On Test-Time Scaling for Vision-Language Models

Fawaz Sammani[1,2], Tzoulio Chamiti[1,2], and Nikos Deligiannis[1,2]

[1] ETRO Department, Vrije Universiteit Brussel, Belgium
[2] imec, Kapeldreef 75, B-3001 Leuven, Belgium
{fawaz.sammani, tzoulio.chamiti, nikos.deligiannis}@vub.be

Abstract. Test-time scaling is a paradigm where large models use additional compute at inference to achieve better performance, without changing model weights. While it has been widely studied for Large Language Models (LLMs), its applicability to Large Vision-Language Models (LVLMs) remains less explored and analyzed, with limited analysis of whether, when, and to what extent these approaches transfer to LVLMs. In this work, we ask a simple but fundamental question: can conventional test-time scaling methods developed for LLMs be directly applied to LVLMs? We present the first comprehensive study of test-time scaling for LVLMs, spanning multiple models and model sizes, nine test-time scaling methods, and six diverse benchmarks. Our main findings is that 1) different from previous findings, small, well-performing models benefit the most from test-time scaling, enabling performance improvements of up to around 30%, reaching large models performance, and often outperforming them, 2) LVLMs lose focus when given more compute than necessary, and 3) Visual information is encoded early in the reasoning chain, after which the chain is dominated by text-only reasoning and the contribution of image tokens drops significantly. Finally, we also provide a global and fine-grained analysis on the quality and information sufficiency of the reasoning chains produced. Overall, our findings and analysis provide practical guidance and insights into LVLMs and their deployment in research and industry.

1 Introduction

Test-time scaling (TTS) has emerged as a powerful way to improve model performance by allocating more computation at inference time, while keeping the model parameters fixed. It offers an alternative to parameter scaling (i.e., making models larger), delivering substantial accuracy gains on reasoning tasks, and recent evidence suggests that, for Large Language Models, test-time scaling can be more effective than increasing parameter count [30].

At its core, test-time scaling expands the model search space by eliciting and activating reasoning capabilities already present in the model [43]. Current approaches mainly fall into two paradigms. The first relies on additional training:

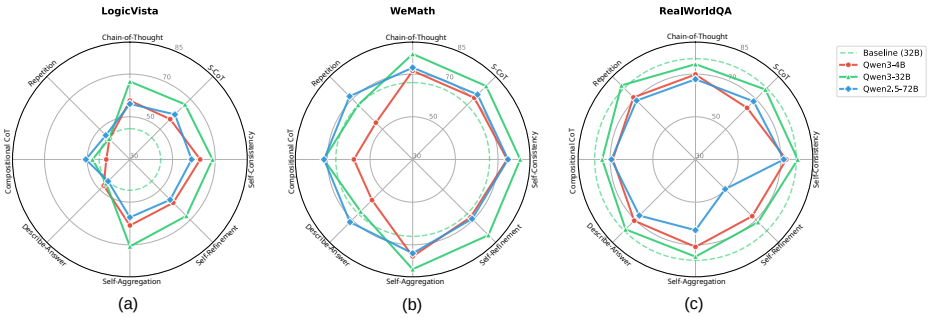

Fig. 1: Radar plot showing 8 test-time scaling methods on three benchmarks: reasoning (LogicVista, WeMath) and perception (RealWorldQA) for Qwen3-VL-4B, Qwen3-VL-32B and Qwen2.5-VL-72B. The dashed circle indicates baseline performance without test-time scaling, while solid lines correspond to test-time scaling strategies.

Large Language Models (LLMs) or Large Vision–Language Models (LVLMs) are fine-tuned on reasoning data via supervised fine-tuning, often followed by reinforcement learning [3, 23, 24, 32, 40, 41, 44, 45]. Models trained in this way are commonly referred to as "thinking models". The second paradigm achieves test-time scaling without further training, instead leveraging the instruction-tuned model and activating its reasoning behavior through prompting [34, 37], sampling [36], aggregation [15], or iterative refinement [19]. These methods are often described as *zero-shot* because they activate reasoning behavior without further training. Motivated by recent evidence that strong reasoning capabilities can already reside in instruction-tuned models [33, 43], our work focuses on this second paradigm.

While test-time scaling has been extensively studied for LLMs, it remains less explored and analyzed for LVLMs, especially across different model scales and benchmarks. As a result, it is still unclear whether, when, and to what extent these approaches transfer to LVLMs. Prior work on LLMs [11, 20, 24] report that test-time scaling is ineffective for small LLMs (<10B): it yields no accuracy gain, and prompting them to reason can even reduce accuracy. Recent work [9] evaluates several test-time scaling methods on small vision-language models and reports a similar trend. Other studies also suggest that Chain-of-Thought prompting (i.e., prompting the model to "think step-by-step") fails to improve performance in LVLMs [22]. This led researchers to develop test-time scaling techniques designed specifically for VLMs, under the assumption that test-time scaling methods for LLMs do not transfer effectively to VLMs.

In this work, we revisit these assumptions and ask a simple but fundamental question: can conventional test-time scaling methods developed for LLMs be directly applied to LVLMs? Our results indicate that the answer is yes—and that these methods are highly effective. We show that standard test-time scaling strategies from the LLM literature transfer well to LVLMs, and especially to small yet capable vision-language models. Interestingly, we find that small models often benefit substantially more from test-time scaling than larger ones, with performance improvements of up to 40%, reaching close-to baseline performance

of large models, and in some cases, even outperforming them. Figure 1 presents a radar plot comparing three different benchmarks across nine methods and three models. Figures 1(a,b) show that the smaller model Qwen3-VL-4B, when combined with test-time scaling, reaches and even surpasses the performance of the larger Qwen3-VL-32B baseline without test-time scaling (indicated by the dashed circle). We also find that these conventional test-time scaling methods significantly perform better than approaches developed specifically for vision-language models (e.g., Compositional CoT [22]).
On the other hand, we find that test-time scaling can hurt performance on perceptual tasks (i.e., tasks that do not require reasoning), rather than maintaining their performance (Figure 1c). By analyzing token budgets, we observe that VLMs become "lost" when given more compute than necessary; they tend to overthink, make wrong assumptions, and accumulate errors that ultimately lead to hallucinations and incorrect final answers. We further support this conclusion with an in-depth attention analysis of LVLMs. Specifically, we find that overly verbose generations rely heavily on early visual encoding; there is a short initial window during which generated tokens attend to the image and encode visual information into the hidden state. After this window, the model increasingly relies on that hidden representation, while attention to the image gradually decays over the remainder of the generation.
As a final contribution, we analyze the quality and information sufficiency of the generated reasoning chains at both global and fine-grained levels, using an LLM-as-a-judge approach and human evaluation.

2 Related Work

Early approaches to reasoning in vision and vision-language tasks [26,28] trained models to generate both an answer and a single reasoning phrase within a unified output text stream. This was later extended to zero-shot settings [27]. With the rise of large language models (LLMs), several techniques have been introduced to improve their performance via prompting [34,37], sampling and aggregation [15,36], refining the model's own answers through iterative feedback [19] and others. The core idea of these techniques is to spend more compute at inference time before producing the final answer. This is commonly referred to as test-time scaling. Several works also introduce approaches for test-time scaling tailored specifically for vision–language models. For example, CCoT [22] extracts a scene graph from the input image and injects it back into the prompt. DCoT [8] performs multi-pass prompting by first extracting bounding-box-based visual cues and then conditioning subsequent steps on these localized signals. DD-CoT [46] separates recognition and reasoning into sequential prompting stages, while [9] aggregates token output distributions over augmented image-text pairs.

Most of these works report that Chain-of-Thought prompting and other inference-time scaling methods fail and even yield negative gains for LVLMs [9, 22], mirroring findings for small LLMs [11,20,24]. In contrast, we revisit this conclusion

by systematically testing whether, when and to what extent do classic LLM test-time scaling methods transfer directly to LVLMs, and we examine this question across model scales and generations and across different benchmarks, coupled with deep analysis of attention and reasoning-chain sufficiency.

3 Test-Time Scaling of Vision-Language Models

A LVLM typically consists of (1) A vision encoder that encodes an image $I \in \mathbb{R}^{3\times H\times W}$ into a sequence of visual tokens $P_v \in \mathbb{R}^{N\times d}$, where N is the number of visual tokens and d is the dimensionality of each token, (2) a pretrained Large Language Model (LLM), and (3) an adapter that projects the visual tokens P_v into the LLM's language embedding space. Some LVLMs additionally compress the visual tokens to reduce the token count. We denote the adapter output by $P_v^{ad} \in \mathbb{R}^{N_m\times d}$, where N_m is the number of (compressed) visual tokens that enter to the LLM. For example, in the Qwen series, $N_m = N/4$. The task of the LVLM is to respond to an instruction provided by the user. For Visual Question Answering (VQA), the instruction is a `question` about the image. The `question` can be (i) an open-ended question, where the model must answer with a single word or phrase; (ii) a multiple-choice question (MCQ), where the model must select one option from a provided list; or (iii) a passage followed by a question. We include a `post-prompt` to instruct the LVLM to answer in a structured format. The full text prompt is therefore {`question`};{`post-prompt`}, where ";" indicates string concatenation (see Section S9 of the Supp. material for exact details on `post-prompt`). This prompt is tokenized into a sequence of text tokens $P_t \in \mathbb{R}^{T\times d}$, where T is the total number of text tokens and d is the token dimensionality. The total input sequence P to the LVLM is the concatenation of the visual and textual tokens, i.e., $P \in \mathbb{R}^{(N_m+T)\times d}$. The LVLM outputs the task `answer`, which may be a single letter (for multiple-choice questions) or a single word or phrase (for open-ended questions).

Methods: We consider a large suite of test-time scaling methods, some borrowed from the LLM literature and others specifically designed for LVLMs. All these methods are applied at inference time to elicit and activate reasoning capabilities of the LVLM, without changing its weights or performing any additional training. All techniques below are implemented via appending a `think-prompt` before `post-prompt`. For example, for Chain-of-Thought prompting, the `think-prompt` is "think step-by-step". The final input prompt P_t is therefore

{`question`};{`think-prompt`};{`post-prompt`}.

The LVLM outputs a reasoning chain r followed by the `answer`. Note that although the `post-prompt` instructs the LVLM to output `answer` in a specific format, the model can still occasionally fail to comply. To reliably extract the answer in such cases, if the answer extraction fails, we perform a second run by providing

{`question`};{`r`};{`answer-only`};{`post-prompt`}.

where `answer-only` prompts the LVLM to answer `question` given r. We consider the techniques below, and provide the `think-prompt` for each method in Section S8 of the Supp. material.

① **Chain-of-Thought (CoT)** prompting [37] elicits the LVLM to generate intermediate reasoning steps before producing a final answer by prompting it to think step by step. This method uses a single forward pass, but consumes more "thinking" tokens.

② **Structured Chain-of-Thought (S-CoT)** prompting is a more structured variant of ① used to guide the LVLM through question decomposition, information organization, logical reasoning, and final summarization.

③ **Plan-and-Solve (PaS)** prompting [34] first asks the model to understand and devise a plan for the problem and then to execute that plan step by step to produce the final answer. Like CoT and S-CoT, this method uses a single forward pass, but consumes more "thinking" tokens.

④ **Self-Consistency** [36] samples multiple independent CoTs (i.e., applies ① for b runs), producing b different reasoning chains for the same problem, and then selects the most frequent answer as the final response (see Section S11 of the Supp. material for sampling hyperparameters). This technique promotes diversity in intermediate reasoning, helping the LVLM explore richer solution paths. It uses b forward passes, each producing more "thinking" tokens.

⑤ **Self-Aggregation** [15] is similar to ④ with one key difference; it first produces b different reasoning chains, but instead of discarding each and keeping only its final answer, it retains all b sampled chains and their answers, concatenates them into a single chain, and prompts the LVLM to aggregate the diverse information across chains to produce a final reasoning chain and answer. This method requires $b + 1$ forward passes, each producing more "thinking" tokens.

⑥ **Self-Refinement** [19] begins with a single CoT reasoning chain (①) and then sequentially prompts the LVLM to refine and improve it for k rounds. This process enables the LVLM to self-correct and self-verify over successive iterations. This method requires $k + 1$ forward passes, the first producing more "thinking" tokens and the successive passes producing more "self-reflective" tokens.

⑦ **Describe-Answer** [7] first prompts the LVLM to caption the image I in full detail, including the objects present and their interactions. The generated caption, together with the image, question and a post-prompt, is then provided as input to the LVLM, which is asked to produce the final answer. This checks whether the LVLM's performance improves if elicited with a text form of the image. This method requires two forward passes, and only the first pass requires more "thinking tokens" where those tokens are represented by the caption.

⑧ **Compositional Chain-of-Thought** prompting [22] first prompts the LVLM to generate a scene graph for the image I given the `question`. The generated scene graph, together with the image, question and a post-prompt, is then provided as input to the LVLM, which is asked to produce the final answer. Similar to ⑥, this method requires two forward passes, and only the first pass requires more "thinking tokens" where those tokens are represented by the scene graph.

⑨ **Prompt Repetition** is a recent technique [12] that simply repeats the `question`. In this setting, `think-prompt` = `question`. Since LVLMs are trained and evaluated autoregressively, earlier prompt tokens cannot attend to later ones. Repeating the question twice helps address this limitation, creating an effect of bidirectional attention. This technique has also proven valuable for language embeddings [31]. This method uses a single forward pass, without producing any "thinking tokens".

4 Experiments and Takeaways

Models: In this work, we study 13 *instruction-tuned* models spanning a range of scales from the Qwen-2.5-VL [2], Qwen-3-VL [1] series, InternVL-3.5 [35] series and the Molmo2 [5] series. For Qwen-2.5-VL, we consider model sizes of {7B, 32B and 72B}. For Qwen-3-VL, we consider model sizes of {2B, 4B, 8B, and 32B}. For InternVL-3.5, we consider model sizes of {2B, 4B, 8B, and 38B}. Finally, for Molmo2, we consider model sizes of {4B, 8B}. To the best of our knowledge, the Qwen-VL and InternVL series are the strongest open-source models, achieving close-to and even surpassing top-tier closed-source models on various benchmarks [1, 2], and heavily used in industry. For all models, we set the maximum token length to 1,024. The total cost of our experiments is approximately $6,000.

Benchmarks: We consider six diverse benchmarks: MMStar [4], RealWorldQA [38], HallusionBench [6], WeMath [25], LogicVista [39] and A-OKVQA [29]. MMStar assesses coarse perception, fine-grained perception, mathematics, science and technology, instance reasoning, and logical reasoning. It therefore serves as both a perception and a reasoning benchmark, and consists of curated, reviewed samples drawn from existing benchmarks such as MMBench [16], MathVista [17], MMMU [42], ScienceQA [18], SEED [14], and AI2D [10]. RealWorldQA is a perception benchmark focused on spatial understanding. A-OKVQA is also a perception benchmark. HallusionBench is both a perception and visual reasoning benchmark, designed to evaluate hallucination in LVLMs. WeMath and LogicVista are mathematical and logical reasoning benchmarks, respectively.

Results: Results are presented in Table 1 for the Qwen Series. We color-code metrics in red when results degrade relative to the baseline, and in green otherwise, with the color intensity indicating the magnitude of the improvement or degradation. We also report the runtime (in seconds) in the last column for each method and model. Runtimes are measured on a single NVIDIA H200 GPU with 140 GB of VRAM, except for Qwen-2.5-VL-72B, which was run in parallel across two H200 GPUs due to memory constrains of a single H200 GPU. For each benchmark, we compute the runtime by averaging over all samples in that benchmark. We then obtain a single runtime value per method and model by averaging across all five benchmarks. By analyzing the results across all models and benchmarks, we find that overall, test-time scaling greatly improves performance across most models and benchmarks.

Table 1: Test-Time scaling results on the Qwen series, with degradation means a lower score than baseline, and improvement means a higher score than baseline

Model	Method	Dataset					
		MMStar	RealWorldQA	HallusionBench	WeMath	LogicVista	Runtime (s)
Qwen3-VL-2B	Baseline [1]	51.69	65.10	63.72	35.46	32.59	0.1
	CoT [37]	61.14	60.65	69.93	55.92	37.95	4.5
	S-CoT	55.05	52.94	68.35	51.03	40.40	4.6
	PaS [34]	57.33	63.27	71.50	55.40	40.62	4.4
	Self-Consistency [36]	63.83	64.31	71.08	64.25	46.88	40.2
	Self-Refinement [19]	60.87	59.22	69.72	56.15	39.06	9.0
	Self-Aggregation [15]	62.26	61.70	71.50	58.68	41.07	24.0
	Describe-Answer [7]	53.49	66.54	66.35	45.00	30.13	6.4
	CCoT [22]	52.65	61.96	60.57	48.74	34.60	5.8
	Prompt Repetition [12]	50.62	65.36	64.04	39.20	32.59	0.1
Qwen3-VL-4B	Baseline [1]	61.71	71.76	70.66	58.22	40.85	0.1
	CoT [37]	68.59	69.80	75.50	71.55	57.59	5.9
	S-CoT	68.47	64.18	75.50	70.86	56.70	8.2
	PaS	69.38	70.46	74.76	67.41	54.69	8.0
	Self-Consistency [36]	71.07	72.55	74.97	74.25	62.95	44.8
	Self-Refinement [19]	68.19	67.58	73.29	68.33	58.71	11.2
	Self-Aggregation [15]	70.51	70.98	75.71	75.40	60.94	31.4
	Describe-Answer [7]	62.62	70.59	71.29	56.95	47.32	9.0
	CCoT [22]	62.81	68.76	69.61	57.59	41.07	4.7
	Prompt Repetition [12]	62.03	71.11	70.45	54.37	43.53	0.1
Qwen3-VL-8B	Baseline [1]	64.95	69.54	72.98	57.76	40.62	0.1
	CoT [37]	69.77	72.03	76.55	74.37	55.13	5.9
	S-CoT	69.08	68.63	75.08	73.62	56.70	7.2
	PaS	70.47	69.28	75.29	74.89	58.04	6.9
	Self-Consistency [36]	72.99	72.68	75.71	76.90	61.61	44.8
	Self-Refinement [19]	69.16	68.63	72.98	72.41	56.92	11.3
	Self-Aggregation [15]	72.19	71.11	75.50	77.47	63.62	31.1
	Describe-Answer [7]	63.92	69.02	72.98	56.61	37.72	9.6
	CCoT [37]	65.47	69.02	72.24	60.40	38.17	5.9
	Prompt Repetition [12]	65.02	69.80	72.45	58.22	40.62	0.1
Qwen2.5-VL-7B	Baseline [1]	62.57	70.20	72.03	58.28	39.06	0.1
	CoT [37]	63.46	65.88	69.61	65.40	38.17	2.1
	S-CoT	64.11	61.70	70.24	62.24	42.41	2.7
	PaS	61.88	64.18	69.30	64.48	44.42	2.5
	Self-Consistency [36]	65.29	66.01	71.08	69.25	41.52	17.3
	Self-Refinement [19]	61.33	47.97	68.66	63.85	41.07	4.6
	Self-Aggregation [15]	62.04	61.44	70.45	67.59	49.11	11.8
	Describe-Answer [7]	60.41	67.45	70.03	54.71	36.16	5.0
	CCoT [37]	60.58	62.75	69.51	59.25	38.39	3.3
	Prompt Repetition [12]	60.38	67.97	71.29	59.48	40.62	0.1
Qwen3-VL-32B	Baseline [1]	72.66	77.25	74.76	66.03	44.42	0.2
	CoT [37]	74.88	74.64	78.13	79.43	66.52	12.2
	S-CoT	74.67	76.34	76.55	78.68	66.52	19.5
	PaS	76.05	76.34	76.13	78.97	66.52	18.9
	Self-Consistency [36]	76.46	77.91	77.71	80.40	68.75	88.9
	Self-Refinement [19]	73.21	71.50	77.92	80.17	67.41	26.5
	Self-Aggregation [15]	75.85	75.56	77.50	81.44	70.76	65.6
	Describe-Answer [7]	70.20	76.34	74.45	64.48	45.76	26.9
	CCoT [37]	72.36	73.73	74.55	71.67	47.54	17.5
	Prompt Repetition [12]	72.07	79.08	75.08	66.15	43.75	0.2
Qwen2.5-VL-32B	Baseline [2]	67.35	71.24	70.35	65.80	44.87	0.2
	CoT [37]	66.87	66.80	70.77	74.31	56.03	8.7
	S-CoT	68.37	69.02	71.92	73.28	56.47	15.2
	PaS	68.31	69.28	70.45	74.48	56.03	14.8
	Self-Consistency [36]	68.97	68.10	71.50	76.78	56.47	64.1
	Self-Refinement [19]	65.62	49.93	71.40	72.87	54.91	20.6
	Self-Aggregation [15]	67.16	60.78	70.45	76.21	58.04	48.8
	Describe-Answer [7]	66.32	69.15	70.14	63.05	39.51	23.1
	CCoT [37]	65.58	67.32	73.50	70.29	46.43	21.0
	Prompt Repetition [12]	66.88	69.41	70.45	66.26	43.75	0.2
Qwen2.5-VL-72B	Baseline [2]	68.65	69.67	74.97	72.99	47.54	0.3
	CoT [37]	68.58	67.58	72.98	73.05	56.03	11.6
	S-CoT	68.77	68.50	73.82	72.99	59.82	16.3
	Self-Consistency [36]	70.86	71.37	72.98	74.66	58.93	85.7
	Self-Refinement [19]	68.89	49.67	71.61	69.37	56.70	25.3
	Self-Aggregation [15]	70.44	63.14	72.87	73.97	57.14	68.6
	Describe-Answer [7]	66.78	67.19	72.66	71.55	44.42	26.7
	CCoT [37]	67.99	69.41	73.40	71.55	50.67	15.6
	Prompt Repetition [12]	69.40	69.02	75.18	71.84	45.98	0.2

Interestingly, and different from prior LLM works which report that small models do not benefit from test-time scaling [11, 20, 24], we observe the opposite in LVLMs. Our results suggest that small models benefit the most from test-time scaling. For example, self-consistency boosts the performance of Qwen3-VL-2B from 35% to 64% on WeMath (approximately a 30% increase in performance). Our findings also suggest that small models with test-time scaling can outperform larger ones. For example, Qwen3-VL-4B with simple CoT and S-CoT already surpasses the baseline performance of Qwen3-VL-32B on HallusionBench, WeMath and LogicVista. On LogicVista for example, Qwen3-VL-4B with CoT achives a +13% improvement over Qwen3-VL-32B. The gain is even larger with Self-Consistency. We also find that simple prompt repetition helps on tasks that require reasoning across the prompt, such as when the prompt describes a mathematical problem and earlier prompt tokens benefit from future prompt tokens. For example, on WeMath, simple prompt repetition yields an improvement of about +4% for Qwen3-VL-2B, while the gains are less pronounced for other models. Figure 2 presents the Pareto frontiers for HallusionBench, LogicVista, and WeMath, comparing Qwen3-VL-4B with test-time scaling against the ×8 larger Qwen3-VL-32B baseline (◆, dashed line). The x-axis represents compute time, while the y-axis shows accuracy. The figure illustrates whether, under a fixed compute budget, it is better to allocate resources to a smaller model with test-time scaling or to a larger model without it. For example, at a compute budget of approximately 9 seconds, the 4B model matches the performance of the 32B model on HallusionBench, nearly matches it on MMStar, and outperforms it on both LogicVista and WeMath. Overall, we find that simple Chain-of-Thought prompting (CoT) offers the best tradeoff between runtime and performance improvements.

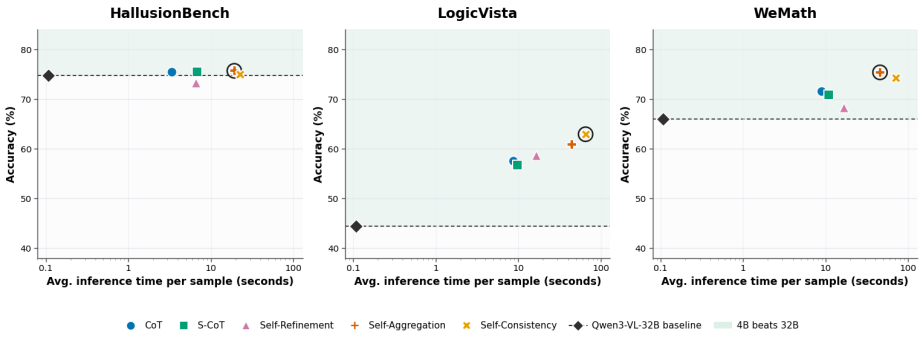

Fig. 2: Compute vs Accuracy Pareto frontiers. We circle the best method.

It is important to note that a very recent work [9] also studies some test-time scaling methods for a small vision-language model (SmolVLM2 [21]) and report failures, and other works [22] investigate chain-of-thought prompting for the LLaVa series, and similarly finds it ineffective. To understand this discrepancy between our findings and the literature, we examined model behavior and found that LLaVA-OneVision [13] and SmolVLM2 [21] often show poor instruction com-

pliance: they answer questions directly while ignoring the `think-prompt`. They behaved more like specialized VQA systems rather than instruction-following chat agents (see Section S12 of the Supp. material for examples). Therefore, our results suggest that *capable* instruction-following models benefit the most from test-time scaling. This suggests that we can deploy cheaper, faster, lower-memory VLMs and use test-time scaling to get near–large-model accuracy (and even outperform them) in production and research.

Takeaway 1: Smaller models benefit the most : Small (2B to 8B scale), yet capable instruction-tuned models, benefit the most from test-time scaling, often reaching close to performance of larger models (and sometimes even outperforming them), suggesting an accuracy–compute trade-off where cheaper, faster and lower-memory VLMs can be deployed.

Small models may not have the same direct ability to solve complex tasks as larger models. However, they still contain relevant information compressed in their weights and possess some capacity for reasoning. When this reasoning ability is activated, they can decompose a difficult problem into a sequence of simpler steps, solve each step one by one, and ultimately arrive at the correct answer.

We also find that test-time scaling methods often degrade performance (rather than maintaining it) on primarily perception-focused benchmarks that require limited or no reasoning (e.g., RealWorldQA). We additionally verify this claim in two ways: (1) by reporting additional results on another perception-only benchmark, A-OKVQA (Section S3.1 of the Supp. material), and (2) by dissecting the individual category scores on MMStar, showing that the degradation occurs mainly in the non-reasoning categories (Section S3.2 of the Supp. material). Notably, perception benchmarks are equally important, as they often reflect typical-day user queries. To test whether this effect also holds for strong closed-source models, we ran experiments on MMStar and RealWorldQA using GPT-5.2, OpenAI's strongest model at the time of writing. The results are reported in Section S3.3 of the Supp. material. We find that this issue is less pronounced for GPT-5.2, but in some cases, still causing degradation.

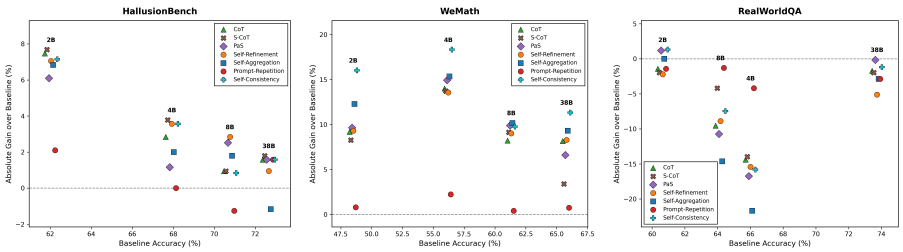

Fig. 3: Absolute Gains of Test-Time Scaling Methods on the InternVL-3.5 series

Generalization to other LVLMs: Figure 3 shows the absolute gains (in %) of test-time scaling methods over the baseline (dashed line) for the InternVL-3.5

series. The results confirm our main findings: smaller models benefit most from test-time scaling, while test-time scaling degrades performance on perceptual-only benchmarks such as RealWorldQA. Tabular results are in Section S1 of the Supp. Material for InternVL-3.5 and in Section S2 for the Molmo2 series.

A question that remains unanswered is: *why does extra test-time compute harm perception-only benchmarks?* We answer this via Token Budget analysis.

Token Budget: Token Budget refers to the maximum number of test-time scaling tokens generated. To study how token length affects performance gains from test-time scaling methods in vision-language models, we repeated all experiments (Section 3) using a reduced token budget. Specifically, we set the maximum output length to 300 tokens. Because we use the reliable answer extraction method (Section 3), we can still reliably extract the final answer even if the reasoning chain is truncated at 300 tokens. Results are presented in Table 2. We report the difference in score when the reasoning chain is stopped early at 300 tokens.

In contrast to findings in the LLM literature suggesting that models often "know" the answer before producing a chain of thought [11]—that is, the conclusion is reached early and the final answer may not depend on the reasoning trace—our experiments indicate the opposite. Reducing the token budget degrades performance, implying (1) the reasoning chain drives the correct answer, and (2) using more tokens to generate intermediate steps is beneficial for reaching the correct answer. This generally holds across all reasoning benchmarks, but not across perceptual and hallucination benchmarks. On RealWorldQA and often on HallusionBench (as seen in Table 2), increasing the token budget can hurt performance, or at least does not result in any improvements. In Section S4 of the Supp. material, we additionally verify this also on (1) the perceptual categories of MMStar, (2) A-OKVQA dataset and (3) InternVL-3.5. This suggests that short answers are best for perceptual tasks, and overly verbose responses drive the LVLM to "lose focus"; if the LVLM is given the opportunity to overthink (via more tokens), it tends to elaborate plausible narratives, make wrong assumptions and accumulate chain errors, leading to hallucinations and eventually drifting away from the correct answer. We provide supporting evidence for this hypothesis via internal model attention analysis in the next section. Overall, our analysis suggests that test-time scaling is primarily helpful when it enables genuinely useful reasoning, For tasks that do not require reasoning, applying test-time scaling yields generally no improvement and, in the worst case, even degrade performance. This implies that **substantial compute can be saved** by identifying whether a task actually requires test-time scaling, for example by training a simple binary classifier.

Takeaway 2: LVLMs lose focus when given more compute than necessary : Perception and hallucination benchmarks favor concise outputs, and extra test-time compute that enables verbosity often drives LVLMs to overthink and hallucinate, misleading the final answer. This suggests that a large amount of compute can be saved for tasks that do not require reasoning.

Table 2: Early stopping at token 300. We report $\Delta = \text{score}_{300} - \text{score}_{1024}$. improvement indicates a positive Δ (300-token early stopping improves performance). degradation indicates negative Δ (performance drops under the 300-token stopping).

Model	Method	MMStar	RealWorldQA	Hall.Bench	WeMath	LogicVista
Qwen3-VL-4B	CoT	-2.07	0.79	-0.42	-11.32	-7.37
	S-CoT	-2.80	3.79	-0.11	-14.89	-13.61
	Self-Ref.	-2.31	-0.26	-0.31	-7.35	-2.45
	Self-Agg.	-2.57	0.26	-0.74	-15.06	-13.84
	Self-Cons.	-3.26	0.65	1.06	-10.46	-9.37
Qwen3-VL-8B	CoT	-0.55	-0.79	-2.10	-6.04	-0.44
	S-CoT	-0.80	0.52	0.84	-10.29	-1.57
	Self-Ref.	-1.82	-0.13	0.84	-5.92	-1.34
	Self-Agg.	-2.87	-1.18	-1.16	-9.60	-8.04
	Self-Cons.	-3.15	-0.78	-0.32	-7.88	-1.57
Qwen3-VL-32B	CoT	-0.62	0.00	-1.58	-7.25	-3.13
	S-CoT	-1.37	1.57	0.21	-11.84	-5.58
	Self-Ref.	-0.15	0.13	-2.32	-8.91	-3.79
	Self-Agg.	-1.36	0.65	-0.63	-10.23	-4.91
	Self-Cons.	-1.64	-0.66	-1.16	-7.24	-6.70
Qwen2.5-VL-7B	CoT	-0.58	0.13	1.37	-2.24	0.00
	S-CoT	-1.11	0.26	-0.84	-2.58	0.00
	Self-Ref.	-0.57	0.27	-0.10	-2.93	0.22
	Self-Agg.	1.24	1.31	-0.63	-5.87	-6.48
	Self-Cons.	0.05	-2.22	-0.73	-4.59	-0.45
Qwen2.5-VL-32B	CoT	-1.28	0.00	1.15	-5.57	-2.91
	S-CoT	-2.66	0.52	0.00	-9.26	-7.59
	Self-Ref.	0.63	2.36	-1.16	-5.86	-2.90
	Self-Agg.	0.05	0.66	0.84	-8.51	-3.80
	Self-Cons.	-0.48	0.27	0.95	-7.30	0.45
Qwen2.5-VL-72B	CoT	-0.34	2.22	0.94	0.05	0.00
	S-CoT	-1.58	1.04	-0.53	-0.75	-0.44
	Self-Ref.	-0.15	10.07	1.68	2.76	-0.23
	Self-Agg.	-1.52	6.14	1.58	-1.10	-0.67

5 Attention Dynamics of Reasoning

We remind the reader from Section 3 that $P_v^{ad} \in \mathbb{R}^{N_m \times d}$ denotes the visual tokens produced by the adapter and passed to the LLM. In this section, we analyze how the LVLM's chain-of-thought reasoning interacts with the image tokens P_v^{ad}. Because most test-time scaling methods rely on CoT prompting as the underlying foundation, we adopt CoT as the basis for our analysis. Given an input prompt P_t (Section 3), the LVLM generates a token sequence g. Let g_y denote the y-th generated reasoning token. We examine, for each g_y, its interaction with (i) the image tokens P_v^{ad} and (ii) the previously generated reasoning tokens $g_{0\ldots y-1}$. Concretely, at layer l and attention head h, the attention map a_l^h for token g_y assigns attention over the N_m image tokens (P_v^{ad}), the T text prompt tokens (P_t), and the preceding generated tokens $g_{1\ldots y-1}$. We decompose a_l^h for g_y into three components: (1) attention allocation to

the image tokens, $a^{img} = \sum a_l^h[1 : N_m]$, (2) attention to the text prompt $a^{pmt} = \sum a[N_m + 1 : N_m + T]$, and (3) attention to previously generated reasoning tokens $a^{gen} = \sum a[N_m + T + 1 : N_m + T + (y - 1)]$, where y is the index of the current generated token. Figure 4 plots a^{img} and a^{gen} attention traces as functions of the generated reasoning tokens for WeMath (Figure 4a) across different models, averaging over all layers and heads and over all samples in each dataset. The attention from g_y to the T text tokens in P_t can be directly derived as $a^{pmt} = 1 - (\sum a^{img} + \sum a^{gen})$; because our focus is the contribution of image attention as tokens are generated, we omit this term from the plots. In Figure 4b, we additionally report an **upper bound** on image attention for WeMath, respectively, by taking, for each g_y, the maximum image-attention value across all layers and heads, i.e., we plot $max[a^{img}]$. Similar trends hold for other datasets and test-time scaling methods. We refer the reader to Section S5 of the Supp. material for the corresponding results. Plotting details are provided in Section S13 of the Supp. material.

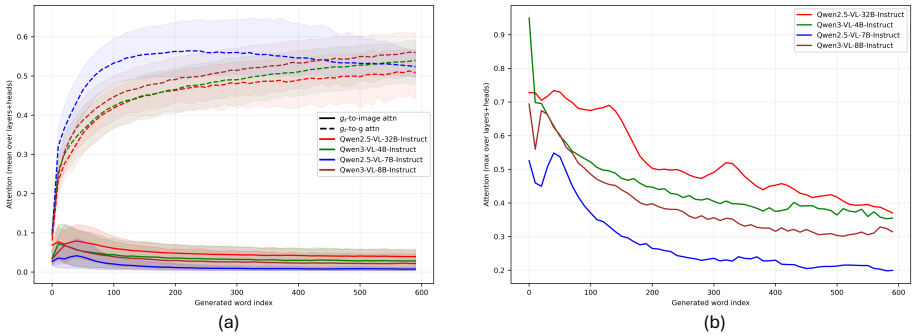

Fig. 4: Attention dynamics during chain-of-thought generation. For each generated token g_y, we measure attention to image tokens P_v^{ad} (*solid curve*) and previously generated tokens (*dashed curve*). **(a)** Average attention across layers, heads, and samples for WeMath. **(b)** Upper bound on image attention, defined as the maximum attention to image tokens across layers and heads.

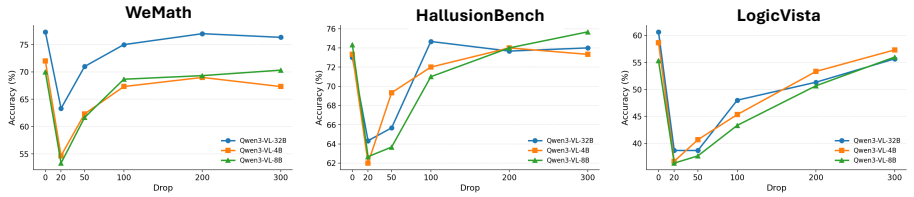

Fig. 5: Image Token Dropping vs Accuracy at Different Generation Steps

From Figure 4, we observe that across both datasets, attention mass quickly shifts toward previously generated reasoning tokens, while average attention to image tokens remains low and decays over the generation. Image attention peaks early and then drops rapidly, while attention to previously generated tokens in-

creases and eventually dominates. This indicates that visual grounding is largely front-loaded: CoTs typically have an early, brief window of image-focused attention during which visual information is encoded and absorbed into the text tokens hidden representations. Even the maximum image attention per token drops steadily as generation progresses.

Taken together, these patterns suggest that long CoTs rely increasingly on previously generated textual tokens for reasoning, and use primarily the visual information already encoded during that initial window rather than repeatedly attending to the image. This helps explain Takeaway 2: when test-time compute exceeds what is necessary and generation becomes overly verbose, the LVLM leans more heavily on the generated tokens to reason, producing assumptions that can accumulate through those interactions, and ultimately drift away from the correct answer.

Image Key-Value Cache Dropping: We verify the above through causal intervention based on image token dropping. Specifically, we take 300 random samples for each dataset, and, at different generation steps, completely remove the image tokens (i.e., their Key/Value cache) from all layers, and continue generation. We then measure the accuracy for each drop step. We consider drop steps at {20, 50, 100, 200 and 300}. The results are shown in Figure 5, where the x-axis represents the drop step and the y-axis represents the accuracy. Dropping image tokens at early generation steps leads to a clear performance degradation. This is because visual information is removed during the critical front-loading phase, when it is encoded and absorbed into the text-token representations. When image tokens are dropped later in the generation process, the effect becomes much less pronounced. After approximately 200 generation steps, removing the image tokens has almost no effect on accuracy. Intervention results for other datasets are presented in Section S6 of the Supp. material. We also provide computational savings (in GFLOPs) in Section S7 of the Supp. material.

Takeaway 3: Long CoTs rely on early visual encoding : During CoT generation, attention to image tokens is strongest at the beginning and gradually declines, while attention to previously generated tokens increases. This suggests that later steps depend on visual information encoded early and absorbed into the text token representations, relying more on prior textual context for reasoning rather than repeatedly attending to the image.

6 Analysis of Multimodal Chain-of-Thoughts

LLM-as-a-Judge: To evaluate the quality of the generated multimodal rationales, we employ an external LLM as a judge, and later validate its agreement with human evaluation. Given a question q, a ground-truth answer x and and a rationale r of length Y produced by the LVLM with CoT prompting, we measure two complementary metrics: **(1) Rationale Sufficiency**, where the judge receives (q, r) and predicts an answer $\hat{x}_r$ based solely on the textual content of

r, without access to the image, external knowledge or the ground-truth answer x. The idea is that if a judge can answer the question solely by reading the reasoning chain r, then r is highly informative and contains sufficient information. The rationale accuracy is computed by comparing $\hat{x}_r$ to x. We then compare this rationale accuracy with that of the LVLMs. **(2) Rationale Dynamics**, a fine-grained analysis of (1), where the rationale r is decomposed into its components, represented by an ordered sequence of sentences $\{r_1, \ldots, r_S\}$ where S represents the total number of sentences in r. Each sentence r_s is evaluated independently to determine whether its information is sufficient enough to supports a positive answer, a negative answer, or an uncertain answer with respect to q. From these sentence-level signals, we derive aggregate statistics including the earliest decision point, shifts or contradictions across reasoning steps, patterns of intermediate uncertainty, and the position of final commitment to the answer within the rationale. All prompts are provided in Section S14 of the Supp. material.

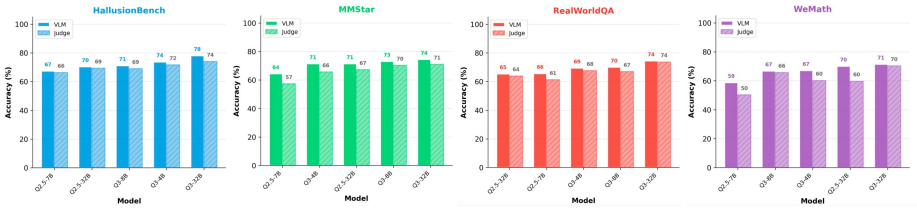

Fig. 6: LVLM vs Judge accuracy across benchmarks and model scales. Results are computed on 300 random samples per benchmark with $Y = 300$ tokens.

Rationale Sufficiency: The results are presented in Figure 6. Across most benchmarks and model scales, judge accuracy closely follows overall LVLM performance, indicating that the rationales capture a meaningful decision-relevant signal. On WeMath, there exists some models with a small gap between the LVLM performance and the judge accuracy (e.g., Qwen-2.5-VL 7B and 32B), where rationale-level accuracy lags behind LVLM accuracy, indicating that intermediate reasoning steps are not fully expressed. In general, we find that higher quality models (either in terms of size or generation) often narrow the gap, indicating better explainability via more informative rationales.

Human Evaluation: We also perform human evaluation to assess the reliability of the LLM judge. The results are presented in Section S10 of the Supp. material and indicate high consistency and strong agreement with the judge, supporting the judge use as a reliable proxy for evaluating rationale sufficiency.

Rationale Dynamics: We focus on HallusionBench as a representative benchmark combining perception, reasoning and hallucination. However, similar trends are observed on other benchmarks. Figure 7 shows the temporal position of key reasoning events on HallusionBench, measured as normalized position within the rationale (0 = start, 1 = end). According to the judge's predictions, uncertainty typically appears early in the chain (▲). That is, there is uncertain information

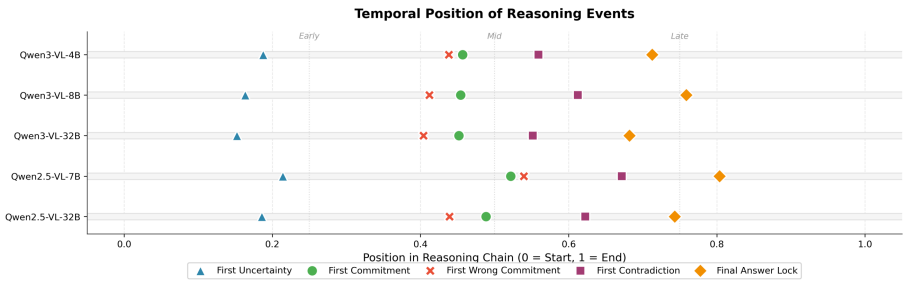

Fig. 7: Analysis of LLM-Judge evaluated rationale dynamics per different model on HallusionBench.

at that point in the chain to derive an answer. Explicit support for the correct answer then starts to appear midway in the chain (●). That is, the chain becomes more informative. This happens in around 75% of the samples. In the other 25% of instances, there is still no support for the correct answer (×). However, for those instances, the information content improves at around 0.6 of the chain (■) giving support for the correct answer. Shortly after, at around 0.75 of the chain (♦), the information is sufficient and informative enough, such that an answer can be derived without being changed in later steps. The observed temporal structure suggests that generated rationales tend to organize reasoning (through information sufficiency) progressively, rather than presenting a fully formed conclusion from the outset.

7 Conclusion

We presented the first broad study of zero-shot test-time scaling for LVLMs. We find that standard LLM test-time scaling methods transfer well to LVLMs and are often *most* effective for small instruction-following models, yielding large gains. However, extra compute can hurt on perception-focused tasks, where long generations cause models to overthink and loose focus. We also show that visual information is encoded early, and image tokens contribute less as generation progresses. Overall, zero-shot test-time scaling is a strong accuracy–compute lever for LVLMs, particularity for small models, but should be applied carefully.

Acknowledgements

Fawaz Sammani is funded by the Fonds Wetenschappelijk Onderzoek (FWO) (PhD) fellowship strategic basic research 1SH7W24N). T. Chamiti and N. Deligiannis acknowledge the ”Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen” programme and the ERC Consolidator Grant IONIAN (No. 101171240, DOI: 10.3030/101171240). Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
3. Chen, H., Tu, H., Wang, F., Liu, H., Tang, X., Du, X., Zhou, Y., Xie, C.: SFT or RL? an early investigation into training r1-like reasoning large vision-language models. Transactions on Machine Learning Research (2025)
4. Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., Zhao, F.: Are we on the right way for evaluating large vision-language models? In: Advances in Neural Information Processing Systems (2024), `https://openreview.net/forum?id=evP9mxNNxJ`
5. Clark, C., Zhang, J., Ma, Z., Park, J.S., Salehi, M., Tripathi, R., Lee, S., Ren, Z., Kim, C.D., Yang, Y., Shao, V., Yang, Y., Huang, W., Gao, Z., Anderson, T., Zhang, J., Jain, J., Stoica, G., Han, W., Farhadi, A., Krishna, R.: Molmo2: Open weights and data for vision-language models with video understanding and grounding. ArXiv **abs/2601.10611** (2026)
6. Guan, T., Liu, F., Wu, X., Xian, R., Li, Z., Liu, X., Wang, X., Chen, L., Huang, F., Yacoob, Y., Manocha, D., Zhou, T.: Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14375–14385 (2024)
7. Guo, J., Li, J., Li, D., Tiong, A.M.H., Li, B.A., Tao, D., Hoi, S.C.H.: From images to textual prompts: Zero-shot visual question answering with frozen large language models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 10867–10877 (2022)
8. Jia, Z., Liu, J., Li, H., Liu, Q., Gao, H.: DCot: Dual chain-of-thought prompting for large multimodal models. In: The 16th Asian Conference on Machine Learning (Conference Track) (2024)
9. Kaya, M.O., Elliott, D., Papadopoulos, D.: Efficient test-time scaling for small vision-language models. In: International Conference on Learning Representations (2026)
10. Kembhavi, A., Salvato, M., Kolve, E., Seo, M., Hajishirzi, H., Farhadi, A.: A diagram is worth a dozen images. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) European Conference on Computer Vision. pp. 235–251. Springer International Publishing, Cham (2016)
11. Lanham, T., Chen, A., Radhakrishnan, A., Steiner, B., Denison, C.E., Hernandez, D., Li, D., Durmus, E., Hubinger, E., Kernion, J., Lukovsiut.e, K., Nguyen, K., Cheng, N., Joseph, N., Schiefer, N., Rausch, O., Larson, R., McCandlish, S., Kundu, S., Kadavath, S., Yang, S., Henighan, T., Maxwell, T.D., Telleen-Lawton, T., Hume, T., Hatfield-Dodds, Z., Kaplan, J., Brauner, J., Bowman, S., Perez,

E.: Measuring faithfulness in chain-of-thought reasoning. ArXiv **abs/2307.13702** (2023)
12. Leviathan, Y., Kalman, M., Matias, Y.: Prompt repetition improves non-reasoning llms. ArXiv **abs/2512.14982** (2025)
13. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., Li, C.: LLaVA-onevision: Easy visual task transfer. Transactions on Machine Learning Research (2025)
14. Li, B., Wang, R., Wang, G., Ge, Y., Ge, Y., Shan, Y.: Seed-bench: Benchmarking multimodal llms with generative comprehension. arXiv preprint arXiv:2307.16125 (2023)
15. Li, Z., Feng, X., Cai, Y., Zhang, Z., Liu, T., Liang, C., Chen, W., Wang, H., Zhao, T.: Llms can generate a better answer by aggregating their own responses. ArXiv **abs/2503.04104** (2025)
16. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? In: European Conference on Computer Vision. pp. 216–233. Springer (2024)
17. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In: International Conference on Learning Representations (2024)
18. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. In: Advances in Neural Information Processing Systems (2022)
19. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B.P., Hermann, K., Welleck, S., Yazdanbakhsh, A., Clark, P.: Self-refine: Iterative refinement with self-feedback. In: Advances in Neural Information Processing Systems (2023)
20. Magister, L.C., Mallinson, J., Adamek, J., Malmi, E., Severyn, A.: Teaching small language models to reason. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 1773–1781. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.acl-short.151, https://aclanthology.org/2023.acl-short.151/
21. Marafioti, A., Zohar, O., Farré, M., noyan, M., Bakouch, E., Jiménez, P.M.C., Zakka, C., allal, L.B., Lozhkov, A., Tazi, N., Srivastav, V., Lochner, J., Larcher, H., Morlon, M., Tunstall, L., Werra, L.V., Wolf, T.: SmolVLM: Redefining small and efficient multimodal models. In: Second Conference on Language Modeling (2025), https://openreview.net/forum?id=qMUbhGUFUb
22. Mitra, C., Huang, B., Darrell, T., Herzig, R.: Compositional chain-of-thought prompting for large multimodal models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 14420–14431 (2023)
23. Muennighoff, N., Yang, Z., Shi, W., Li, X.L., Li, F.F., Hajishirzi, H., Zettlemoyer, L.S., Liang, P., Candès, E.J., Hashimoto, T.: s1: Simple test-time scaling. In: Conference on Empirical Methods in Natural Language Processing (2025), https://api.semanticscholar.org/CorpusID:276079693
24. Ou, L.: Empowering lightweight mllms with reasoning via long cot sft. ArXiv **abs/2509.03321** (2025), https://api.semanticscholar.org/CorpusID:281092552
25. Qiao, R., Tan, Q., Dong, G., Wu, M., Sun, C., Song, X., GongQue, Z., Lei, S., Wei, Z., Zhang, M., et al.: We-math: Does your large multimodal model achieve human-like mathematical reasoning? arXiv preprint arXiv:2407.01284 (2024)

26. Sammani, F., Deligiannis, N.: Uni-nlx: Unifying textual explanations for vision and vision-language tasks. 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) pp. 4636–4641 (2023)
27. Sammani, F., Deligiannis, N.: Zero-shot natural language explanations. In: The Thirteenth International Conference on Learning Representations (2025)
28. Sammani, F., Mukherjee, T., Deligiannis, N.: Nlx-gpt: A model for natural language explanations in vision and vision-language tasks. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 8312–8322 (2022)
29. Schwenk, D., Khandelwal, A., Clark, C., Marino, K., Mottaghi, R.: A-okvqa: A benchmark for visual question answering using world knowledge. In: European Conference on Computer Vision (2022), https://api.semanticscholar.org/CorpusID:249375629
30. Snell, C.V., Lee, J., Xu, K., Kumar, A.: Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In: International Conference on Learning Representations (2025)
31. Springer, J.M., Kotha, S., Fried, D., Neubig, G., Raghunathan, A.: Repetition improves language model embeddings. In: International Conference on Learning Representations (2025)
32. Suma, A., Dauncey, S.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. ArXiv **abs/2501.12948** (2025)
33. Venkatraman, S., Jain, V., Mittal, S., Shah, V., Obando-Ceron, J., Bengio, Y., Bartoldson, B.R., Kailkhura, B., Lajoie, G., Berseth, G., Malkin, N., Jain, M.: Recursive self-aggregation unlocks deep thinking in large language models. ArXiv **abs/2509.26626** (2025)
34. Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R.K.W., Lim, E.P.: Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In: Annual Meeting of the Association for Computational Linguistics (2023)
35. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., Wang, Z., Chen, Z., Zhang, H., Yang, G., Wang, H., Wei, Q., Yin, J., Li, W., Cui, E., Chen, G., Ding, Z., Tian, C., Wu, Z., Xie, J., Li, Z., Yang, B., Duan, Y., Wang, X., Hao, H., Li, S., Zhao, X., Duan, H., Deng, N., Fu, B., He, Y., Wang, Y., He, C., Shi, B., He, J., Xiong, Y., Lv, H., Wu, L., Shao, W., Zhang, K., Deng, H., Qi, B., Qi, B., Guo, Q., Zhang, W., Gu, Y., Ouyang, W., Wang, L., Dou, M., Zhu, X., Lu, T., Lin, D., Dai, J., Zhou, B., Su, W., Chen, K., Qiao, Y., Wang, W., Luo, G.: Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. ArXiv **abs/2508.18265** (2025), https://api.semanticscholar.org/CorpusID:280710824
36. Wang, X., Wei, J., Schuurmans, D., Le, Q.V., Chi, E.H., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. In: International Conference on Learning Representations (2023)
37. Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain of thought prompting elicits reasoning in large language models. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022)
38. xai: Realworldqa (2024), https://huggingface.co/datasets/xai-org/RealworldQA
39. Xiao, Y., Sun, E., Liu, T., Wang, W.: Logicvista: Multimodal llm logical reasoning benchmark in visual contexts (2024)
40. Xu, G., Jin, P., Wu, Z., Li, H., Song, Y., Sun, L., Yuan, L.: Llava-cot: Let vision language models reason step-by-step. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2087–2098 (2025)

41. Yao, H., Yin, Q., Zhang, J., Yang, M., Wang, Y., Wu, W., Su, F., Shen, L., Qiu, M., Tao, D., Huang, J.: R1-shareVL: Incentivizing reasoning capabilities of multimodal large language models via share-GRPO. In: Advances in Neural Information Processing Systems (2025)
42. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., Sun, H., Su, Y., Chen, W.: 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 (2024)
43. Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Yue, Y., Song, S., Huang, G.: Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In: Advances in Neural Information Processing Systems (2025)
44. Zhang, R., Zhang, B., Li, Y., Zhang, H., Sun, Z., Gan, Z., Yang, Y., Pang, R., Yang, Y.: Improve vision language model chain-of-thought reasoning. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1631–1662 (2025). https://doi.org/10.18653/v1/2025.acl-long.82, https://aclanthology.org/2025.acl-long.82/
45. Zhang, Z., Zhang, A., Li, M., Zhao, H., Karypis, G., Smola, A.J.: Multimodal chain-of-thought reasoning in language models. Transactions on Machine Learning Research (2023)
46. Zheng, G., Yang, B., Tang, J., Zhou, H.Y., Yang, S.: DDCot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. In: Advances in Neural Information Processing Systems (2023), https://openreview.net/forum?id=ktYjrgOENR