Boosting Reasoning in Large Multimodal Models via Activation Replay

Supplementary Material

A. Related Works

Activation Engineering has recently emerged as an effective approach to improve and leverage the transparency of large pretrained models [2, 22]. By simply manipulating internal representations [1, 9, 10, 13], these models can be directed towards desired behaviors, like unlearning [7, 11] or instruction following [12]. Some recent studies have discovered that this approach could mitigate overthinking [4] or encourage self-correction [21]. We study how post-train affects *contextual* activations, replaying them for improving reasoning, which methodologically follows [14, 18].

B. Inference Latency

Latency. Activation Replay incurs added latency from two sources: an extra prefill pass with the base LMM and gradient updates on the learnable tokens. We report a time comparison result in table below. Both costs are manageable in practice, with either low-resolution visual inputs or fewer learnable tokens (*i.e.*, sharing one learnable among adjacent tokens). Crucially, gains are preserved with these efficient settings. In terms of memory, the replay requires about extra 2× memory of standard inference, which can be reduced to about 1.5× or less with mentioned strategies.

Table 1. Inference Latency.

	Ratio	Time Per Sample (s)				ME
		prefill	*grad*	*infer*	*latency*$_\downarrow$	
MM-Eureka-7B	N/A	N/A	N/A	12.87	N/A	45.1
+ *ours*	100%	0.21	4.43	11.85	6.64	47.7
+ *low-res prefill*	50%	0.15	4.01	15.18	5.16	48.3
+ *fewer learnables*	25%	0.15	2.82	13.76	**2.97**	47.5

Table 2. Additional Time Costs with Activation Replay.

	MMR1-Math		DeepEyes	
	ME	MN	H4	H8
Relative Time Cost	0.77×	1.42×	1.05×	0.41×

Replay for Agentic/Video Reasoning. For visual agentic data that involve searching on high-resolution images and video data that need reasoning across frames, learning tokens are no longer efficient solutions. For them, we use *direct intervention* instead.

Details. Note that Activation Replay brings addtional compute during test time, we provide a comparison over two tasks to clarify this, including MMR1-Math-v0 [5] over two datasets (MathVerse and MathVision [16, 19]) as mathematical testbeds, and DeepEyes [20] over two subsets of HRBench [17] as agentic testbeds. Note that agentic reasoning typically involves much longer reasoning traces than mathematical tasks. We also point out that our design only involve modulations of input contexts, while decoding is the same as baselines. For timecosts, baselines involve prefilling and decoding, while Activation Replay involves prefilling, decoding and test-time input context manipulations.

Discussions. We find that the overall elapsed time for learnable token manipulations is around 0.15×, especially considering those reasoning LMMs that output long chain-of-thought traces, the relatively elapsed time could be shorter. We observe in some cases Activation Replay is even comparably faster in overall (*e.g.*, DeepEyes on HRBench 8K, as in Table 2). For o3-like agentic LMMs characterized by multi-turn visual searching, our approach achieves slightly better performance (75.9 *v.s.* 75.4) with less turns.

Table 3. More Pass@K results. In this table, MV and MA are short for MathVision$_{mini}$ and MathVista, respectively.

Model	MV		MA	
	@8	@16	@2	@4
MMR1-Math [5]	62.8	71.0	77.0	81.3
+ *replay*	65.1 $_{2.3\uparrow}$	72.4 $_{1.4\uparrow}$	79.1 $_{2.1\uparrow}$	82.3 $_{1.0\uparrow}$
MM-Eureka [8]	57.2	63.8	77.3	80.3
+ *replay*	59.5 $_{2.3\uparrow}$	67.1 $_{3.3\uparrow}$	77.3 $_{0.0\uparrow}$	81.0 $_{0.7\uparrow}$
VL-Rethinker [15]	67.8	76.9	77.9	83.6
+ *replay*	68.8 $_{1.0\uparrow}$	78.6 $_{1.7\uparrow}$	77.4 $_{0.5\downarrow}$	83.8 $_{0.2\uparrow}$
VLAA-Thinker [3]	72.7	84.5	75.8	82.9
+ *replay*	74.8 $_{2.1\uparrow}$	89.5 $_{5.0\uparrow}$	77.2 $_{1.3\uparrow}$	83.6 $_{0.7\uparrow}$

Table 4. Activation Replay with Low (L) *v.s.* High-Entropy Activations (H).

Model		MN	LV
MM-Eureka	B	25.5	35.6
	R	30.6	49.2
	L	31.5 $_{0.9\uparrow}$	51.0 $_{1.8\uparrow}$
	H	29.9 $_{0.7\downarrow}$	49.0 $_{0.2\downarrow}$

C. Pass@K

We provide details and more Pass@K results here, as mentioned in our manuscript.

Table 5. Activation Replay with Intervention (V) *v.s.* Leaarnable Token Manipulation (M).

Model		ME	MN	LV
MM-Eureka	B	41.1	25.5	35.6
	R	45.1	30.6	49.2
	V	45.1 $_{0.0\uparrow}$	31.6 $_{1.0\uparrow}$	50.5 $_{1.3\uparrow}$
	M	47.7 $_{2.6\uparrow}$	31.5 $_{0.9\uparrow}$	51.0 $_{1.8\uparrow}$
VL-Rethinker	B	41.1	25.5	35.6
	R	47.0	30.3	46.1
	V	47.5 $_{0.5\uparrow}$	33.5 $_{3.2\uparrow}$	47.6 $_{1.5\uparrow}$
	M	49.2 $_{2.2\uparrow}$	33.2 $_{2.9\uparrow}$	49.7 $_{3.8\uparrow}$

Table 6. Activation Replay with Static (S) *v.s.* Dynamic Thresholding (D, *ours*).

Model		ME	MN	LV
MM-Eureka	B	41.1	25.5	35.6
	R	45.1	30.6	49.2
	S	46.1 $_{1.0\uparrow}$	32.9 $_{2.3\uparrow}$	47.9 $_{1.3\downarrow}$
	D	47.7 $_{2.6\uparrow}$	31.5 $_{0.9\uparrow}$	51.0 $_{1.8\uparrow}$
VL-Rethinker	B	41.1	25.5	35.6
	R	47.0	30.3	46.1
	S	45.7 $_{1.3\downarrow}$	32.2 $_{1.9\uparrow}$	48.5 $_{2.4\uparrow}$
	D	49.2 $_{2.2\uparrow}$	33.2 $_{2.9\uparrow}$	49.7 $_{3.8\uparrow}$

Detail. For MathVision$_{mini}$ [16], we present Pass@8 and Pass@16 and for MathVista [6], we compare Pass@2 and Pass@4 instead, due to the resource limitations. We set temperature to 1.0 for all RLVR LMMs to sample responses.
Result. The result is provided in Table 3. Apart from results presented in our manuscript, we present results of VLAA-Thinker [3] here. The performance gains from Activation Replay mostly hold.

D. Comparisons

D.1. Replay with High-Entropy Activations

We apply the proposed learnable token manipulation strategy to regulate high-entropy activations instead. The results are consistent with that of intervention study from Table **??** (manuscript). As presented in Table 4, replaying high-entropy activations, the strategy brings performance drop instead.

D.2. Intervention

We collect results of two replay strategies, namely direct intervention and learnable token manipulation, and present them in Table 5, In some cases, direct intervention are comparable to the learnable token manipulation (*e.g.*, MathVision). Overall, we find that learnable token manipulation brings better performance gains.

D.3. Thresholding

As mentioned in manuscript, we try a static thresholding to differentiate low-entropy activations from high-entropy ones. The static threshold is pre-computed by a validation set. As presented in Table 6, although with more test-time compute, we find that the static strategy brings comparable performance gains (sometimes even performance drop), as compared to results from dynamic strategy.

E. Case Study

E.1. Math

We show a case study when applying Activation Replay to MM-Eureka-Qwen-32B [8] in Figure 1. Notice that despite both inferences start with apply Rule 4 mistakenly. While with re-evaluating, the RLVR LMM again apply another wrong rule (Rule 2), while with Activation Replay, the LMM successfully applies Rule 1, and answer correctly.

E.2. Agent

We also show two case studies to present how Activation Replay affects o3-like agentic reasoning [20].
Localisation. One type of error correction is, RLVR LMM locates wrongly with tools. As in Figure 2, RLVR LMM falsely pinpoints the air conditioner outdoor unit on the red shed on the *right*, leading to the final false recognition ("GREE"). In comparison, Activation Replay correctly pinpoints and zoom in the region of "air conditioner on the left" and correctly recognizes the region.
Recognition. Another type is recognition-level error with correct tool call. As given in Figure 3, despite RLVR LMM zooms in the correct region, the recognition is false ("MILK EGG"), while with Activation Replay, recognition is correct ("Trash Box").

E.3. Perturbation Study

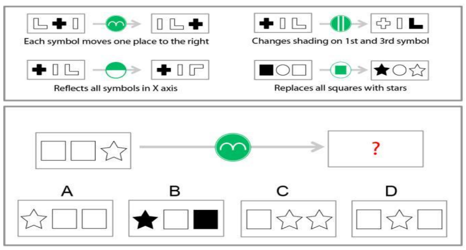
As mentioned in our manuscript, we also provide the perplexity shifts of incorrect responses in Figure 4. Note that as KL divergence of low-entropy logits increase, the perplexity of incorrect responses decreases, which is opposite to that observed for correct responses. Since Activation Replay improves Pass@1 in multiple tasks, we conjecture that this might indicate a potential cause for why our approach take effects. We leave this for future work.

References

[1] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37: 136037–136083, 2024. 1

[2] Lukasz Bartoszcze, Sarthak Munshi, Bryan Sukidi, Jennifer Yen, Zejia Yang, David Williams-King, Linh Le, Kosi

Question



What diagram replaces the question mark? Select from A, B, C, and D. (A) (B) (C) (D)

MM-Eureka-Qwen-32B

To solve this problem, we need to apply the rules given in the top section of the image to the sequence in the bottom section. Let's break it down step by step:

1. **Understanding the rules:**
 - **Rule 1:** Each symbol moves one place to the right.
 - **Rule 2:**Changes shading on 1st and 3rd symbol.
 - **Rule 3:**Reflects all symbols in X axis.
 - **Rule 4:**Replaced all squares with stars.

2. **Analyzing the given sequence:**
 - The sequence starts with: $ \square \square \star $.
 - The green circle ( ) next to the sequence indicate that we need to apply one of the rules.

3. **Applying the rules:**
 - The green circle ( ) suggests we should apply **Rule 4:** "Replaced alls with stars."
 - According to Rule 4, we need to replace all square ($ \square $) with stars ($ \star $).

4. **Applying Rule 4:**
 - The original sequence is: $ \square \square \star $.
 - Replaced alls with stars: $ \star \star \star $.

5. **Matching with the options:**
 - None of the options directly match $ \star \star \star $, but we need to re-evaluate the rules and the sequence again.

6. **Re-evaluting the rules:**
 - The green circle ( ) might suggest a different rule. Let's recheck:
 - The green circle ( ) could also suggest ****Rule 2**:** "Changes shading on 1st and 3rd symbol."
 - According to Rule 2, we need to change the shading on the 1st and 3rd symbol.

7. **Applying **Rule 2**:**
 - The original sequence is: $ \square \square \star $.
 - Change shading on 1st and 3rd symbol: $ \star \square \star $.

8. **Matching with the options:**
 - The sequence $ \star \square \star $ matches with option **C**.

<answer> The answer is $ \boxed{**C**} $. </answer>

+ Activation Replay (ours)

To solve this problem, we need to apply the rules given in the top section of the image to the sequence in the bottom section. Let's break it down step by step:

1. **Understanding the rules:**
 - **Rule 1:** Each symbol moves one place to the right.
 - **Rule 2:**Changes shading on 1st and 3rd symbol.
 - **Rule 3:**Reflects all symbols in X axis.
 - **Rule 4:**Replaced all squares with stars.

2. **Analyzing the given sequence:**
 - The sequence starts with: $ \square \square \star $.
 - The green circle ( ) indicates that we need to apply one of the rules.

3. **Applying the rules:**
 - The green circle ( ) suggests we should apply **Rule 4:** "Replaced alls with stars."
 - According to Rule 4, we need to replace all square ($ \square $) with stars ($ \star $).

4. **Applying Rule 4:**
 - The original sequence is: $ \square \square \star $.
 - Replaced alls with stars: $ \star \star \star $.

5. **Matching with the options:**
 - None of the options directly match $ \star \star \star $, but we need to consider the transformation rules again.
 - Let's re-evaluate the rules and the sequence:
 - The sequence $ \square \square \star $ might need to be transformed using **Rule 1**.

6. **Applying **Rule 1**:**
 - Rule 1: Each symbol moves one place to the right.
 - Moving each symbol one place to the right: $ \star \square \square $.

7. **Re-evaluting the options:**
 - The options are:
 - A: $ \star \square \square $.
 - B: $ \star \square \blacksquare $.
 - C: $ \square \star \star $.
 - D: $ \square \star \square $.

8. **Final match:**
 - The sequence $ \star \square \square $ matches with option A.

<answer> The answer is $ \boxed{A} $. </answer>

Figure 1. Case Study on Mathematical Reasoning.

Asuzu, and Carsten Maple. Representation engineering for large-language models: Survey and research challenges. *arXiv preprint arXiv:2502.17601*, 2025. 1

[3] Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models, 2025. 1, 2

[4] Yao Huang, Huanran Chen, Shouwei Ruan, Yichi Zhang, Xingxing Wei, and Yinpeng Dong. Mitigating overthinking in large reasoning models via manifold steering. *ArXiv*, abs/2505.22411, 2025. 1

[5] Sicong Leng, Jing Wang, Jiaxi Li, Hao Zhang, Zhiqiang Hu, Boqiang Zhang, Yuming Jiang, Hang Zhang, Xin Li, Lidong Bing, et al. Mmr1: Enhancing multimodal reasoning with variance-aware sampling and open resources. *arXiv preprint arXiv:2509.21268*, 2025. 1

[6] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel

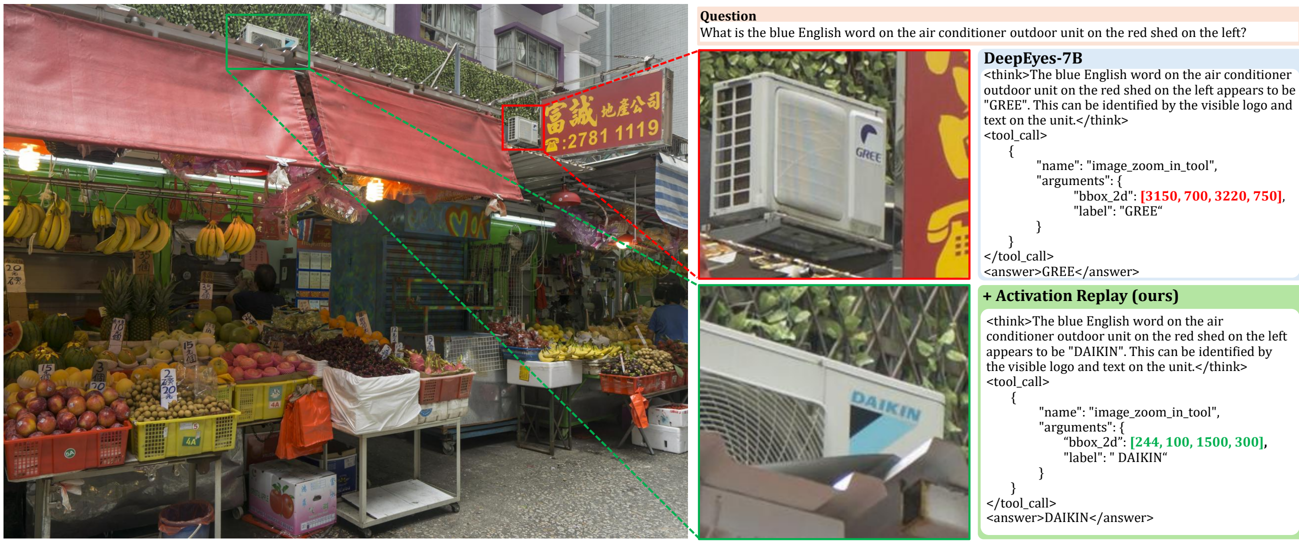


Figure 2. **False Tool Call**. Case Study of Multi-Turn o3-Like Agent [20].

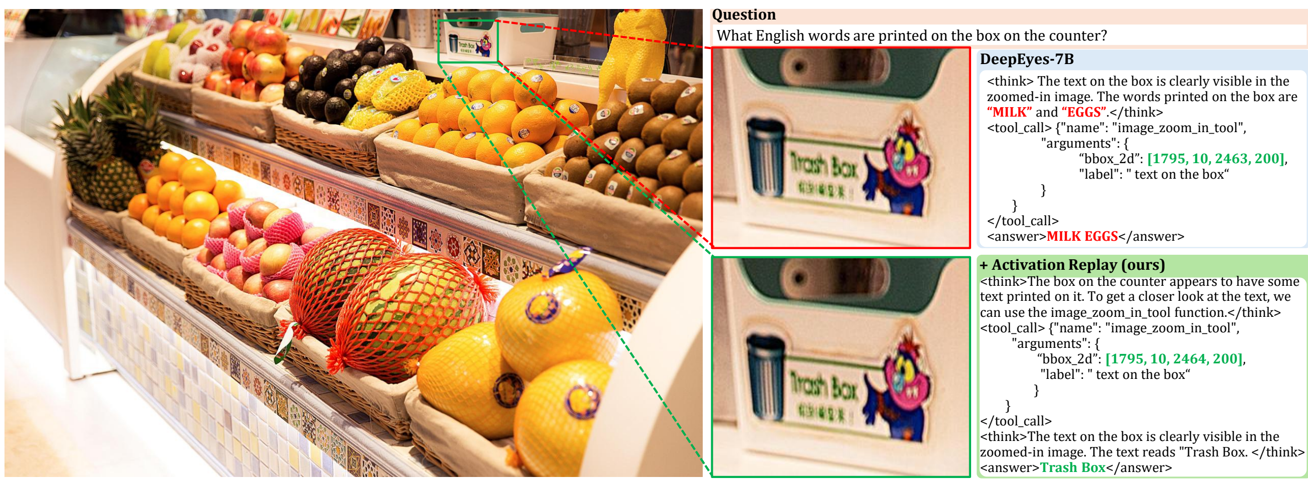


Figure 3. **False Recognition**. Case Study of Multi-Turn o3-Like Agent [20].

Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 2

[7] Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *ArXiv*, abs/2402.16835, 2024. 1

[8] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, et al. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *CoRR*, 2025. 1, 2

[9] Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, E. Ponti, and Shay B. Cohen. Spectral editing of activations for large language model alignment. *ArXiv*, abs/2405.09719, 2024. 1

[10] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, 2024. 1

[11] William F. Shen, Xinchi Qiu, Meghdad Kurmanji, Alexandru Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas Donald Lane. Lunar: Llm unlearning via neural activation redirection. *ArXiv*, abs/2502.07218, 2025. 1

[12] Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. *arXiv preprint arXiv:2410.12877*, 2024. 1

[13] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David S. Udell, Juan J. Vazquez, Ulisse Mini, and Monte Stuart MacDiarmid. Steering language models with activation engineering. 2023. 1

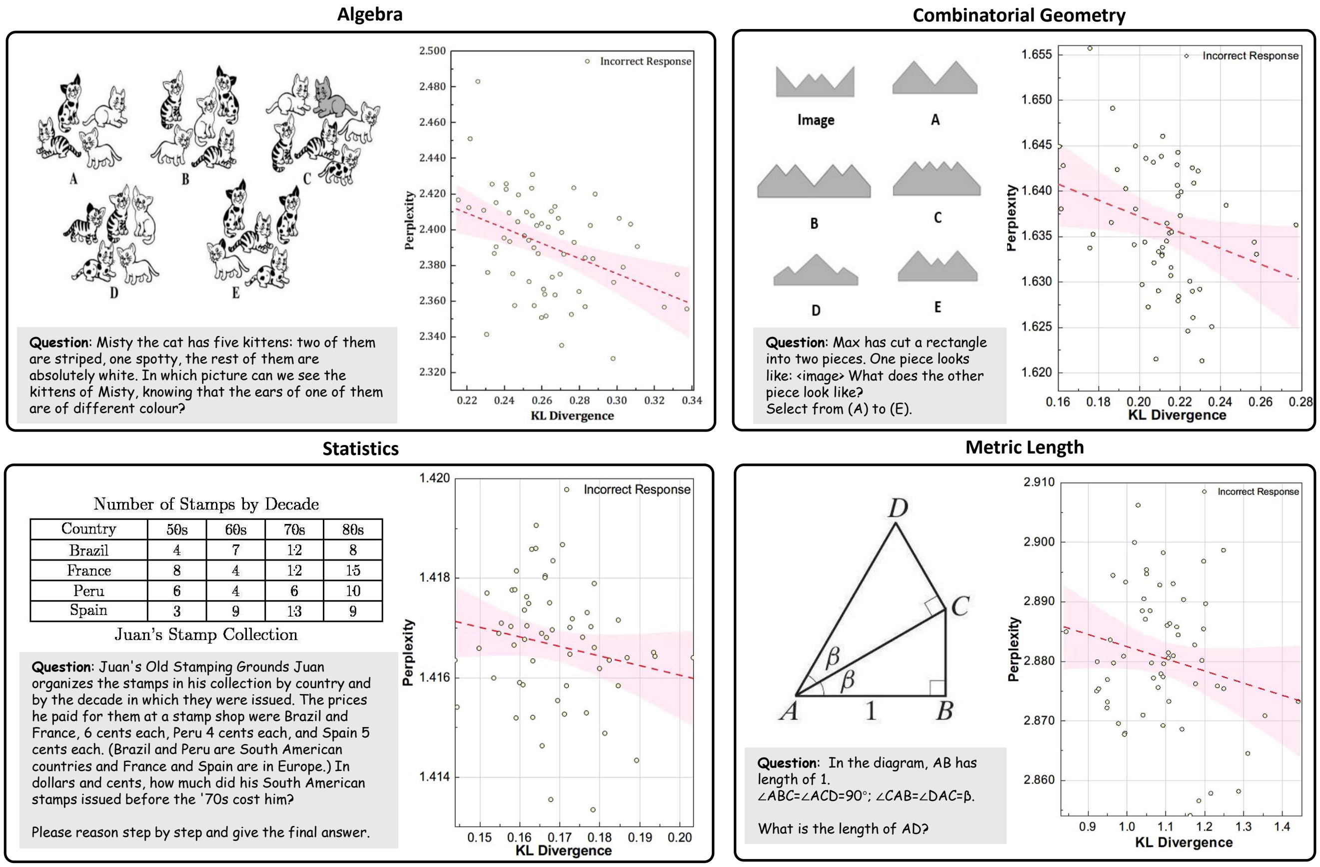


Figure 4. Perplexity Shifts on Incorrect Responses.

[14] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020. 1

[15] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025. 1

[16] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024. 1, 2

[17] Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. *arXiv preprint*, 2024. 1

[18] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*, 2023. 1

[19] 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

[20] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025. 1, 2, 4

[21] Rongyi Zhu, Yuhui Wang, Tanqiu Jiang, Jiacheng Liang, and Ting Wang. Self-improving model steering. *arXiv preprint arXiv:2507.08967*, 2025. 1

[22] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023. 1