PixelWorld: Towards Perceiving Everything as Pixels

Anonymous authors
Paper under double-blind review

Abstract

Recent agentic language models increasingly need to interact directly with real-world environments containing intertwined visual and textual information through raw camera pixels, rather than relying on separate image and tokenized text processing, underscoring the necessity of a *unified perception* paradigm. To close this gap, we explore this idea through Perceive Everything as Pixels (PEAP) and release PixelWorld, a benchmark that renders natural-language, tabular, mathematical and diagrammatic inputs into a single pixel space. Experiments show that PEAP attains competitive accuracy on semantic-understanding tasks, indicating that a vision transformer can capture global textual semantics without explicit tokens. In contrast, reasoning-intensive benchmarks (math and code) exhibit sharp performance drops; however, Chain-of-Thought prompting partially mitigates this gap, hinting that explicit reasoning traces compensate for the missing token structure. We also find that when visual and textual information are closely integrated, representing everything as pixels reduces preprocessing complexity and avoids misalignment issues that often arise in separate pipelines. PixelWorld therefore serves as a practical benchmark for evaluating unified vision–language models and supports broader exploration of PEAP across diverse tasks.

1 Introduction

In recent years, large vision-language models (L-VLMs) (Wang et al., 2024a; OpenAI, 2025; Team, 2024) have achieved impressive performance across a wide range of real-world tasks. These models typically process visual inputs as pixels and textual inputs as discrete tokens—two distinct modalities that are handled separately. However, such modality-specific processing leads to a fragmented understanding of multimodal inputs and increases the complexity of engineering pipelines. This separation becomes particularly problematic in modern agent-based systems such as Computer Agents (Zheng et al., 2024; Koh et al., 2024) and Embodied Agents (Tellex et al., 2020; Driess et al., 2023), which are increasingly expected to perform complex real-world tasks including navigation in physical environments (Elnoor et al., 2024), booking flights (Chen et al., 2024a), and repairing software bugs on platforms like GitHub (Yang et al., 2024). These tasks involve deeply intertwined visual and textual information, where decoupled tokenization and perception modules can result in high preprocessing overhead (Xie et al., 2024; Koh et al., 2024) and degraded performance due to information loss and layout inconsistencies (Dagan et al., 2024; Chai et al., 2024).

Due to these limitations, we propose a unified perception paradigm: Perceive Everything as Pixels (PEAP). In this paradigm, both text and visual inputs are treated uniformly in the pixel space, allowing a vision-language model (VLM) to jointly model multimodal inputs without separate tokenization or modality-specific encoders. To identify the benefits and challenges of this paradigm, we introduce PixelWorld, a comprehensive benchmark suite designed to assess how well VLMs perform on existing benchmarks under the PEAP setting.

In PixelWorld, we select 10 representative commonly used benchmarks, covering a diverse range of modalities and task scenarios. For each dataset, we construct both traditional token-based and pixel-based (PEAP) input formats using image synthesis and OCR techniques (see Table 1). We then evaluate vision–language models of varying scales, from Qwen2VL-2B to GPT-4o. Cross-modal evaluation in Section 3 yields three overarching insights: **Insight 1**: in intrinsically multimodal settings such as website rendering, slide compre-

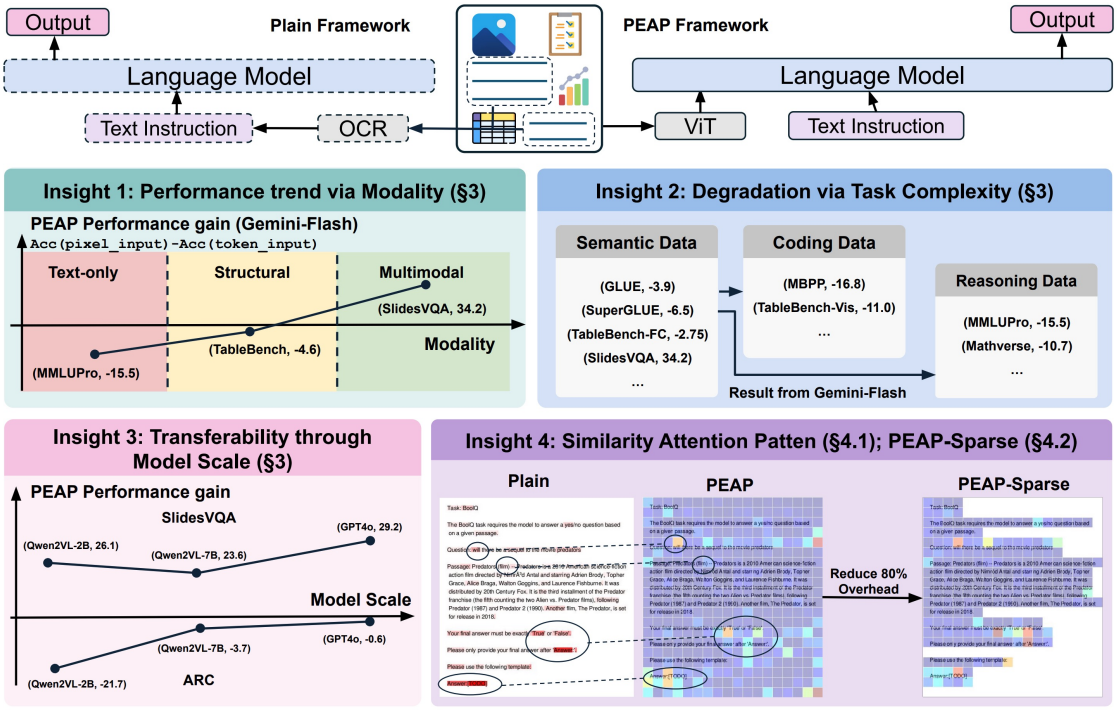


Figure 1: PEAP framework: we investigate the possibility of perceive everything as pixels. This framework aligns better with human perception reducing the need for excessive pre-processing. Evaluated on our benchmark PixelWorld, PEAP boosts performance on multimodal tasks (e.g., websites, slides, documents) but struggles with complex, text-centric tasks (e.g., reasoning and coding). Larger models achieve better transferability between pixel- and token-based performance compared to smaller ones. We also observed that text and images exhibit similar attention patterns, and reduced the overhead of model reasoning through patch pruning by PEAP-Fast.

hension, and document understanding, PEAP eliminates OCR noise and consistently boosts performance; **Insight 2**: pixelising inputs for reasoning intensive tasks such as math and code incurs marked accuracy drops, yet the gap narrows as model capacity grows, suggesting scale is critical for cross-modal transfer; **Insight 3**: larger models demonstrate superior instruction-following and long-context reasoning across modalities, while smaller models struggle, emphasizing the need for scale-aware training under the pixel paradigm.

To further understand these findings, we conduct additional analyses from three perspectives: (1) Representation analysis: We visualize the attention patterns of Qwen2VL-7B and find consistent global structures between token- and pixel-based inputs, suggesting that vision encoders can serve as universal tokenizers. (2) Efficiency optimization: We measure inference latency and show that while PEAP increases processing time due to input size, our proposed PEAP-Fast algorithm prunes blank patches and achieves up to 80% speedup without accuracy loss. (3) Prompt sensitivity: We explore input prompting strategies and find that Chain-of-Thought (CoT) boosts performance on PEAP more effectively than standard input.

In summary, our contributions are as follows:

1. **PixelWorld**: We introduce a unified benchmark that converts text, structural, and multimodal datasets into pixels, providing a direct stress test and diagnostic tool for PEAP. The dataset and evaluation code are publicly released to foster a more holistic yardstick for future L-VLMs research and to ease multimodal data collection.

Dataset Name	Size	Task	Modality Transfer	Split
Text-only				
GLUE Wang (2018)	59,879	Natural language understanding	Synthesis	test
SuperGLUE Sarlin et al. (2020)	19,294	Natural language understanding	Synthesis	test
MMLU-Pro Wang et al. (2024b)	12,032	Domain knowledge and reasoning	Synthesis	test
ARC Clark et al. (2018)	3,548	Science question answering	Synthesis	test
GSM8K Cobbe et al. (2021)	1,319	Math problem solving	Synthesis	test
MBPP Austin et al. (2021)	757	Programming tasks	Synthesis	test
Structured				
TableBench Wu et al. (2024)	888	Table data understanding and analysis	Synthesis	test
Multimodal				
MathVerse Zhang et al. (2025)	788	Math and visual reasoning	Natural	test
MMMU-Pro Yue et al. (2024)	1,730	Multimodal reasoning	Synthesis	test
SlidesVQA (Tanaka et al., 2023)	2,136	Multimodal question answering	OCR	test
Wiki-SS (Ma et al., 2024)	3,000	Multimodal retrieval question answering	OCR	train

Table 1: Overview of datasets categorized by modality, usage, size, and split. Modality Transfer means the method to adopt the dataset into counterpart modality. For OCR, we adopt the result from the origin datasets. For WikiSS-QA, since the positive document of the test set is not released, we subsample 3,000 training data points randomly to evaluate.

2. **Task–scale insights**: PEAP consistently improves layout-heavy or intrinsically multimodal tasks (e.g., website and document understanding) but degrades on reasoning- and code-centric benchmarks; the performance gap diminishes as model size increases, highlighting scale as a key factor for transferability.
3. **Efficiency & interpretability**: We propose PEAP-Fast, which removes blank pixel patches to achieve up to a 3× latency reduction without harming accuracy. Attention visualizations reveal similar global patterns between pixel- and token-based models, suggesting that vision encoders can act as a universal multimodal tokenizer.

2 Datasets

Several representative datasets covering different skill domains are selected, as shown in Table 1. We primarily utilize the prompts provided by the datasets. If no prompts are available, we apply a default prompt. By default, we employ Direct Prompting; however, for more complex and mathematical datasets such as MBPP (Austin et al., 2021), MMLU-Pro (Wang et al., 2024b), and MathVerse (Zhang et al., 2025), we adopt Chain-of-Thought (CoT) prompting to enhance performance. All evaluations are conducted in a zero-shot manner to mitigate potential performance degradation caused by the sensitivity of instruction-tuned large models to few-shot prompting.

To evaluate both Token-based and Pixel-based methods, we require paired Text-input and Image-input prompts. We adopted modality transfer strategies to reduce reliance on the information modality provided by existing datasets, as detailed in Table 1. For datasets categorized as *Text-Only* and *Structured*, all data is originally in plain text format, necessitating image synthesis prior to evaluation. For *Multimodal* datasets, textual content embedded in images is extracted using OCR, or the textual components provided by the original datasets are directly utilized for evaluation. Notably, the MathVerse dataset (Zhang et al., 2025) inherently includes a Text-Only modality, offering detailed textual descriptions of image-based information.

Image Data Synthesis For text-only and structured datasets, we developed an image data synthesis pipeline to generate diverse image inputs for evaluation. Image widths were adaptively adjusted between 512 and 1024 pixels based on text length, with a fixed height of 256 pixels. Font sizes ranged from 15 to 25 points, and padding varied from 5 to 30 pixels. To enhance robustness, we applied various types of noise,

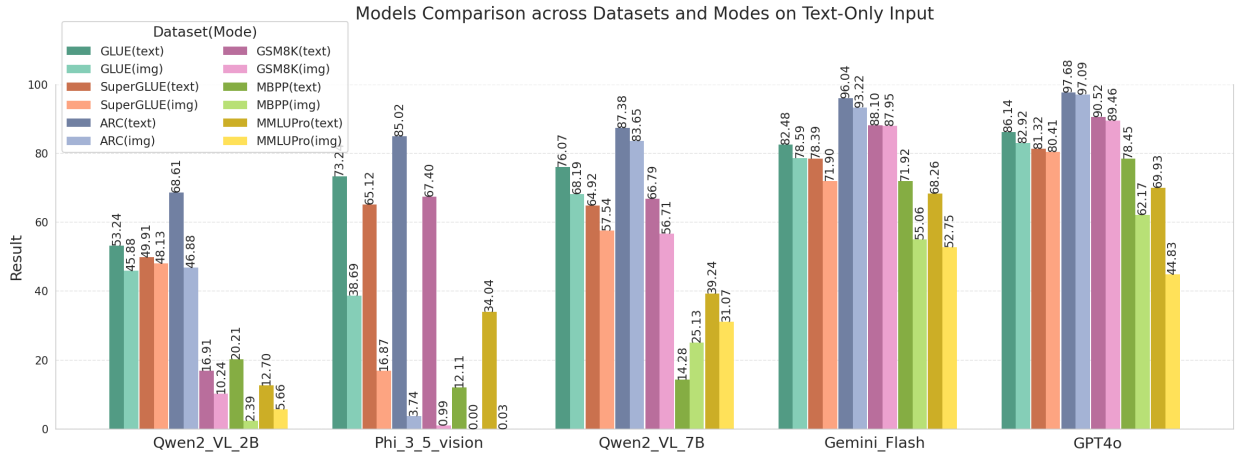


Figure 2: The performance of **text-only** datasets. The comparison is made between text input and synthesized image input. Most models demonstrate comparable performance on language understanding datasets such as SuperGLUE, GLUE, and ARC. However, notable performance disparities emerge between text-based input and synthesized image input on mathematical reasoning tasks (e.g., MMLU-Pro, GSM8K) and programming tasks (e.g., MBPP). Phi-3.5-Vision exhibits consistently poor performance across all vision tasks, primarily due to its insufficient instruction-following capabilities.

including radial, horizontal, vertical, and Multi-Gaussian noise, as well as high-frequency Gaussian noise to simulate distortions commonly introduced by real-world cameras. For structured datasets, such as tables, data was rendered as images using the Python package `dataframe_image`. Example inputs from different tasks are provided in Appendix A.

3 Experiments

In this section, we will detail our baseline, metrics and models. The experimental results will be organized by 'Text Input', 'Structued Input' and 'Multimodal Input'.

Baseline We establish the baseline by using the same VLMs with text-only prompts. To ensure fairness, we employ identical prompts and add the instruction *"Please follow the instruction in the image"* when applying PEAP. This ensures that the VLMs can correctly process instructions embedded within images. Ideally, the baseline and PEAP should yield equivalent performance. This comparison helps identify areas for improvement in existing VLMs.

Metrics For question-answering tasks such as *WikiSS-QA*, *SlidesVQA*, and *TableBench*, we adopt *ROUGE-L* as our primary metric, as it effectively captures the alignment between generated answers and ground truth by measuring the longest common subsequence. For classification benchmarks, including *MMLU-Pro*, *GLUE*, *SuperGLUE*, *ARC*, and *MathVerse*, we use accuracy, which directly reflects the model's performance in selecting correct options. For *GLUE* and *SuperGLUE*, we follow their standard evaluation protocols, utilizing task-specific metrics such as Matthews correlation, F1 score, and Pearson correlation. For the code generation task *MBPP*, we evaluate performance using the pass@1 rate, which measures whether the generated code successfully passes all test cases. For the mathematical reasoning dataset *GSM8K*, we employ exact match accuracy, as these problems require precise numerical answers. For the visualization subtask of *TableBench*, following the original codebase, we treat it as a code generation task and evaluate the correctness of the generated visualizations.

Model Selection To validate PixelWorld, we selected a diverse set of vision-language models (VLMs) with varying scales to ensure the robustness and generalizability of our findings. It also allowed us to analyze the behavior of models across different sizes. We evaluated several widely used vision-language models

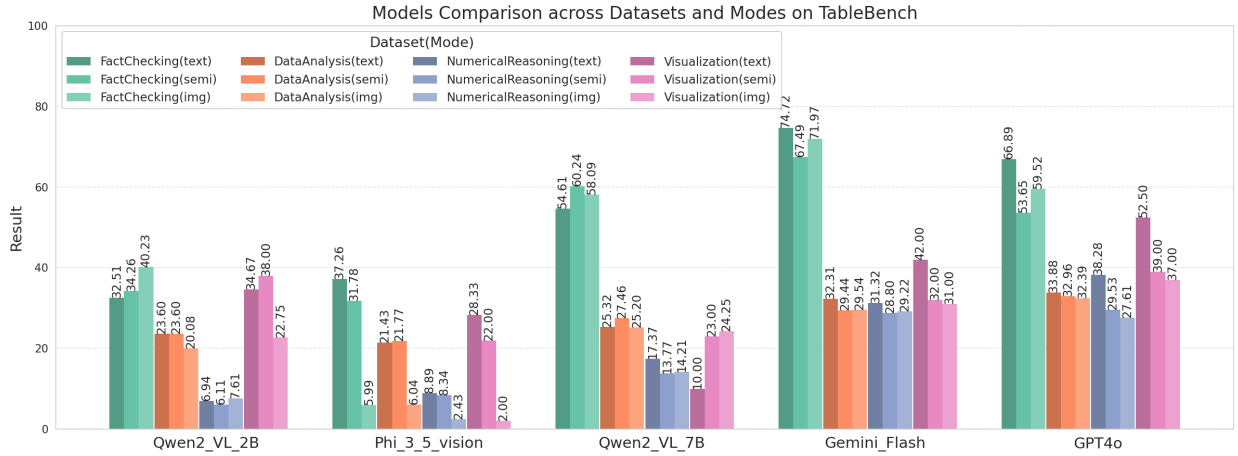


Figure 3: The performance of the **structured** dataset. We report all the subsets for the TableBench. In the *semi* setting, questions were presented as text, while tables were rendered as synthetic images. We observed that for tasks involving reasoning (numerical reasoning) and coding (visualization subset), synthetic images yielded inferior performance compared to text. However, for tasks emphasizing semantic understanding, such as data analysis and fact checking, synthetic images achieved performance comparable to or even surpassing text. Additionally, we found that the semi approach often performed worse than either text or synthetic images individually, providing insights into potential limitations and future directions for leveraging vision-language models (VLMs).

(VLMs), including Qwen2VL-2B Wang et al. (2024a), Phi-3.5-3.2B Abdin et al. (2024), Qwen2VL-7B Wang et al. (2024a), Gemini-Flash Team (2024), and GPT-4o OpenAI (2025).

3.1 Text Input

Figure 2 reports model accuracy on text-only datasets (e.g., ARC, MMLU-Pro, GLUE, GSM8K, SuperGLUE, MBPP). Two major insights emerge:

Better Transferability in Larger Models Larger language models (e.g., GPT-4o, Gemini-Flash) exhibit better transferability between text and image-based performance, while smaller models struggle with both transferability and instruction following. For instance, on the ARC dataset, GPT-4o's performance declines by only 0.59 points when transitioning from text to synthetic images, whereas the smaller Qwen2-VL-2B suffers a substantial 21.73-point drop (from approximately 68.61 to 46.88). This trend suggests that more capable models preserve their reasoning abilities across modalities, while smaller models face greater difficulty. Additionally, smaller models (e.g., Phi-3.5-vision) not only show weaker overall performance on standard benchmarks but also struggle significantly when instructions are presented as images. Their performance consistently lags behind that of larger models, particularly on tasks like MBPP. This supports *Insight 3* in Figure 1.

Performance Degradation with More Complex Tasks We observe significant drops on benchmarks requiring advanced reasoning, such as mathematical, coding or domain-specific tasks. For example, when moving from text to image inputs on the MMLU-Pro dataset, GPT-4o exhibits a drop of more than 25 points. In contrast, on GLUE and SuperGLUE, the decline remains under 5 points. These findings indicate that while existing large models achieve comparable performance between text and visual modalities on simpler tasks, a gap still exists at a deeper level in visual-based and text-based understanding, demonstrating room for improvement in modality adaptation training.

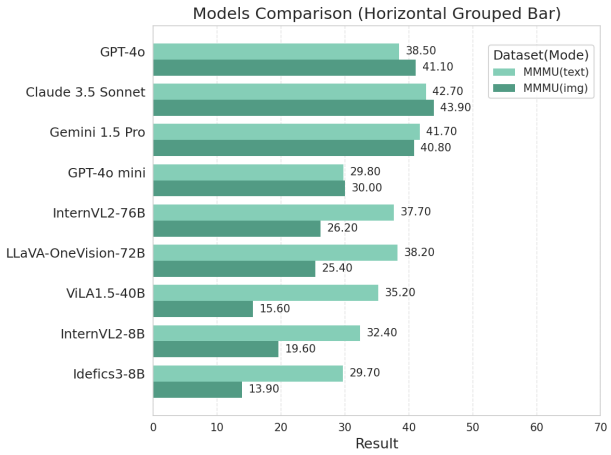


Figure 4: The performance of the **multimodal** dataset (MMMU-Pro). We adopt the result reported by the origin paper. We can observe that strong models perform better in PEAP.

3.2 Structured Input

Figure 3 summarizes model performance on four TableBench subsets: Fact Checking, Data Analysis, Numerical Reasoning, and Visualization.

Reasoning Complexity Impacts Performance Fact Checking and Data Analysis show moderate performance drops, as they rely on semantic understanding. In contrast, Numerical Reasoning and Visualization—requiring more intricate reasoning and coding—exhibit larger declines when switching to synthetic images. Combined with *"Performance Degradation with More Complex Tasks"* in Section 3.1, this supports *Insight 2* in Figure 1.

Smaller Performance Gaps with Structured Data Compared to text-only tasks, structured tasks show smaller performance gaps between text and image inputs. Notably, Qwen2VL-2B even outperforms its text-based results on Fact Checking, suggesting robust visual representations can aid semantic tasks in smaller models.

Challenges with Mixed-Modality Inputs The "semi" format—where tables appear as images while questions remain text-based—performs worse than either fully text-based or fully image-based formats. This suggests that conventional VQA approaches, which process text and images using separate encoders, may be more susceptible to performance bottlenecks. As multimodal scenarios become increasingly prevalent, PEAP is expected to demonstrate superior performance compared to mixed-modality methods.

3.3 Multimodal Input

Figure 5 presents model performance on multimodal datasets, including text-only and vision-only subsets of Mathverse and VQA tasks like SlidesVQA and WikiSS-QA. Results on MMMU-Pro (Figure 4) use reported values from the original paper. Three key observations emerge:

Image Inputs Enhance Disambiguation Incorporating images improves performance by reducing ambiguity compared to text-only benchmarks. In SlidesVQA, all models outperform their text-only baselines, while in WikiSS-QA and MMLU-Pro, visual context provides clarifying information, leading to accuracy gains in larger models. Combined with *"Smaller Performance Gaps with Structured Data"* in Section 3.2, this supports *Insight 1* in Figure 1.

Challenges in Complex Reasoning While multimodal inputs aid basic tasks, complex reasoning remains a bottleneck. In Mathverse, visual cues help but fail to support multi-step logical deductions. Even Gemini-Flash shows accuracy drops on intricate reasoning tasks. Additionally, WikiSS-QA poses challenges due to its long-context nature. Smaller models struggle with PEAP, and GPT-4o underperforms in token-based

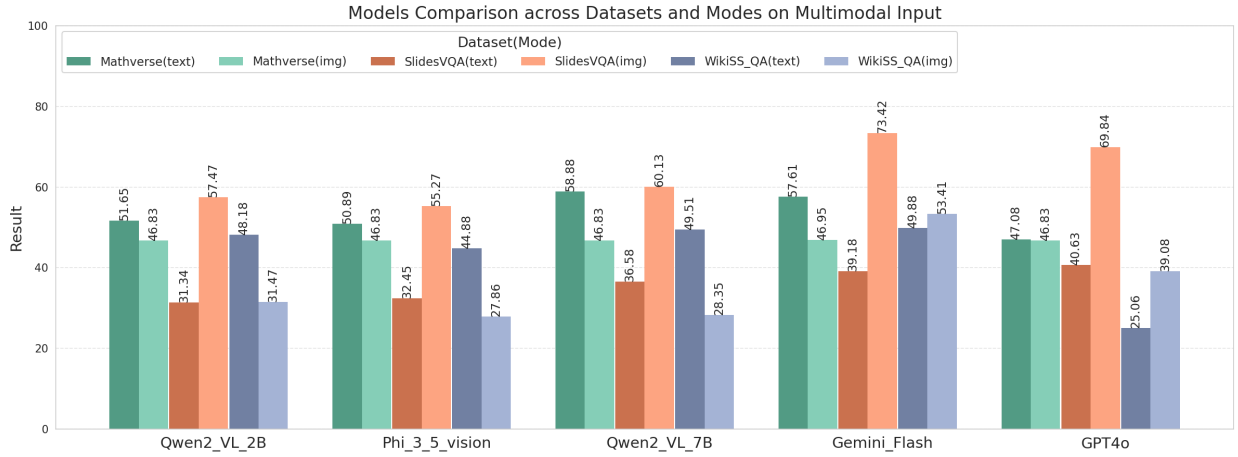


Figure 5: The performance of the **multimodal** datasets (except MMMU-Pro). We compare text-only and vision-only subsets in Mathverse, while SlidesVQA and WikiSS-QA are evaluated as VQA tasks. Larger models perform better on text-based tasks with more modalities. GPT-4o tends to generate longer responses in long-context QA, leading to performance degradation on WikiSS-QA.

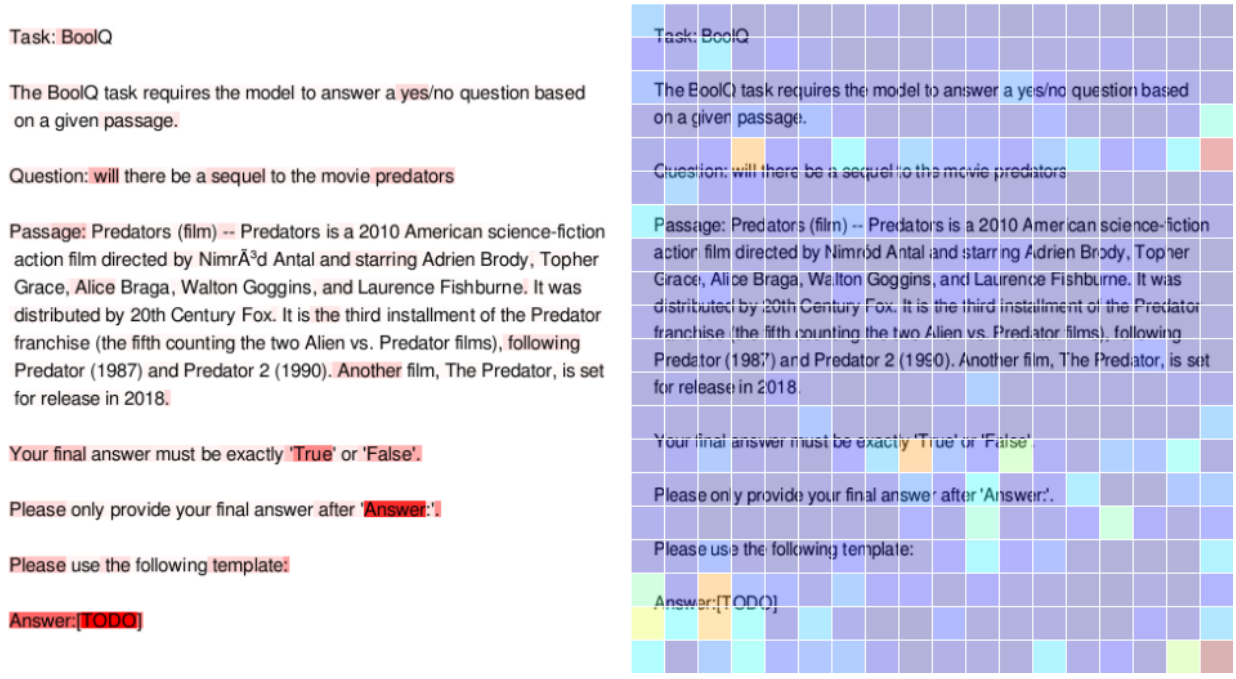


Figure 6: Last Layer Attention Heatmap on Qwen2VL-7B between token-based (left) and pixel-based (right) inference.

tasks, highlighting difficulties in processing extended contextual dependencies. This aligns with Sections 3.1 and 3.2.

Larger Models Benefit More from Multimodal Data Larger models gain more from multimodal inputs. On SlidesVQA, Gemini_Flash improves by 34.24 points, compared to Qwen2-VL-7B's 23.55-point boost. This suggests that larger models, with more extensive prior knowledge and advanced architectures, leverage multimodal data more effectively than smaller models.

	SuperGLUE Evaluation Results		
Task	**Text**	**PEAP**	**PEAP-Fast**
BoolQ	79.69%	82.11%	80.89%
CB	67.70%	40.77%	39.57%
COPA	93.00%	91.00%	86.00%
MultiRC	65.90%	61.28%	60.80%
ReCoRD	12.54%	5.94%	6.08%
RTE	82.31%	72.92%	77.26%
WiC	53.29%	55.80%	55.64%
WSC	63.46%	65.38%	59.62%
Final Score	64.74%	59.40%	58.23%

Table 2: Performance of *Qwen2VL-7B* on SuperGLUE dataset by Text, PEAP and PEAP-Fast. We can observe the comparable performance between PEAP and PEAP-Fast.

	Inference Time (s)			Overhead (%)	
Subset	**Text**	**PEAP**	**PEAP-Fast**	**PEAP**	**PEAP-Fast**
BoolQ	369	1,381	906	274.80	145.55
CB	8	22	15	175.00	87.50
COPA	39	38	22	-2.56	-43.59
MultiRC	609	3,861	2,550	534.80	318.71
ReCoRD	7,016	19,012	14,288	171.01	103.72
RTE	68	117	92	72.06	35.29
WiC	69	224	157	224.64	127.54
WSC	11	36	27	227.27	145.45
Total	8,089	24,690	18,051	205.27	123.19

Table 3: Inference Time (s) of *Qwen2VL-7B* on SuperGLUE dataset with single A100 server by PEAP and PEAP-Fast. We can observe a 82.08% overhead reduce on PEAP-Fast method. Overhead is calculated as the percentage increase in time relative to the text method.

4 Discussion

4.1 Q1: Does PEAP have the same attention?

To investigate whether VLMs behaves similarly on textual and image inputs, we visualized the Average Attention of Qwen2-VL-7B's final layer using a heatmap (see Figure 6). Concretely, we examined its responses on a SuperGLUE *BoolQ* example, comparing the model's attention maps for text-based versus image-based inference.

As shown in Figure 6, the model largely focuses on task-relevant elements such as the question prompt ("will there be a sequel ..."), the key words in the passage (e.g., "film", "starring", "Alice"), and the required answer format ("Answer: True/False"). This holds true across both textual and visual representations, indicating Qwen2-VL-7B exhibits comparable attention patterns irrespective of input modality. However, we also observe that certain blank patches in the image-based input can receive disproportionately high attention. This suggests that while the visual encoder parallels the text encoder in many respects, it still has redundancy.

4.2 Q2: How to make PEAP more efficient?

As a trade-off for generalization, image-based inference often requires significantly more computational resources than text-based inference. This is partly due to the additional overhead from the ViT backbone and higher redundancy in image tokens. To estimate the performance gap quantitatively, we conducted

Metric	Direct		CoT		Improve (CoT - Direct)	
	Text	PEAP	Text	PEAP	Text	PEAP
BoolQ	79.88%	81.71%	81.13%	80.73%	1.25%	-0.98%
CB	67.70%	34.78%	81.04%	59.57%	13.34%	24.79%
COPA	93.00%	87.00%	89.00%	83.00%	-4.00%	-4.00%
MultiRC	65.73%	62.28%	69.08%	60.41%	3.35%	-1.87%
ReCoRD	12.50%	5.88%	6.37%	4.66%	-6.13%	-1.22%
RTE	82.31%	72.92%	83.03%	77.26%	0.72%	4.34%
WiC	52.82%	54.39%	54.39%	53.92%	1.57%	-0.47%
WSC	65.38%	61.54%	57.69%	61.54%	-7.69%	0.00%
Overall	64.92%	57.56%	65.22%	60.14%	0.30%	2.58%

Table 4: Comparison of Direct and CoT performance across Text and Image modalities, along with their respective improvements (CoT - Direct), presented as percentages.

experiments on SuperGLUE (Table 2). The results show that inference latency for image-based inputs can exceed text-based methods by 150% to 250%.

To reduce redundancy in visual inputs, we propose **PEAP-Fast**, which first identifies empty patches via a simple variance-based threshold—if the pixel-value variance in a patch is lower than a preset threshold, that patch is treated as empty and is pruned from all attention computations. Crucially, we preserve the original positional embeddings for the remaining tokens, ensuring no loss of spatial layout perception. This strategy aligns with how humans naturally focus on salient regions rather than blank spaces, thereby significantly reducing context length without sacrificing structural information. Testing PEAP-Fast on SuperGLUE reveals a minor accuracy drop of only 1.17% (Table 2). More importantly, the average overhead decreases from 205.27% to 123.19%, yielding an 82.98% reduction (Table 3). These results demonstrate that removing empty patches offers substantial computational savings while maintaining strong performance, making image-based inference more practical for real-world deployments. Attention heatmap between PEAP and PEAP-Fast are shown in Appendix B.

4.3 Q3: Is PEAP sensitive to the prompting method?

Massive experimental results in Section 3 show that the performance gap between image and text inputs still exists, potentially due to domain gaps in datasets or insufficient instruction following in image inputs. To address this, we applied CoT-style prompts to the SuperGLUE dataset to enhance cross-domain instruction following (Table 4). Notably, Qwen2VL-7B showed significant improvements in tasks where image input underperformed compared to text input, such as CB and RTE. Overall, CoT prompts improved image input performance by 2.58%, surpassing the 0.3% improvement observed for text input.

5 Related Work

Multimodal Large Language Models and Benchmarks Recent progress in multimodal AI has led to the development of models like GPT-4o OpenAI (2025), Gemini Team (2024), and Claude-3.5 Anthropic (2025), which integrate vision-based training to improve instruction-following capabilities. Benchmarks for these models have evolved from task-specific datasets, such as VQA Agrawal et al. (2016) and DocVQA Mathew et al. (2021), to more comprehensive evaluations, including MMMU-Pro Yue et al. (2024), MMBench Liu et al. (2024), and MegaBench Chen et al. (2024b). However, most current research focuses on the semantic understanding of visual content, with only a few benchmarks—such as MathVerse Zhang et al. (2025) and MMMU-Pro Yue et al. (2024)—addressing text recognition and comprehension within images. Our work shifts the focus towards evaluating how well large language models understand language through visual input compared to traditional token-based input.

Screenshot LMs Recent studies have demonstrated that pretraining on synthetic screenshots can enable vision-language models (VLMs) to achieve performance comparable to that of BERT on language modeling tasks Lee et al. (2022); Rust et al. (2023); Gao et al. (2024). This approach allows models to better capture text structures without relying on OCR-based methods. Furthermore, our analysis highlights a performance gap between existing VLMs on vision-based tasks and their text-only counterparts, particularly in the absence of relevant pretraining. Interestingly, in certain scenarios, VLMs perform as well as or even better than text-only models, underscoring the potential of this research direction. In the context of document retrieval, recent advancements Faysse et al. (2024); Ma et al. (2024) have shown that large-scale pretraining on screenshots can outperform traditional OCR-based methods, further reinforcing the advantages of vision-language pretraining.

Language Tokenization Tokenization methods, such as Byte Pair Encoding (BPE) Shibata et al. (1999); Sennrich et al. (2016), are widely used in language modeling, but recent studies suggest that they may not always be optimal. For instance, MegaByte Yu et al. (2023) demonstrated that fixed-length tokenization can improve both computational efficiency and cross-modal capabilities. Similarly, BLT Pagnoni et al. (2024) proposed entropy-based tokenization, while LCM team et al. (2024) emphasized the benefits of processing higher-level semantic concepts rather than individual tokens. Inspired by these approaches, we explore whether adaptive image patches can effectively infer textual meaning. At a higher level, we investigate the unification of text and image inputs into a shared representation space, enabling reasoning through abstract semantic concepts rather than traditional token-based methods.

6 Conclusion

We present PixelWorld, a benchmark that renders text, tables, code, and images as pixels, enabling direct evaluation of the Perceive Everything as Pixels (PEAP) paradigm. Experiments yield three takeaways: **(1) Semantic understanding.** PEAP matches token baselines on sentence-/paragraph-level tasks, while its patch-level attention closely mirrors token attention, pointing toward "vision-as-token" models. **(2) Reasoning.** Accuracy drops on math, logic, and program-repair benchmarks; Chain-of-Thought prompts narrow but do not eliminate this gap. **(3) Multimodal tasks.** Pixel input outperforms OCR pipelines on websites, slides, and documents by preserving spatial context and avoiding recognition errors. To curb the higher latency of pixel inputs, we introduce PEAP-Fast, which prunes blank patches and accelerates inference by up to 3× without hurting accuracy. Taken together, these findings underscore both the promise and trade-offs of PEAP. PixelWorld thus serves as a practical stress test and diagnostic benchmark, encouraging the community to adopt PEAP as a holistic yardstick while guiding research on efficiency improvements and on closing the reasoning gap in next-generation multimodal agents.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt,

Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering, 2016. URL https://arxiv.org/abs/1505.00468.

Anthropic. Claude 3.5: A sonnet of progress, 2025. URL https://www.anthropic.com/news/claude-3-5-sonnet. Accessed: 2025-01-13.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Yekun Chai, Yewei Fang, Qiwei Peng, and Xuhong Li. Tokenization falling short: On subword robustness in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1582–1599, 2024.

Aili Chen, Xuyang Ge, Ziquan Fu, Yanghua Xiao, and Jiangjie Chen. Travelagent: An ai assistant for personalized travel planning. *arXiv preprint arXiv:2409.08069*, 2024a.

Jiacheng Chen, Tianhao Liang, Sherman Siu, Zhengqing Wang, Kai Wang, Yubo Wang, Yuansheng Ni, Wang Zhu, Ziyan Jiang, Bohan Lyu, Dongfu Jiang, Xuan He, Yuan Liu, Hexiang Hu, Xiang Yue, and Wenhu Chen. Mega-bench: Scaling multimodal evaluation to over 500 real-world tasks, 2024b. URL https://arxiv.org/abs/2410.10563.

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.

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.

Gautier Dagan, Gabriel Synnaeve, and Baptiste Roziere. Getting the most out of your tokenizer for pre-training and domain adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: an embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 8469–8488, 2023.

Mohamed Elnoor, Kasun Weerakoon, Gershom Seneviratne, Ruiqi Xian, Tianrui Guan, Mohamed Khalid M Jaffar, Vignesh Rajagopal, and Dinesh Manocha. Robot navigation using physically grounded vision-language models in outdoor environments. *arXiv preprint arXiv:2409.20445*, 2024.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models, 2024. URL https://arxiv.org/abs/2407.01449.

Tianyu Gao, Zirui Wang, Adithya Bhaskar, and Danqi Chen. Improving language understanding from screenshots, 2024. URL https://arxiv.org/abs/2402.14073.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding, 2022. URL https://arxiv.org/abs/2210.03347.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024. URL https://arxiv.org/abs/2307.06281.

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

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021. URL https://arxiv.org/abs/2007.00398.

OpenAI. Hello gpt-4o, 2025. URL https://openai.com/index/hello-gpt-4o/. Accessed: 2025-01-13.

Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, et al. Byte latent transformer: Patches scale better than tokens. *arXiv preprint arXiv:2412.09871*, 2024.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FkSp8VW8RjH.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4938–4947, 2020.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. Byte pair encoding: A text compression scheme that accelerates pattern matching. 1999.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images, 2023. URL https://arxiv.org/abs/2301.04883.

Gemini Team. Gemini: A family of highly capable multimodal models, 2024. URL https://arxiv.org/abs/2312.11805.

LCM team, Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R. Costa-jussà, David Dale, Hady Elsahar, Kevin Heffernan, João Maria Janeiro, Tuan Tran, Christophe Ropers, Eduardo Sánchez, Robin San Roman, Alexandre Mourachko, Safiyyah Saleem, and Holger Schwenk. Large concept models: Language modeling in a sentence representation space, 2024. URL https://arxiv.org/abs/2412.08821.

Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):25–55, 2020.

Alex Wang. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

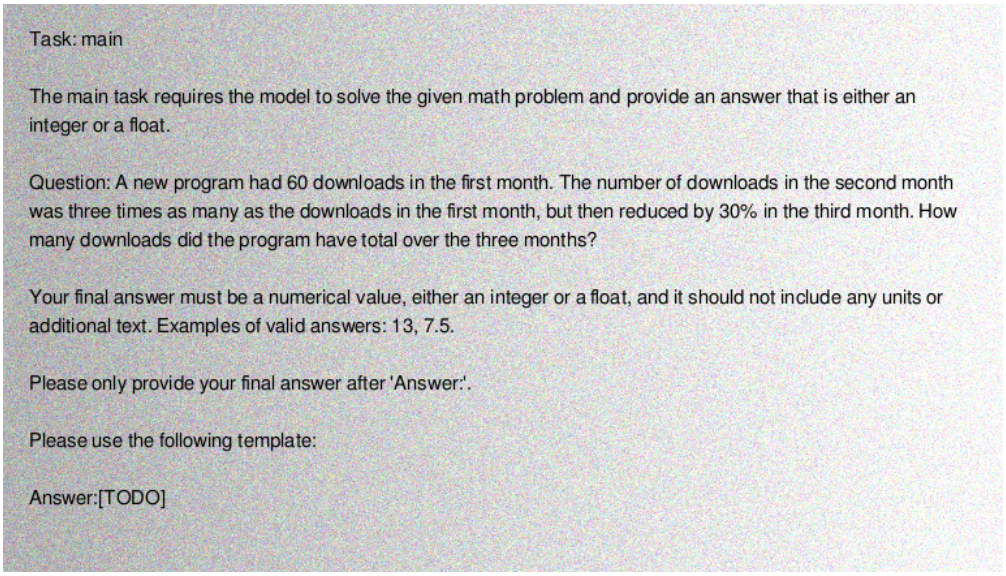


Figure 7: An example input of GSM8K dataset, using Direct Prompt.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024b.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xinrun Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, et al. Tablebench: A comprehensive and complex benchmark for table question answering. *arXiv preprint arXiv:2408.09174*, 2024.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024.

John Yang, Carlos E Jimenez, Alex L Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriel Synnaeve, Karthik R Narasimhan, et al. Swe-bench multimodal: Do ai systems generalize to visual software domains? *arXiv preprint arXiv:2410.03859*, 2024.

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. Megabyte: Predicting million-byte sequences with multiscale transformers. *Advances in Neural Information Processing Systems*, 36:78808–78823, 2023.

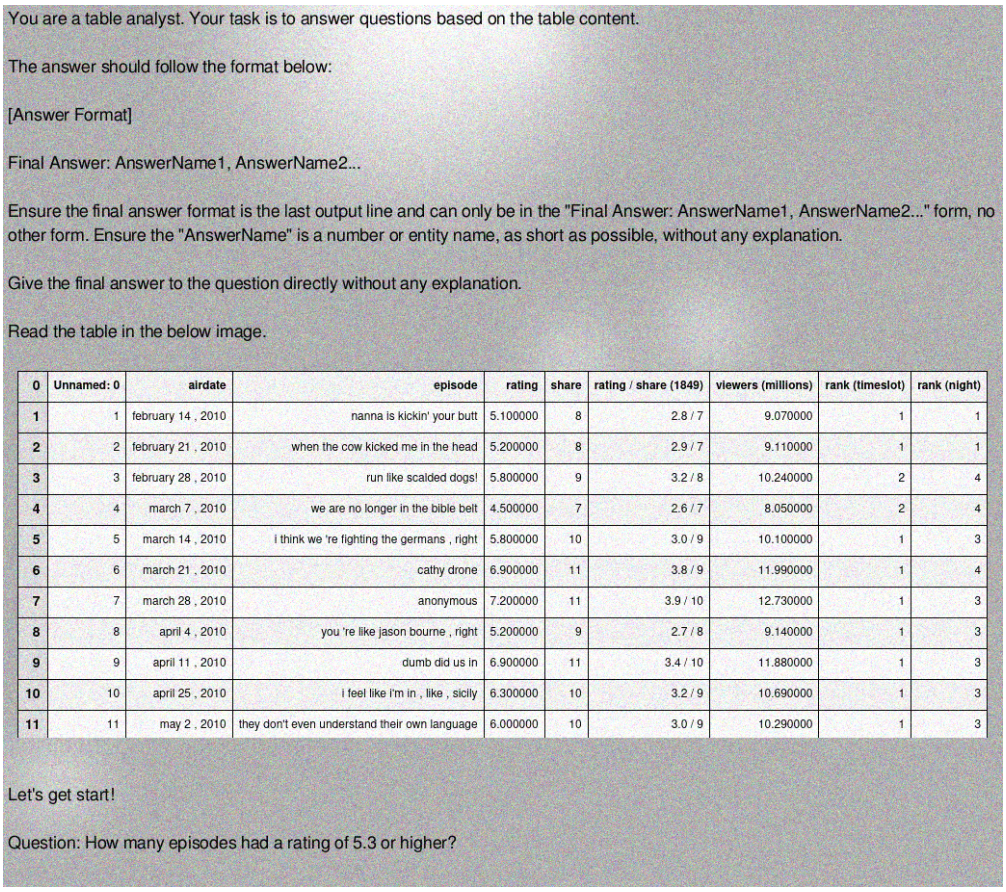
Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark, 2024. URL `https://arxiv.org/abs/2409.02813`.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2025.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. In *Forty-first International Conference on Machine Learning*, 2024.

A Example Input

Figure 7 and Figure 8 gives two examples about the vision input.



0	Unnamed: 0	airdate	episode	rating	share	rating / share (1849)	viewers (millions)	rank (timeslot)	rank (night)
1	1	february 14 , 2010	nanna is kickin' your butt	5.100000	8	2.8 / 7	9.070000	1	1
2	2	february 21 , 2010	when the cow kicked me in the head	5.200000	8	2.9 / 7	9.110000	1	1
3	3	february 28 , 2010	run like scalded dogs!	5.800000	9	3.2 / 8	10.240000	2	4
4	4	march 7 , 2010	we are no longer in the bible belt	4.500000	7	2.6 / 7	8.050000	2	4
5	5	march 14 , 2010	i think we 're fighting the germans , right	5.800000	10	3.0 / 9	10.100000	1	3
6	6	march 21 , 2010	cathy drone	6.900000	11	3.8 / 9	11.990000	1	4
7	7	march 28 , 2010	anonymous	7.200000	11	3.9 / 10	12.730000	1	3
8	8	april 4 , 2010	you 're like jason bourne , right	5.200000	9	2.7 / 8	9.140000	1	3
9	9	april 11 , 2010	dumb did us in	6.900000	11	3.4 / 10	11.880000	1	3
10	10	april 25 , 2010	i feel like i'm in , like , sicily	6.300000	10	3.2 / 9	10.690000	1	3
11	11	may 2 , 2010	they don't even understand their own language	6.000000	10	3.0 / 9	10.290000	1	3

Figure 8: An example input of TableBench dataset, using Direct Prompt.

B Attention Heatmap before and after ImageFast Method

Figure 9 presents a heatmap comparison between PEAP and PEAP-Fast. PEAP-Fast effectively reduces redundant patches while preserving attention on key regions.

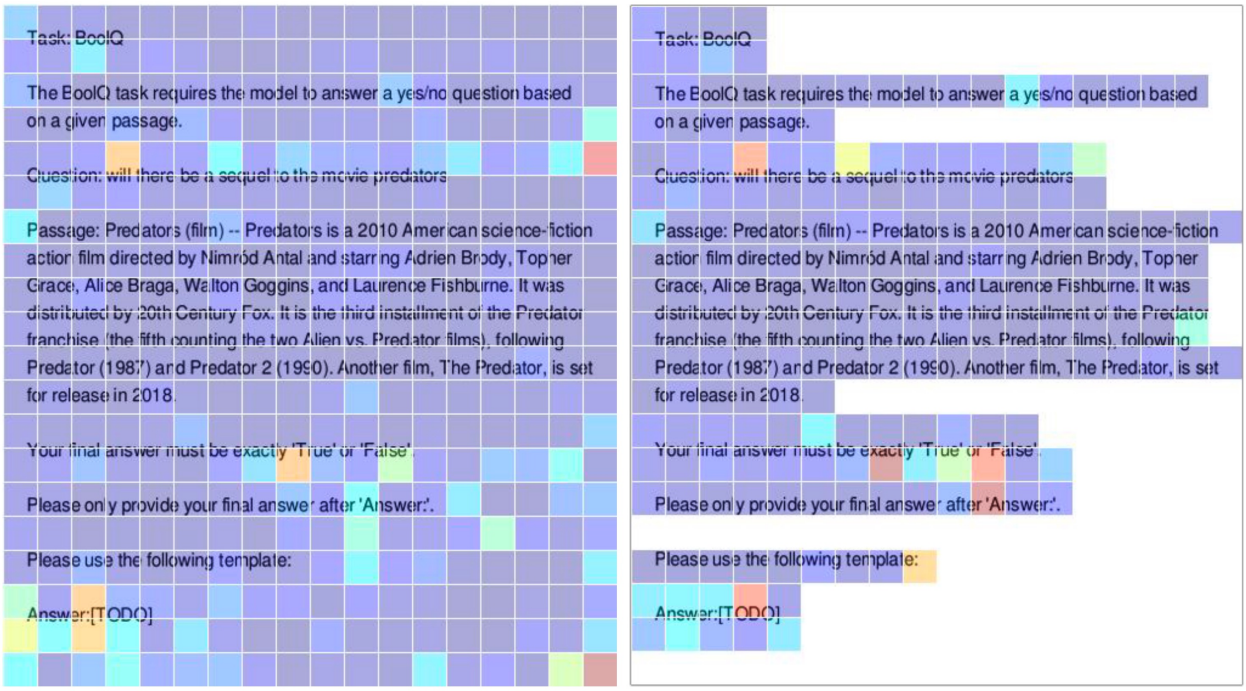


Figure 9: Last Layer Attention Heatmap on Qwen2VL-7B between PEAP (left) and PEAP-Fast (right).