Can Visual Encoder Learn to See Arrows?

Naoyuki Terashita[1*] Yusuke Tozaki[1,2†] Hideaki Omote[1,3†] Congkha Nguyen[1]
Ryosuke Nakamoto[1] Yuta Koreeda[1] Hiroaki Ozaki[1]
[1]Hitachi, Ltd. [2]Kyoto Sangyo University [3]Gifu University

Abstract

The diagram is a visual representation of a relationship illustrated with edges (lines or arrows), which is widely used in industrial and scientific communication. Although recognizing diagrams is essential for vision language models (VLMs) to comprehend domain-specific knowledge, recent studies reveal that many VLMs fail to identify edges in images. We hypothesize that these failures stem from an overreliance on textual and positional biases, preventing VLMs from learning explicit edge features. Based on this idea, we empirically investigate whether the image encoder in VLMs can learn edge representation through training on a diagram dataset in which edges are biased neither by textual nor positional information. To this end, we conduct contrastive learning on an artificially generated diagram–caption dataset to train an image encoder and evaluate its diagram-related features on three tasks: probing, image retrieval, and captioning. Our results show that the finetuned model outperforms pretrained CLIP in all tasks and surpasses zero-shot GPT-4o and LLaVA-Mistral in the captioning task. These findings confirm that eliminating textual and positional biases fosters accurate edge recognition in VLMs, offering a promising path for advancing diagram understanding.

1. Introduction

Diagram is a simplified and structured visual representation of relationships using shapes connected by edges (lines or arrows). Flowcharts, electronic circuits, and chemical structure diagrams are all examples of diagrams, and they play a major role in industrial and scientific communication. For a vision language model (VLM) to fully understand the context and knowledge in such domains, it is critical to accurately recognize diagram images.

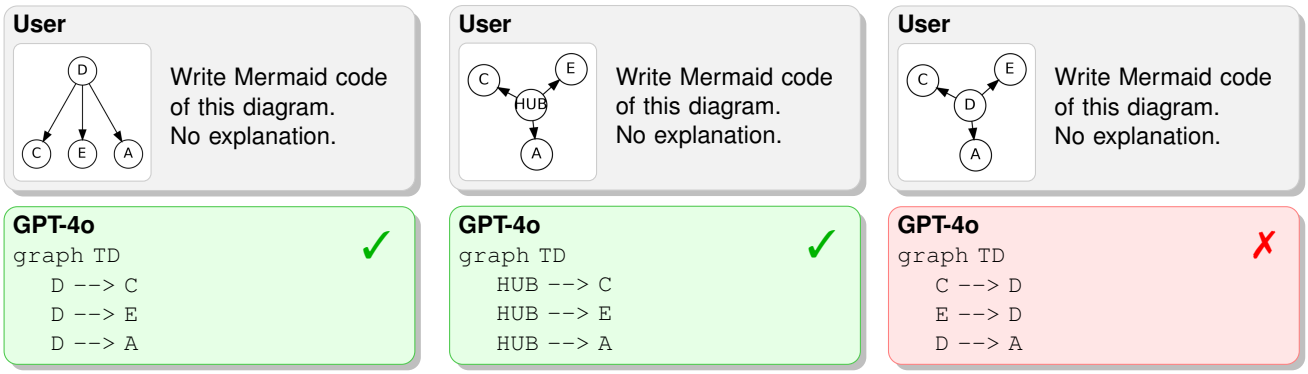
However, recent studies suggest that VLMs might not accurately recognize edges in diagram images. Yoshida et al. [24] have indicated that the feature representations of the CLIP [19] encoder, widely used in VLMs, may not contain sufficient information to classify the presence and direction of arrows in diagram images. From a similar motivation, Rahmanzadehgervi et al. [20] proposed a VLM benchmark that requires the model to answer questions about lines and shapes, demonstrating even large VLMs such as GPT-4o [17] and Gemini-1.5 Pro [23] can sometimes fail on even simple questions.

One reason VLMs often fail to recognize edges is that their visual training relies too heavily on positional or textual biases, hindering VLMs from learning edge features. This can be demonstrated through a simple experiment shown in Fig. 2; GPT-4o succeeds in describing a diagram when it can rely on common-sense biases derived from node positions (a) or textual cues (b), but fails when no such clues are available (c). Recent benchmarks on visual math problems [26] and flowchart VQA [22] have also shown that VLMs tend to rely on textual and positional biases.

Based on these observations, this study experimentally demonstrates that eliminating textual and positional biases during training enables visual models to learn edge features. To this end, we artificially generate a dataset of diagram images and text captions designed so that the presence and direction of edges cannot be inferred from text or position. We train CLIP, a common image encoder in VLMs, through contrastive learning on this dataset, then evaluate how well it captures edge information using three tasks: linear probing, image retrieval, and a newly proposed task called diagram captioning. In the linear probing, we classify edge existence and direction with acquired features, while the image retrieval evaluates the model's ability to find images representing the identical graph with possibly different visual layout. In our diagram captioning, we train a text decoder on the image encoder to predict the edge sets that appear in given diagram images.

Results from all three tasks show that our finetuned model substantially outperforms the original pretrained CLIP, indicating that our approach encourages acquiring edge representations invariant with textual and positional information. On our diagram captioning, the finetuned models also exceed the zero-shot performance of GPT-4o

*E-mail: naoyuki.terashita.sk@hitachi.com
†This work was done while the authors were interns at Hitachi, Ltd.



(a) Exploiting positional bias (b) Exploiting textual bias (c) No bias to exploit

Figure 1. Examples of diagram captioning by GPT-4o [17]: (a) inferring relationships based on conventional top-down hierarchies, (b) leveraging semantic relationships between node labels, and (c) struggling when neither positional nor textual biases are available. All results were produced by `gpt-4o-2024-08-06` with temperature 0.

and LLaVA-Mistral, highlighting the current limitation of large VLMs and the effectiveness of our approach.

2. Related Work

Recently, large vision language models (LVLMs) have achieved human-level performance on a variety of VQA tasks [6, 11, 17, 23], yet it has become clear that they rely heavily on textual content and positional layout to answer.

Chen et al.[5] demonstrated that GeminiPro [23] solves 42.90% of MMMU tasks [25] without any image inputs. This reveals that existing results of common VQA benchmarks might not reflect the actual vision capability of LVLMs. Further, in visual math benchmarks [4, 13, 25], removing problem texts regarding visual information substantially drops performance [26], indicating that many LVLMs merely rely on textual information. The limited capability in figure recognition is further illustrated by Rahmanzadehgervi et al. [20], who reveal that even state-of-the-art models like GPT-4V [16] fail at simple visual tasks such as counting overlapping shapes or determining line segment intersections. LVLMs also tend to rely on layout information. In a flowchart VQA task, simply flipping the layout vertically significantly degrades performance [22].

In this work, we show that by removing biases tied to text or positional information, VLMs can learn to recognize lines and arrows purely from visual inputs.

3. Learning with Debiased Diagram Dataset

As shown in Fig. 2, our approach consists of artificially generating diagram–text pairs that exclude text and positional biases (Sec. 3.1), followed by contrastive learning to finetune CLIP (Sec. 3.2).

3.1. Diagram Dataset without Positional and Textual Biases

We aim to build an image dataset with captions that eliminate biases arising from text and positional information. To have such a dataset with sufficient diversity, we generate diagram images and their Mermaid-style captions from randomly generated directed graphs. Note that we use "graph" to refer to the abstract mathematical structure and "diagram" to denote its visual representation.

Each sample in our dataset pairs an image with text representing a directed graph containing different numbers of alphabet-labeled nodes. The directed graphs are generated so that their edges are generated independently with a fixed probability for each pair among eight nodes, excluding self-loops and bidirectional edges. When a generated graph has more than one weakly connected component, we keep only the largest one, having graphs with different numbers of nodes (two to eight). For each graph, we draw a diagram image as in Fig. 2 whose node positions are laid out by the force-directed placement [8] with a random initial node layout. This initialization ensures that the same graph can produce diagram images with different layouts. The captions describe the generated directed graph in Mermaid format, where each line denotes a directed edge; e.g., `A --> B` indicates an edge from node "A" to node "B". We generate 100k image–caption pairs and use 10% of them as a test set.

3.2. Training Encoders via Contrastive Learning

We finetune pretrained CLIP models using contrastive learning on our artificially generated dataset. CLIP [19] is a dual-encoder architecture, comprising an image and text encoder, that learns joint representations from pairs of images and their captions. During training, CLIP minimizes a contrastive loss that brings the embeddings of matching image–text pairs closer while pushing apart the embeddings

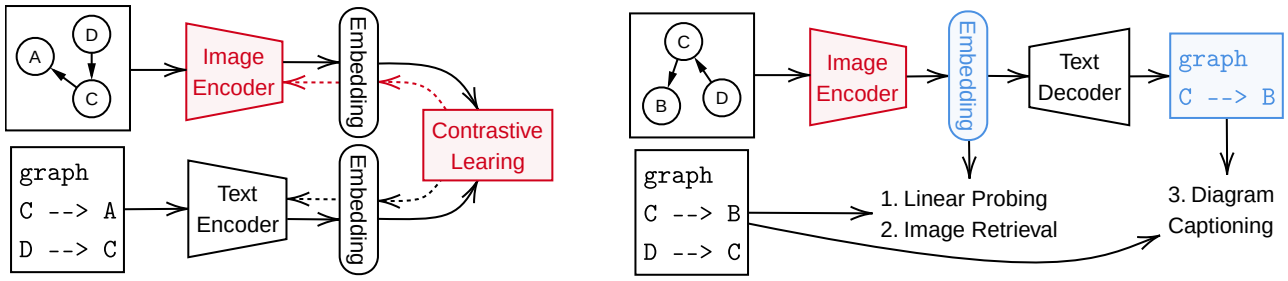


Figure 2. Overview of our approach: (a) training a CLIP model with diagram–caption pairs that eliminate positional and textual biases, and (b) evaluating the model on three tasks: linear probing, image retrieval, and diagram captioning.

of non-matching pairs. We specifically target CLIP for our approach because its image encoders serve as foundational components in numerous state-of-the-art large vision language models [2, 10, 11, 27].

For our experiments, we adopt two pretrained CLIP models with different image encoder sizes: CLIP-ViT-B/32 [14] and CLIP-ViT-L/14 [15]. CLIP-ViT-B/32 has 12 hidden layers and outputs 512-dimensional embeddings, whereas CLIP-ViT-L/14 has twice as many hidden layers and outputs 768-dimensional embeddings. We implement standard contrastive learning using our artificial dataset with a sufficiently large number of training steps.

4. Evaluation of Image Encoder

We evaluate the finetuned image encoder on three tasks that rely on diagram recognition: the linear probing (Sec. 4.1), image retrieval (Sec. 4.2), and diagram captioning (Sec. 4.3).

4.1. Linear Probing

Linear probing [1, 7] measures how well the extracted features encode information through classification tasks on features. In this study, we train and evaluate a simple logistic regression on top of the frozen image encoder to quantify the recognition capability of nodes and edges.

We define three binary classification tasks: *node existence*, *edge existence*, and *edge direction* classification. In node existence classification, for a given node label ("A" to "H"), we predict whether it appears in the diagram. Edge existence classification is a task to predict if an undirected edge exists between a given pair of nodes (ignoring direction). If either node is missing from the graph, that test sample is excluded to purely evaluate the edge recognition ability. Finally, in edge direction classification, we check whether a specific directed edge exists (e.g., from A to B). If no edge exists in either direction between the given two nodes, we skip that sample to solely evaluate the direction-related performance. We compute the accuracy for every possible label, node pair, or directed edge to report the average. For all tasks, we use balanced undersampling to ensure that the accuracies of all tasks can be compared to the chance rate (0.5).

To see the effect of our additional contrastive learning, we adopt pretrained CLIP-ViT-B/32 and CLIP-ViT-L/14 without additional contrastive training as baselines. As shown in Tab. 1, the pretrained baseline models perform poorly at edge direction classification (roughly at chance level), although they excel at text recognition (node labels). In contrast, both finetuned models show significant improvements from their baselines, especially in edge-direction classification (e.g., ViT-L/14 jumps from near-chance to 86% accuracy). These results indicate that removing textual and positional biases via contrastive learning lets the image encoder acquire edge-related features.

4.2. Image Retrieval

Our image retrieval task requires the model to retrieve all diagram images that represent the same directed graph as a given query image, which falls within the broader task category called content-based image retrieval [3, 21]. This task tests whether the learned features are invariant to node positions, as the query and target diagrams can have different layouts while representing identical graph structures.

We newly generate 1,000 query graphs using the same method in Sec. 3.1 ensuring each query graph appears in our test dataset (but possibly with a different layout). We encode both the query and all test images with the same image encoder, rank them by cosine similarity, and measure Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) up to the top 100 results.

As shown in Tab. 1, our finetuned ViT-B/32 and ViT-L/14 achieve MAP and MRR scores above 0.97, indicating that they successfully learn diagram features that are invariant to node positions. Fig. 3 shows examples of query images and retrieval results from finetuned and pretrained ViT-L/14. The pretrained image encoder mostly tends to focus on text and layout similarity, and it succeeds only when the

Table 1. Performance comparison of pretrained and finetuned CLIP models on diagram understanding tasks.

Method	Linear Probing (Mean Accuracy)			Image Retrieval	
	Node existence	Edge existence	Edge direction	MAP@100	MRR@100
Random	0.500	0.500	0.500	0.0004	0.001
Pretrained ViT-B/32	0.959	0.639	0.518	0.067	0.108
Pretrained ViT-L/14	0.999	0.725	0.509	0.131	0.170
Finetuned ViT-B/32	0.994	0.726	0.857	0.973	0.973
Finetuned ViT-L/14	**1.000**	**0.735**	**0.860**	**0.996**	**0.996**

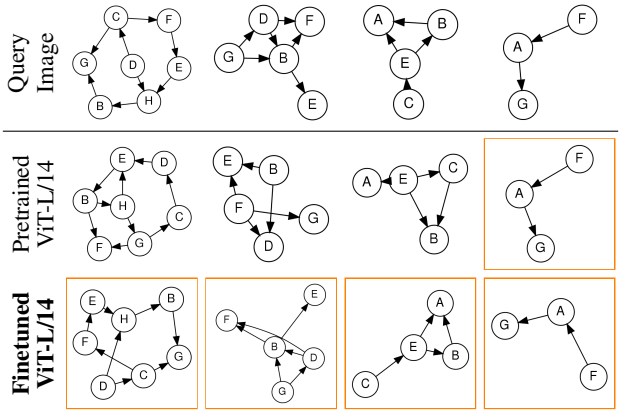


Figure 3. Examples of query images (top row) and the top retrieved images using the pretrained ViT-L/14 (middle row) and finetuned ViT-L/14 (bottom row). Images surrounded by orange lines represent true positives that share the same directed graph structures as the queries.

node layouts are extremely similar. By contrast, the finetuned models correctly retrieve matching graphs even if the node layouts differ significantly.

4.3. Diagram Captioning

This section proposes a new task called diagram captioning that requires a VLM to describe the edges in a given diagram image, accompanied by training a text decoder.

Our diagram captioning is a task to predict a Mermaid-style description of the diagram presented in an input image. Performance is measured by the micro F1-score of the predicted edge set, obtained by parsing edge descriptions in the predicted Mermaid text (e.g., `A --> B`).

To construct our VLM, we pair CLIP's image embeddings with GPT-2 [9, 18] as a text decoder. Our GPT-2 uses cross-attention on the image embeddings and previous tokens, predicting the next token probabilities. We train on our artificial image–caption dataset with cross-entropy loss, freezing the image encoder's weights, and select checkpoints based on validation loss from 1% of the training set. We compare the finetuned models with baselines that use the original CLIP encoders for image features, as well as with zero-shot inference from large VLMs, namely GPT-4o [17] (`gpt-4o-2024-08-06`) and LLaVA-Mistral [12] (which also uses CLIP-ViT-L/14). Zero-shot inference is prompted to generate a Mermaid-format caption describing the given diagram image.

Table 2. Diagram captioning performance comparison.

Method	F1-score
Llava-Mistral [12]	0.118
GPT-4o [17]	0.500
Pretrained ViT-B/32 (+GPT-2)	0.413
Pretrained ViT-L/14 (+GPT-2)	0.668
Finetuned ViT-B/32 (+GPT-2)	0.516
Finetuned ViT-L/14 (+GPT-2)	**0.966**

Tab. 2 shows that the finetuned ViT-L/14 encoder achieves an F1 of 0.966, outperforming pretrained ViT-L/14 and clearly beating GPT-4o with zero-shot. This indicates that the improved edge representation confirmed by the linear probing and image retrieval also benefits practical downstream tasks. The lower performance of the pretrained ViT-L/14 in both the diagram captioning and linear probing (Tab. 1) indicates that simply adapting the decoder is not enough; the bottleneck lies in the image encoder. GPT-4o and LLaVA-Mistral were shown to struggle with tasks that have no textual and positional cues to rely on, which is consistent with our findings in Fig. 2. Although a supervised instruction tuning on these models would likely improve performance and provide insightful results, we leave this for our future work. We also tested our models on diagram images whose graphs are non-isomorphic to any training sample (even as unlabeled and undirected). Despite a slight performance degradation, our finetuned models still significantly outperformed the baselines, showing their strong generalization to unseen graph structures.

5. Conclusion

We showed that removing textual and positional biases enables VLMs to learn edge recognition in diagrams. Using a synthetic dataset and contrastive learning on CLIP-based encoders, our finetuned models outperformed pretrained baselines across linear probing, image retrieval, and diagram captioning. This highlights the effectiveness of removing textual and positional biases for teaching VLMs to capture diagram structure.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. 3

[2] Adham Awadalla, Iris Gao, Jonathan Gardner, Jack Hessel, Younes Hanafy, Wenzheng Zhu, Karan Marathe, Yacine Bitton, Samir Gadre, Shixiang Sagawa, et al. OpenFlamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 3

[3] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *Eur. Conf. Comput. Vis.*, pages 584–599. Springer, 2014. 3

[4] Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P Xing, and Liang Lin. GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning. *arXiv preprint arXiv:2105.14517*, 2021. 2

[5] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Adv. Neural Inform. Process. Syst.*, 2024. 2

[6] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme, Sebastian Goodman, Xiao Wang, Yi Tay, Siamak Shakeri, Mostafa Dehghani, Daniel Salz, Mario Lučić, Michael Tschannen, Arsha Nagrani, Hexiang Hu, Mandar Joshi, Bo Pang, Ceslee Montgomery, Paulina Pietrzyk, Marvin Ritter, Alexander J. Piergiovanni, Matthias Minderer, Filip Pavetić, Austin Waters, Gang Li, Ibrahim Alabdulmohsin, Lucas Beyer, Julien Amelot, Kenton Lee, Andreas Steiner, Yang Li, Daniel Keysers, Anurag Arnab, Yuanzhong Xu, Keran Rong, Alexander Kolesnikov, Mojtaba Seyedhosseini, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI-X: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023. 2

[7] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Annu. Meet. Assoc. Comput. Linguist.*, pages 2126–2136, Melbourne, Australia, 2018. Association for Computational Linguistics. 3

[8] Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. 2

[9] HF Canonical Model Maintainers. `gpt2` (revision 909a290), URL: `https://huggingface.co/gpt2`, 2022. 4

[10] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 3

[11] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 2, 3

[12] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. `https://llava-vl.github.io/blog/2024-01-30-llava-next/`, 2024. 4

[13] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *Int. Conf. Learn. Represent.*, 2024. 2

[14] OpenAI. CLIP ViT-B/32. `https://huggingface.co/openai/clip-vit-base-patch32`, 2021. Hugging Face Model Hub. Accessed: 2025-04-12. 3

[15] OpenAI. CLIP ViT-L/14-336. `https://huggingface.co/openai/clip-vit-large-patch14-336`, 2021. Hugging Face Model Hub. Accessed: 2025-04-12. 3

[16] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[17] OpenAI. Hello, GPT-4o, 2023. URL: `https://openai.com/index/hello-gpt-4o/`, Accessed: 2025-02-14. 1, 2, 4

[18] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 4

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2

[20] Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In *ACCV*, pages 18–34, 2024. 1, 2

[21] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 806–813, 2014. 3

[22] Shubhankar Singh, Purvi Chaurasia, Yerram Varun, Pranshu Pandya, Vatsal Gupta, Vivek Gupta, and Dan Roth. FlowVQA: Mapping multimodal logic in visual question answering with flowcharts. In *Annu. Meet. Assoc. Comput. Linguist.*, pages 1330–1350, 2024. 1, 2

[23] Team Gemini, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 1, 2

[24] Haruto Yoshida, Keito Kudo, Yoichi Aoki, Ryota Tanaka, Itsumi Saito, Keisuke Sakaguchi, and Kentaro Inui. How well do vision models encode diagram attributes? In *the ACL 2024 Student Research Workshop*, 2024. 1

[25] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9556–9567, 2024. 2

[26] 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*, pages 169–186. Springer, 2024. 1, 2

[27] Deyao Zhu, Jiaming Chen, Xiaoqian Shen, Xiatian Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 3