Mode vs. Image Mode

Question: Marissa is hiking a 12-mile trail. She took 1 hour to walk the first 4 miles, then another hour to walk the next two miles. If she wants her average speed to be 4 miles per hour, what speed (in miles per hour) does she need to walk the remaining distance?

Text Mode Response (✓ Correct):

```
Marissa has hiked 4 miles in 1 hour, then 2 more miles in another hour.  So far, she has hiked 6 miles
in 2 hours.
She wants her average speed over the entire 12-mile trail to be 4 miles per hour.
Average speed = total distance / total time
So, to average 4 mph over 12 miles, she needs:
Total time = 12 miles / 4 mph = 3 hours
She has already used 2 hours, so she has 1 hour left to hike the remaining 6 miles.
Therefore, she needs to walk the remaining 6 miles in 1 hour.
Speed = distance / time = 6 miles / 1 hour = 6 miles per hour
<answer>6</answer>
```

Image Mode Response (✗ Incorrect):

```
<answer>12</answer>
```

Figure S11: Example of reasoning collapse on GSM8K (Qwen3-VL-8B). In text mode, the model produces detailed step-by-step reasoning (593 characters) and arrives at the correct answer. In image mode, the model skips all reasoning and outputs only an incorrect answer (19 characters).

O Human Annotation Details

This section describes the human annotators involved in the error analysis (Section 5.1) and the instructions they received.

Annotator Background. The human annotation was conducted by two students who hold graduate degrees in computer science or a related field and have prior experience with qualitative coding methods. Annotators were fluent in English and familiar with the task domains covered by the benchmarks.

Annotation Procedure. The annotation followed a human-in-the-loop grounded theory protocol, as described in Section 5.1. The annotators' responsibilities included:

1. **Reviewing proposed codes (Phase 2):** For each new code, merge, or description update proposed by GPT-5.2 during the constant comparison phase, an annotator reviewed the proposal along with

the associated error instances. The annotator approved, rejected, or modified the proposal based on whether the code accurately captured the failure mode and was consistent with existing codes.

2. **Axial coding (Phase 3):** After saturation was reached, one expert annotator grouped the refined codes into higher-level categories through discussion, producing the final seven-category taxonomy reported in Section 5.1.

Annotation Guidelines. During the open and constant comparison phases, annotators followed these principles:

- **Code granularity:** Codes should be specific enough to capture distinct failure modes but general enough to apply across multiple instances. Avoid overly narrow codes that apply to only one or two errors.
- **Mutual exclusivity:** Each error should be assignable to exactly one code. If an error appears to fit multiple codes, the annotator should either refine the code definitions or create a new code that better captures the failure.
- **Consistency:** When reviewing a proposed code, compare it against existing codes to ensure no conceptual overlap. If overlap exists, consider merging codes or clarifying boundaries.
- **Focus on root cause:** Classify errors based on the primary failure mode, not secondary symptoms. For example, if a calculation error leads to an incorrect final answer, classify it as a calculation error rather than a factual error.

Error Classification Prompt. Figure S12 shows the prompt used to classify errors. All three annotators (human expert, GPT-5.2, and Claude) used identical guidelines based on this prompt, which defines each error category with explicit criteria and provides decision rules to resolve ambiguous cases.

Inter-Annotator Agreement. To validate the reliability of our error taxonomy, we conducted an inter-annotator agreement study with three annotators: the agreement of two human experts, GPT-5.2, and Claude (claude-opus-4-5-2025-11-01). We sampled 50 errors with human labels and had each annotator independently classify them using the prompt in Figure S12. Table S7 reports pairwise Cohen's kappa (κ) and raw agreement percentages.

Annotator Pair	Cohen's κ	Raw Agreement
Human vs. GPT-5.2	0.57 (Moderate)	75.0%
Human vs. Claude	0.69 (Substantial)	81.8%
GPT-5.2 vs. Claude	0.64 (Substantial)	79.5%
Three-way agreement	—	68.2%

Table S7: Inter-annotator agreement for error classification. Cohen's κ values indicate moderate to substantial agreement across all annotator pairs.

The substantial agreement between human and model annotators ($\kappa \geq 0.57$) suggests that our error taxonomy is well-defined and can be reliably applied. The three-way agreement of 68.2% indicates that the majority of errors receive consistent classifications across all annotators, with remaining disagreements occurring at inherently ambiguous boundaries (e.g., distinguishing conceptual errors from reasoning errors when a model misapplies domain knowledge).

P Limitations

We acknowledge several limitations of this work.

Causality of the reasoning collapse Our diagnostic claim is that image-mode performance degradation is associated primarily with shortened reasoning chains rather than with perceptual or knowledge failures. We support this through three converging lines of evidence: output-length statistics, the error-category distribution shift (Figure 6a), and the recovery achieved by restoring long reasoning traces via self-distillation. While this kind of approach is common in empirical research, we would

Error Classification Prompt

You are an expert error analyst classifying why a model got a question wrong.
Categories:

1. **Conceptual/FactualRecall**: The model states or relies on an incorrect fact or misconception that you can explicitly identify. You must be able to say "The model incorrectly believes X, but the correct fact is Y." This includes questions asking about definitions of concepts.
2. **Reasoning**: The model's logic or inference is flawed. This includes: drawing wrong conclusions, faulty causal reasoning, misapplying a doctrine/rule/principle to a situation, flawed moral/legal/logical judgment, or making analytical errors. Use this when the model reasons through the problem but reaches wrong conclusions without stating an explicit factual error.
3. **Calculation/Mathematical**: Errors involving numbers or formulas. This includes: arithmetic mistakes, using the wrong formula, miscalculating values, or errors in mathematical reasoning or derivation.
4. **Incomplete/Partial**: The response is empty, truncated, or lacks a parsable answer.
5. **Format/Output**: The model's answer content is correct but formatting is wrong (missing tags, wrong structure). Never use this if the answer is wrong.
6. **Question Interpretation**: The model misread the question itself. Rarely used.
7. **Incorrect Rationale**: Only use when: (a) the model's final answer exactly matches the gold answer, and (b) the reasoning contradicts the answer. If the answer is wrong, this category cannot apply.
8. **MISC**: Truly uncategorizable. Rarely used.

Critical Rules:

- First check if the model's answer matches the gold answer. If the answer is wrong, you cannot use "Incorrect Rationale" or "Format/Output".
- Most errors are either Conceptual/FactualRecall or Reasoning. These two categories cover $\sim$90% of errors.
- Conceptual/FactualRecall vs Reasoning distinction: Conceptual/FactualRecall = model states an incorrect fact; Reasoning = model misapplies correct knowledge.
- Calculation/Mathematical: Use for any error involving numbers, formulas, or mathematical reasoning.

First, verify: Does the model's answer match the gold answer? Then classify the error type.

Figure S12: Prompt used for error classification in the inter-annotator agreement study.

like to emphasize that this evidence is *correlational and intervention-based* rather than fully causal. Several additional plausible mechanisms could co-explain the observed pattern, and our experiments do not cleanly separate them. Disentangling distributional, perceptual, and behavioral contributions is an important direction we leave to future work.

Rapidly evolving model landscape. The MLLM field is advancing at an unprecedented pace, with model providers continually refining their training recipes, data curation strategies, and architectural designs. Our evaluation captures a snapshot of seven models at a particular point in time. Indeed, we observe that newer models such as InternVL3.5-8B already exhibit smaller modality gaps, suggesting that some of the issues we identify may be addressed through improved training procedures. However, the underlying question of *how* multi-stage training paradigms and language processing methods, which differ fundamentally from human perceptual learning, shape a model's visual text understanding remains open. Our diagnostic framework and error taxonomy provide tools for continued investigation as the field evolves.

Scope of models and benchmarks. While we evaluate seven MLLMs spanning different architectures and scales, this represents a fraction of the available models. Our findings may not generalize to architectures with fundamentally different vision-language fusion mechanisms (e.g., early fusion, pixel-based tokenization). Similarly, our benchmark selection focuses on reasoning-intensive tasks; performance patterns on other task types (e.g., visual grounding, multi-image reasoning) may differ.

Statistical reporting. Our HumanEval results use single-sample pass@1 estimates over only 164 problems, which is well known to be high-variance. While we discuss this caveat, we did not generate multiple samples per problem and report unbiased pass@k estimates. HumanEval results should therefore be read as illustrative of qualitative trends rather than as precise capability measurements.

For our larger benchmarks (MMLU, ARC, GSM8K, GPQA), the sample sizes support more confident conclusions, but we did not run multiple seeds for the self-distillation experiments due to compute constraints and report results from single training runs.

Zero-shot evaluation only. All experiments are conducted in a zero-shot setting to avoid confounding factors from in-context examples. However, few-shot prompting or chain-of-thought demonstrations may mitigate the reasoning collapse we observe while introducing additional confounding factors. Future work should investigate whether the modality gap persists under different prompting strategies.

Grounded-theory methodology. Our error analysis follows Nelson's computational grounded theory framework, with GPT-5.2 performing initial open coding and constant comparison and human annotators reviewing every proposed code change and conducting axial coding. While adopting the approach from prior work, we acknowledge that this allocation differs slightly from classical grounded theory, where human interpretation drives the most generative coding phase. Our human–GPT-5.2 agreement of $\kappa = 0.57$ is "moderate" rather than "substantial," and the boundary between Conceptual/Factual and Reasoning errors—precisely the distinction underlying our central claim that image mode preserves knowledge and reasoning capability—is the most contested. Human–Claude agreement is higher ($\kappa = 0.69$), and three-way agreement is 68.2%, which we view as adequate for taxonomy validation but not fully eliminating interpretive ambiguity. Additionally, with 150 errors per model–dataset cell, fine-grained per-cell claims about error distributions are statistically limited; our findings are most reliable in aggregate and should be interpreted in aggregate.

Q Output Lengths by Correctness

Tables S8 and S9 compare response lengths before and after self-distillation, measured in whitespace-delimited words and split by answer correctness. Self-distillation often restores comparable text/image lengths: for Qwen3-VL-8B on GSM8K, the ratio falls from 33.04 to 1.02 after GSM8K training, though recovery is not universal. InternVL3-8B on MMLU often transcribes the image before answering (122.3 words in image mode versus 4.8 in text mode), so length alone does not measure reasoning quality.

Table S8: Mean response lengths (words) before and after self-distillation, split by answer correctness. T/I denote Pure Text/Pure Image; S/V denote MMMU-Pro Standard/Vision. A dash indicates unavailable outputs. Correct and incorrect means are conditioned separately within each stage.

		Baseline			**Self-distilled**		
Benchmark	Mode	All	Correct	Incorrect	All	Correct	Incorrect
Qwen3-VL-8B	Distilled on GSM8K						
GSM8K	T	121.5	93.9	521.4	112.3	86.0	530.3
	I	3.7	6.7	2.4	109.9	85.3	423.3
ARC	T	8.7	9.0	5.3	2.0	1.8	4.1
	I	3.5	3.5	3.2	8.4	8.0	11.9
MMLU	T	38.7	35.5	50.3	7.9	6.4	12.3
	I	7.7	7.8	7.4	40.2	27.1	75.0
MMMU-Pro	S	1512.4	932.3	2399.5	1437.9	790.0	2353.3
	V	1580.7	889.7	2149.3	1413.9	692.1	1992.8
Qwen3-VL-8B	Distilled on MMLU						
GSM8K	T	121.5	93.9	521.4	136.6	118.5	402.4
	I	3.7	6.7	2.4	136.9	118.8	239.6
ARC	T	8.7	9.0	5.3	216.4	210.0	282.8
	I	3.5	3.5	3.2	211.0	203.9	288.1
MMLU	T	38.7	35.5	50.3	262.1	234.0	335.0
	I	7.7	7.8	7.4	258.1	228.6	333.2
MMMU-Pro	S	1512.4	932.3	2399.5	1702.0	978.3	2700.5
	V	1580.7	889.7	2149.3	1907.8	979.0	2633.8
InternVL3-8B	Distilled on GSM8K						
GSM8K	T	131.4	141.5	95.6	148.5	146.0	171.7
	I	142.0	144.4	140.2	147.3	145.7	162.0
ARC	T	13.1	12.6	18.8	97.4	94.4	129.9
	I	2.4	2.3	3.3	41.2	38.3	65.6
MMLU	T	4.8	4.3	5.5	119.5	105.9	160.7
	I	122.3	111.0	128.1	77.7	62.1	111.0
MMMU-Pro	S	215.2	156.3	269.7	270.3	203.6	350.3
	V	194.7	150.9	216.6	300.2	213.9	341.9
InternVL3-8B	Distilled on MMLU						
GSM8K	T	131.4	141.5	95.6	–	–	–
	I	142.0	144.4	140.2	149.7	145.4	175.2
ARC	T	13.1	12.6	18.8	182.9	181.1	201.3
	I	2.4	2.3	3.3	174.5	172.9	190.3
MMLU	T	4.8	4.3	5.5	217.4	205.0	251.9
	I	122.3	111.0	128.1	201.1	187.8	236.1
MMMU-Pro	S	215.2	156.3	269.7	301.0	243.3	372.3
	V	194.7	150.9	216.6	334.7	263.3	372.6

Table S9: Text/image mean-length ratios before and after self-distillation. GSM8K and MMLU identify the distillation data; 1.00 denotes equal mean lengths. ⋆MMMU-Pro uses Standard/Vision ratios as a descriptive forgetting control. Ratios use unrounded means; a dash indicates unavailable outputs.

	Qwen3-VL-8B			**InternVL3-8B**		
		Self-distilled			Self-distilled	
Benchmark	Baseline	GSM8K	MMLU	Baseline	GSM8K	MMLU
GSM8K	33.04	1.02	1.00	0.93	1.01	–
ARC	2.48	0.24	1.03	5.55	2.37	1.05
MMLU	5.04	0.20	1.02	0.04	1.54	1.08
MMMU-Pro⋆	0.96	1.02	0.89	1.11	0.90	0.90

R Significance Tests

We mark Holm-adjusted significance ($p < 0.05$) directly in the result tables. Tests pair the same benchmark items: two-sided exact McNemar for binary correctness, and 10,000 paired bootstrap resamples for gap closure and fractional HumanEval rendering scores. Gap closure is $(A_{\mathrm{T}} - A_{\mathrm{I}})_{\mathrm{base}} - (A_{\mathrm{T}} - A_{\mathrm{I}})_{\mathrm{distilled}}$, resampling all four outcomes together. Holm families comprise 35 text–image and 105 other text–mode contrasts; 12 tests each for baseline gaps, residual gaps, and signed gap closure; and four tests per MMMU-Pro configuration. Unavailable tests retain their place in the planned families (set to $p = 1$ for adjustment only). Rendering tests use a separate 60-test family (Appendix F).

For Figure 6, we compare image/text error shares within model–benchmark pairs having at least ten coded errors in each mode, using 10,000 within-pair label permutations and Holm correction across eight categories. Format, incomplete/partial, and miscellaneous error shares differ significantly; the calculation-error contrast does not survive this benchmark-matched test. Estimated absolute category-error rates weight each pair's error share by its full-evaluation error rate.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction claim that we (1) systematically diagnose the modality gap across 7 models, 7 benchmarks, and 5 input modes (Section 3–4), (2) identify reasoning collapse as the root cause through error analysis of 4,000+ examples (Section 5), and (3) propose self-distillation to close the gap (Section 6). All claims are supported by experimental results in the corresponding sections.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We include a dedicated Limitations section in Appendix P, discussing: (1) the rapidly evolving model landscape and how multi-stage training paradigms may affect visual text understanding, (2) scope of models and benchmarks evaluated, (3) zero-shot evaluation only, (4) self-distillation tested on two architectures, and (5) annotation completeness.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: This is an empirical paper that does not include theoretical results, theorems, or formal proofs. All contributions are based on experimental evaluation and qualitative analysis.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide: model names and versions (Section 3.3), dataset descriptions and sources (Section 3.1), image rendering specifications including resolution 1280×720 and fonts (Section 3.2), decoding parameters in Appendix (temperature, batch size, max generation length), LoRA hyperparameters for self-distillation (Section 6), and detailed input format examples for all datasets (Appendix A–B). We also release all the code and data to reproduce all results.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

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

5. **Open access to data and code**

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

   Answer: [Yes]

   Justification: We release code and data at `https://github.com/KaiserWhoLearns/ReadingNotThinking`. All datasets used (MMLU, ARC, GPQA, GSM8K, HumanEval, QASPER, SQuAD) are publicly available. The repository includes evaluation scripts, image rendering code, and self-distillation training code with instructions to reproduce all experimental results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We specify decoding parameters in the Appendix, LoRA hyperparameters for self-distillation (rank r=64, learning rate 2×10^{-4}, 2 epochs, batch size 16), evaluation metrics for each dataset in the experimental setup section, and input format specifications in Appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We use bootstrap significance testing for our evaluation results to assess statistical significance of the observed performance differences across modalities and models.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report GPU types, GPU-hours, and API costs in Appendix J. FLOPs analysis comparing modalities is provided in Appendix K.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics. Our research evaluates publicly available models on public benchmarks, proposes a training method to improve model capabilities, and does not involve human subjects, deception, or potential for direct harm.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This is foundational research diagnosing and improving visual text understanding in multimodal LLMs. The work does not introduce new capabilities with direct negative societal implications beyond those already present in the underlying models. Improvements in visual text understanding are broadly beneficial for accessibility and document processing.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release pre-trained models with high misuse risk. We release evaluation code and LoRA adapters for self-distillation, which are lightweight modifications to existing public models and do not introduce new misuse vectors.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets (MMLU, ARC, GPQA, GSM8K, HumanEval, QASPER, SQuAD) and models (Qwen-VL, InternVL, Pixtral, GPT-5.2) are properly cited with their original papers. All datasets used are publicly available for research purposes.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We release evaluation code, image rendering scripts, and self-distillation training code as supplementary material. Documentation includes instructions for reproducing experiments and details about data preparation.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing. Human annotation for error analysis was conducted by domain experts (described in Appendix O).

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects. The human annotation was performed by paper authors analyzing model outputs, which does not constitute human subjects research.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We use LLMs as part of our core methodology: (1) GPT-5 as an LLM-as-judge for evaluating QASPER and SQuAD (Section 3.1), and (2) GPT-5.2 for part of the grounded-theory error analysis coding process (Section 5.1). Both usages are explicitly described in the paper.

    Guidelines:

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