Differences That Matter:
Auditing Models for Capability Gap Discovery and Rectification

Supplementary Material

Abstract

In the appendix, we provide additional information that could not be included in the main paper due to space limitations. The appendix is structured as follows:

A. Method and Implementation Details

A.1. Instruction Prompts in AuditDM

As mentioned above, the auditor $\mathcal{A}$ relies on three instruction prompts, p_c, p_e, and p_q, to guide the generation of image-regeneration captions $C = \mathcal{A}(I, p_c)$, image-editing commands $E = \mathcal{A}(I, p_e)$, and new questions $Q = \mathcal{A}(I', p_q)$. We provide the prompts here:

Prompt for image regeneration ($\mathbf{p_c}$):

```
You are given an image. Produce a
detailed, literal caption that would
allow a model to regenerate the image,
but also introduce small alterations to
certain visual attributes. Return a
single final caption describing the
modified version only.
```

Prompt for image editing ($\mathbf{p_e}$):

```
You are given an image. Generate a
single image-editing command that
describes how to modify the image.
The modification must remain plausible
in the real world. The command should
be specific, actionable, and unambiguous.
Return the editing command only.
```

Prompt for question generation ($\mathbf{p_q}$):

```
You are given an image. Generate a
single question that can be answered
solely based on its visible content.
Return the question only.
```

In addition, in the main paper, we compare our model to a baseline to evaluate the failure search success rate. The baseline uses the same system without fine-tuning, relying solely on prompt engineering to identify failure cases. The prompts used for the baseline are provided below:

Baseline prompt for image regeneration ($\mathbf{p_c^b}$):

```
You are given an image. Produce a
detailed, literal caption that would
allow a model to regenerate the image,
but introduce small changes that are
easy for models to get wrong. Return
a single caption describing the
modified version only.
```

Baseline prompt for image editing ($\mathbf{p_e^b}$):

```
You are given an image. Generate a
single image-editing command that
makes a realistic change but is
challenging for vision models. The
modification must remain plausible
in the real world. The command
should be specific, actionable, and
unambiguous. Return the editing
command only.
```

Baseline prompt for question generation ($\mathbf{p_q^b}$):

```
You are given an image. Generate a
single question answerable from its
visible content but challenging for
vision-language models. Return the
question only.
```

A.2. Auditor Training

For all experiments in this paper, we fine-tune a pretrained Gemma3-4B [25] as the auditor, and use FLUX.1-dev [12] for image generation, FLUX.1-Kontext-dev for image editing. For the reference MLLM, we consider two scenarios: (1) **Model comparison.** When training the auditor to compare the behavior of two models, we designate one as the target MLLM and the other as the reference MLLM, and fine-tune the auditor to generate image–question pairs that maximize the discrepancy between their outputs. (2) **Single-model analysis.** When training the auditor to probe the weaknesses and failure modes of a single target model, we take that model as the target MLLM, and construct a reference ensemble from PaliGemma2 [24], Gemma3 [25], and Qwen2.5-VL [2]. Note that the target model is not included in the ensemble.

Table 1. **PaliGemma2-3B training hyperparameters for each task.**

	Epochs	Batch size	Learning rate	Weight Decay	LLM dropout	Label smoothing
VQAv2	14	256	1×10^{-5}	1×10^{-7}	0.0	0.0
GQA	4	256	1×10^{-5}	1×10^{-7}	0.0	0.0
OK-VQA	10	256	5×10^{-6}	0.0	0.0	0.0
AI2D	16	256	1×10^{-5}	1×10^{-6}	0.0	0.0
DocVQA	10	256	1×10^{-5}	1×10^{-6}	0.0	0.0
ChartQA	40	256	1×10^{-5}	1×10^{-6}	0.1	0.2
RefCOCO	120	256	1×10^{-5}	0.0	0.0	0.3
COCOCap	8	256	1×10^{-5}	1×10^{-6}	0.0	0.0

During training, only the auditor is updated, while all other modules remain fixed. The auditor is fine-tuned for $1,000$ steps with AdamW [20], using an initial learning rate of 3×10^{-6}, a 10% warm-up, cosine learning rate decay to 1×10^{-6}, and a global batch size of 256.

A.3. PaliGemma2 Training

We provide additional experimental details for PaliGemma2 here. For all experiments involves PaliGemma2, we default to 448px^2 unless otherwise noted.

Benchmarks. Following PaliGemma2 [24], we evaluate on a range of academic benchmarks and fine-tune PaliGemma2-3B separately for each task. We consider four common tasks: general VQA, text-oriented VQA, image segmentation, and image captioning. For general VQA, we use VQAv2 [1], GQA [9], and OK-VQA [21]; for text-oriented VQA, we use AI2D [11], DocVQA [23], and ChartQA [22]; for segmentation and captioning, we use RefCOCO [10] and COCOCap [17], respectively. Following PaliGemma [3], the models are trained on the corresponding training splits and evaluated on the validation or test splits, depending on their availability. For the VQA tasks, we report exact-match accuracy; for RefCOCO, we report mean Intersection-over-Union (mIoU); and for COCOCap, we report CIDEr score.

Training configurations. For each task, we first fine-tune an auditor on a random subset of the training data using the hyperparameters in Sec. A.2. Specifically, since each model is trained with a global batch size of 256 for $1,000$ steps, the auditor is exposed to at most 256K images during fine-tuning. To construct the training pool for the auditor, we first filter the training set by image resolution and then randomly select 256K images to form the image pool for auditor training.

After training the auditor, we use it to generate new image–question pairs from the training data, producing one new pair for each original example. We then mix these generated pairs with the original training set to further fine-tune the PaliGemma2-3B model at 448px^2 resolution. Following PaliGemma [3], all models are trained until convergence. We provide hyperparameters for each task in Table 1

A.4. Gemma3 Training

To further demonstrate the effectiveness of AuditDM for model failure rectification and model improvement, we also apply AuditDM to Gemma3-4B and evaluate it on eight widely used, task-diverse multimodal benchmarks. We provide the experimental details below.

Training data. AuditDM requires only images as input. Therefore, we collect data from DataComp-1B [8] and a mixture of academically annotated datasets, following LLaVA-OneVision [14] and FineVision [26]. In addition to standard image filters such as resolution and caption alignment, we also aim to ensure sufficient diversity and complexity in the images. To this end, we follow DenseWorld [15] and generate dense understandings of all filtered images, which allows us to further select images based on factors such as the number of objects, the diversity of object categories and relationships, overall scene richness, and the presence of fine-grained visual details. This additional filtering step removes overly simple, overly complex, or low-content images and ensures that the final training set contains images that can give rise to challenging yet meaningful probing questions or counterfactual images. As a result, we obtain a pool of 1.3M images for training, including 736K from DataComp-1B and 600K from the remaining academically annotated datasets.

Benchmarks. We use the VLMEvalKit framework [6] to evaluate all models on a diverse suite of 8 benchmarks, including MMBench-v1.1 [18], MMTBench [28], Seed-Bench-IMG [13], MME [7], MMMU [29], MMStar [5], RealWorldQA [27], and POPE [16].

Training details. After constructing the training image set, we train the auditor for 1,000 steps on these images to identify weaknesses of the Gemma3-4B model, saving checkpoints every 250 steps. The resulting auditors (checkpoints saved at different steps) are then used to generate new image–question pairs. We aggregate and deduplicate these images, and then use them to further fine-tune the Gemma3-4B model. We subsequently train new auditors on the newly fine-tuned Gemma3-4B model to analyze its remaining weaknesses. This sequence, consisting of (i) training the auditor with the latest MLLM, (ii) regenerating data from the unlabeled pool using the trained auditor, and (iii) fine-tuning the MLLM with the new data, constitutes one refinement iteration, and we perform two such iterations. While we observe further improvements with additional iterations in preliminary experiments, we restrict ourselves to two iterations to keep the compute cost within a reasonable budget (see Sec. B.3). In each iteration, the auditor is trained following the hyperparameters in Sec. A.2. Then, given the newly generated samples, we fine-tune Gemma3-4B with a global batch size of 512 and a learning rate of 2×10^{-6} for one epoch.

Table 2. **Training Gemma3-4B on the same images without AuditDM.**

Model	MMBench-v1.1	MMTBench	Seed-Bench-IMG	MME	MMMU	MMStar	RealWorldQA	POPE
Gemma3-4B	67.6	53.2	65.7	1376.0	39.6	46.1	54.5	85.1
Gemma3-4B (with the same images)	69.6 (+2.0)	54.8 (+1.6)	66.9 (+1.2)	1321.6 (-54.4)	40.5 (+0.9)	45.6 (-0.5)	56.3 (+1.8)	84.2 (-0.9)
Gemma3-4B + AuditDM (Ours)	75.0 (+7.4)	58.9 (+5.7)	72.9 (+7.2)	1450.3 (+74.3)	45.2 (+5.6)	52.4 (+6.3)	61.4 (+6.9)	85.5 (+0.4)

Table 3. **Empirical validation of assumptions.**

Type	Percentage (%)
Target failures	81.3
Ambiguous questions	11.5
Unanswerable questions	7.2

Table 4. **Training data size performance comparison.**

	ori	20%	60%	100%
Avg. Score	58.83	62.54	63.70	64.47

B. Additional Experimental Results

B.1. Training Gemma3-4B without AuditDM

In the main paper, we demonstrate that AuditDM significantly improves the performance of Gemma3-4B on various benchmarks. To further verify that these gains come from AuditDM rather than from additional fine-tuning on the selected images, we also fine-tune Gemma3-4B on the same images under the same settings. Since some of the images do not have original questions, we use a pre-trained Gemma3-4B model (the same initial model used by AuditDM) to generate new questions. The results are shown in Table 2. We observe that fine-tuning on the same data without AuditDM yields only marginal improvements on a few benchmarks, whereas our method achieves substantial gains across all benchmarks.

B.2. Empirical Validation of Assumptions

As discussed earlier, to ensure that the observed failures are attributable to the target model rather than artifacts of the auditor (e.g., generating meaningless questions), the diffusion model (e.g., producing inaccurate or unrealistic images), or the ensemble (e.g., providing incorrect answers), we rely on two assumptions: (1) *Answerable instances:* when the ensemble models agree on an answer, the question–image pair is likely to be meaningful and answerable. (2) *Rarity of target correctness:* it is relatively rare for target model to be uniquely correct while all models in the ensemble are wrong.

In this section, we show that the ensemble indeed identifies genuine failures of the target model, i.e., (1) the question–image pairs are meaningful and answerable, and (2) they expose weaknesses of the target model. Specifically, we use the auditor to generate 1,000 new question–image pairs on which the target model and the ensemble disagree. We then manually verify the correctness of their answers. We report (i) the percentage of samples that truly expose weaknesses of the target model (*i.e.*, *target failures*), (ii) the percentage of samples where the target model is still accurate despite disagreeing with the ensemble (*i.e.*, *ambiguous questions*, where both answers can be considered correct), and (iii) the percentage of samples where the question itself is not answerable from the image (*i.e.*, *unanswerable questions*). The results are provided in Table 3. We can see that 81.3% of the samples indeed expose genuine weaknesses of the target model, while 11.5% are acceptable or ambiguous (i.e., both target model and ensemble provide correct/acceptable yet differing answers). Only 7.2% are noise (shared failures or unanswerable). Thus, 92.8% of the samples provide valid, high-quality training signals with correct labels. This 7.2% noise is minimal and well-tolerated, as evidenced by significant gains across all benchmarks. In addition, we expect a higher percentage of target failures when using a more powerful ensemble.

B.3. Computational Complexity

To improve a target model such as Gemma3-4B, AuditDM consists of three steps: (1) training an auditor model, (2) generating large-scale data with the trained auditor, and (3) fine-tuning the target model. In our experiments with Gemma3-4B, the first stage requires 29 hours on 16 H100 GPUs for 1,000 training steps, and the second stage requires 63 hours on 16 H100 GPUs to generate all images using Ray implementation. However, we emphasize that using LLMs or diffusion models for data generation is already a widely adopted but time-consuming practice [4, 19]. As all these methods depend on inference with an LLM or diffusion model for data generation, our overall time cost is comparable to theirs.

Data scale. Our training set comprised 1.3M images, each paired with five questions, totaling 6.5M samples. As shown in Table 4, we observed consistent performance gains even when using only 20% of the dataset, which required approximately 12 hours of preparation. These low-resource experiments demonstrate that our approach yields significant results even under constrained data and hardware environments.

C. Additional Qualitative Examples

We provide additional examples of the failures identified by our model in PaliGemma2 in Fig. 1, Fig. 2, and Fig. 3.

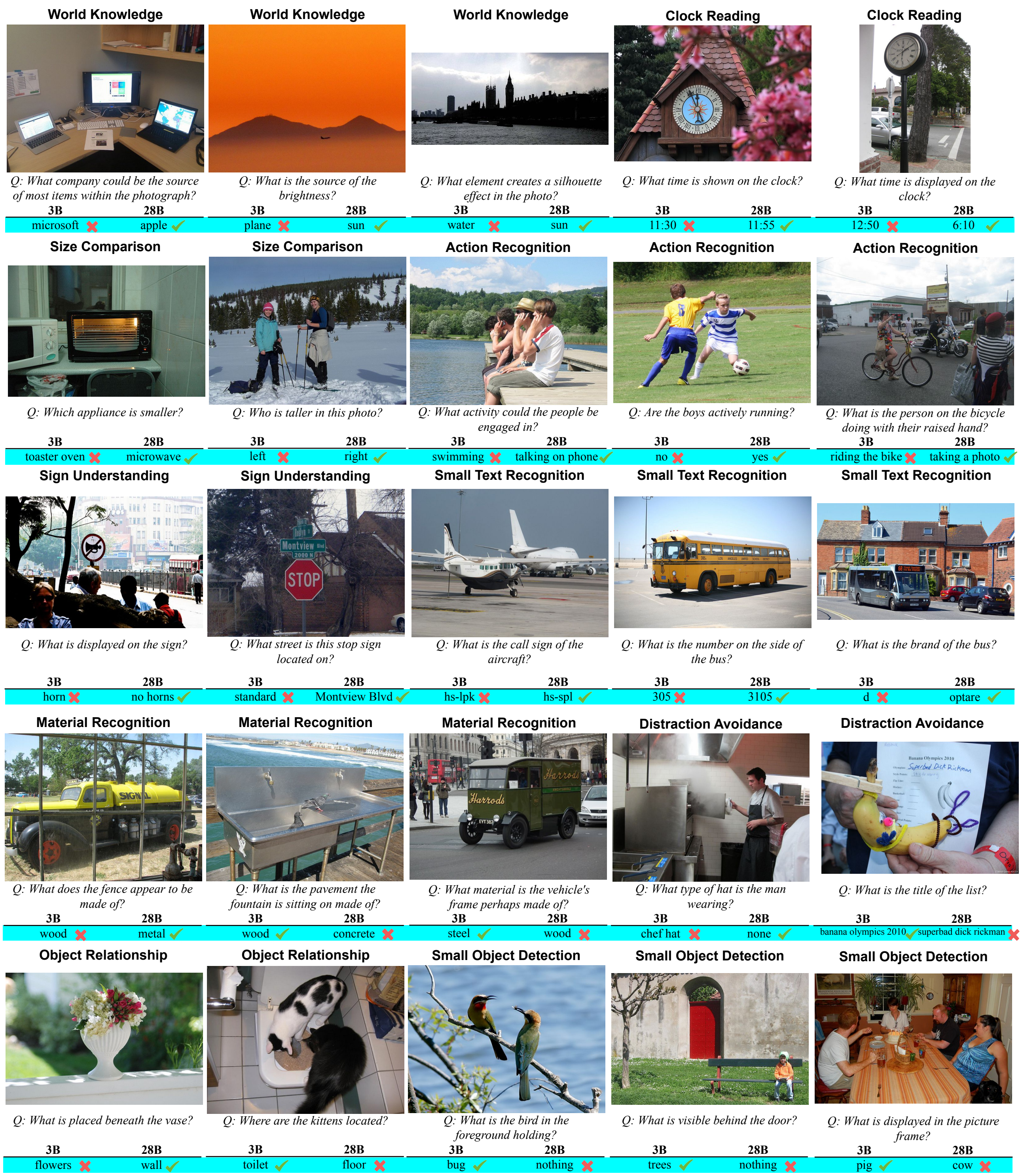

Figure 1. **Generated examples for each failure category.** To better demonstrate the effectiveness, we focus on examples with *original images* and *generated questions*.

Specifically, Fig. 1 and Fig. 2 show additional examples in their original image ratios for each weakness category. Note that, to ensure a fair comparison between the 3B and 28B models while eliminating potential influences of image

Figure 2. **Generated examples for each failure category.** To better demonstrate the effectiveness, we focus on examples with *original images* and *generated questions*.

style, we generate probing questions on the original images without altering their visual appearance. These results further confirm that the identified failures stem from intrinsic model limitations rather than superficial visual differences.

Fig. 3 presents additional examples involving visual changes. Our model efficiently discovers small image modifications that significantly alter the model's predictions. This highlights the model's sensitivity to fine-grained, task-irrelevant visual cues and reveals a lack of robustness in its visual reasoning process. At the same time, these examples provide valuable insights into the visual factors that influence model behavior, making its weaknesses and vulnerabilities more interpretable.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In Proceedings of the IEEE international conference on computer vision, pages 2425–2433, 2015. 2

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923, 2025. 1

[3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. arXiv preprint arXiv:2407.07726, 2024. 2

[4] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In European Conference on Computer Vision, pages 370–387. Springer, 2024. 3

[5] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? Advances in Neural Information Processing Systems, 37:27056–27087, 2024. 2

[6] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In Proceedings of the 32nd ACM International Conference on Multimedia, pages 11198–11201, 2024. 2

[7] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin,

Figure 3. **Additional examples of fine-grained visual cues that are irrelevant to the task or question but still alter the model's predictions.** We target PaliGemma2-28B (448px^2) and showcase small modifications that fool the 28B model but not the 3B model, highlighting the effectiveness of our method in finding failures even in very powerful models. For each example, we show the original image (left), the modified image (right), and the corresponding answers.

Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394, 2023. 2

[8] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. Advances in Neural Information Processing Systems, 36:27092–27112, 2023. 2

[9] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6700–6709, 2019. 2

[10] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 787–798, Doha, Qatar, 2014. Association for Computational Linguistics. 2

[11] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In European conference on computer vision, pages 235–251. Springer, 2016. 2

[12] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. 1

[13] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. arXiv preprint arXiv:2307.16125, 2023. 2

[14] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326, 2024. 2

[15] Xiangtai Li, Tao Zhang, Yanwei Li, Haobo Yuan, Shihao Chen, Yikang Zhou, Jiahao Meng, Yueyi Sun, Shilin Xu, Lu Qi, et al. Denseworld-1m: Towards detailed dense grounded caption in the real world. arXiv preprint arXiv:2506.24102, 2025. 2

[16] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355, 2023. 2

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 2

[18] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In European conference on computer vision, pages 216–233. Springer, 2024. 2

[19] Zheng Liu, Hao Liang, Bozhou Li, Tianyi Bai, Wentao Xiong, Chong Chen, Conghui He, Wentao Zhang, and

Bin Cui. Synthvlm: High-efficiency and high-quality synthetic data for vision language models. arXiv preprint arXiv:2407.20756, 2024. 3

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017. 2

[21] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In Proceedings of the IEEE/cvf conference on computer vision and pattern recognition, pages 3195–3204, 2019. 2

[22] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In Findings of the association for computational linguistics: ACL 2022, pages 2263–2279, 2022. 2

[23] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 2200–2209, 2021. 2

[24] Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, et al. Paligemma 2: A family of versatile vlms for transfer. arXiv preprint arXiv:2412.03555, 2024. 1, 2

[25] 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. 1

[26] Luis Wiedmann, Orr Zohar, Amir Mahla, Xiaohan Wang, Rui Li, Thibaud Frere, Leandro von Werra, Aritra Roy Gosthipaty, and Andrés Marafioti. Finevision: Open data is all you need. arXiv preprint arXiv:2510.17269, 2025. 2

[27] x.ai. Grok-1.5 vision preview. https://x.ai, 2024. Accessed: 2025-11-19. 2

[28] Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, et al. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi. arXiv preprint arXiv:2404.16006, 2024. 2

[29] 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9556–9567, 2024. 2