I-RAVEN-X: Benchmarking Generalization and Robustness of Analogical and Mathematical Reasoning in Large Language and Reasoning Models

Giacomo Camposampiero
IBM Research – Zurich, ETH Zurich
giacomo.camposampiero1@ibm.com

Michael Hersche
IBM Research – Zurich
michael.hersche@ibm.com

Roger Wattenhofer
ETH Zurich
wattenhofer@ethz.ch

Abu Sebastian
IBM Research - Zurich
ase@zurich.ibm.com

Abbas Rahimi
IBM Research – Zurich
abr@zurich.ibm.com

Abstract

We introduce I-RAVEN-X, a symbolic benchmark designed to evaluate generalization and robustness in analogical and mathematical reasoning for Large Language Models (LLMs) and Large Reasoning Models (LRMs). I-RAVEN-X extends I-RAVEN by increasing operand complexity, attribute range, and introducing perceptual uncertainty. Compared to LLMs, empirical results on I-RAVEN-X show that LRMs achieve improved productivity and systematicity on longer reasoning relations and wider attribute ranges, respectively. For instance, LRMs experience a significantly smaller degradation on arithmetic accuracy ($80.5\% \rightarrow 63.0\%$) compared to LLMs ($59.3\% \rightarrow 4.4\%$). However, LRMs are still significantly challenged by reasoning under uncertainty (-61.8% in task accuracy) and cannot effectively explore multiple probabilistic outcomes in superposition.

1 Introduction

Abstract reasoning is often regarded as a core feature of human intelligence. A wide range of benchmarks to assess abstract reasoning has been proposed in the past decade [1–4]. Raven's Progressive Matrices (RPM) [5, 6], a task associating vision with relational and analogical reasoning in a hierarchical representation, is one of the most prominent of them thanks to its extensive use to benchmark for abstract reasoning, analogy-making, and out-of-distribution (OOD) testing [7–11]. RAVEN [11] represented the first attempts to build an automatically-generated dataset of RPM samples, enabling large-scale training of ML methods. I-RAVEN [8] improved RAVEN, proposing a new generation algorithm to avoid shortcut solutions that were possible in the original dataset. However, RAVEN and I-RAVEN exhibit some limitations that hinder their reliability as benchmarks for reasoning proficiency in LLMs and LRMs. Firstly, most of the problems involve only a few operands, representing an overly simplistic subset of analogical and mathematical relations. Most importantly, the test problems and their corresponding solutions are openly available online, increasing the risk of potential data leakage from the model's pre- and post-training stages as previously observed in other settings [12]. Furthermore, assuming the availability of an *oracle perception* has become a standard practice in their translation from visual to textual (symbolic) tasks (necessary to test language-only models) [13–15]. This assumption is reasonable when the scope of the investigation is limited to the reasoning component; however, it falls short when we zoom out to more complex, end-to-end systems, as it bypasses crucial steps of the original visual analogical reasoning, such as filtering irrelevant attributes and accounting for the uncertainty of the perception module.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

To tackle these problems, this paper makes the following contributions:

1. introduces I-RAVEN-X, an enhanced, symbolic version of the standard I-RAVEN benchmark that enables testing the generalization and robustness to simulated perceptual uncertainty in text-based language and reasoning models (see Figure 1),
2. highlights that LRMs consistently generalize better than LLMs in terms of productivity and systematicity, but significantly fail to reason under uncertainty.

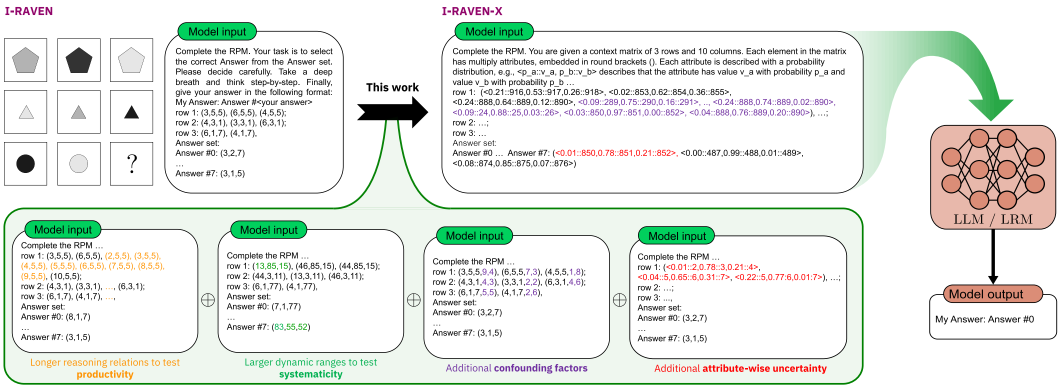
2 Methods

2.1 I-RAVEN-X: testing generalization and robustness of reasoning in LLMs and LRMs

We propose a fully-symbolic, parametrizable dataset to evaluate LLMs and LRMs, dubbed I-RAVEN-X. Some examples from the dataset are included in Figure 1. I-RAVEN-X enhances the original I-RAVEN (more extensively described in Appendix A) over different dimensions:

1. **Productivity**: we parametrize the number of operands in the reasoning relations (e.g., using 3×10 matrices instead of 3×3, 🟧 in Figure 1);
2. **Systematicity**: we introduce larger dynamic ranges for the operand values (e.g., 1000 attribute values instead of 10, 🟩 in Figure 1);
3. **Robustness to confounding factors**: we augment the set of original attributes in RPM with randomly sampled values, which do not contribute to the reasoning (🟪 in Figure 1);
4. **Robustness to non-degenerate value distributions**: we smoothen the distributions of the input values corresponding to the generative factors (🟥 in Figure 1).

Practically speaking, 1. and 2. enable testing the generalization of LLMs and LRMs to longer reasoning chains and an increased number of concepts. On the other hand, 3. and 4. allow to loosen the strong assumption of an oracle perception, simulating an imperfect sensory front-end while operating with text-only language models, hence providing an additional focus on the robustness of reasoning under uncertainty. More details on the design of $1-4$ are included in Appendix B. In addition, the original I-RAVEN was narrowed down to a single constellation (`center`, containing a single object per panel), which was observed to be simultaneously a strong test for a wide range of logical and arithmetic skills and unexpectedly challenging.



```
Complete the RPM. Your task is to select the correct Answer from the Answer set. Please decide carefully. Take a deep breath and think step-by-step. Finally, give your answer in the following format:
My Answer: Answer #<your answer>
row 1: (3,5,5), (6,5,5), (4,5,5);
row 2: (4,3,1), (3,3,1), (6,3,1);
row 3: (6,1,7), (4,1,7),
Answer set:
Answer #0: (3,2,7)
…
Answer #7: (3,1,5)
```

```
Complete the RPM. You are given a context matrix of 3 rows and 10 columns. Each element in the matrix has multiply attributes, embedded in round brackets (). Each attribute is described with a probability distribution, e.g., <p_a::v_a, p_b::v_b> describes that the attribute has value v_a with probability p_a and value v_b with probability p_b …
row 1:  (<0.21::916,0.53::917,0.26::918>, <0.02::853,0.62::854,0.36::855>,
<0.24::888,0.64::889,0.12::890>, <0.09::289,0.75::290,0.16::291>, .., <0.24::888,0.74::889,0.02::890>,
<0.09::24,0.88::25,0.03::26>, <0.03::850,0.97::851,0.00::852>, <0.04::888,0.76::889,0.20::890>), …;
row 2: …;
row 3: …
Answer set:
Answer #0 …  Answer #7: (<0.01::850,0.78::851,0.21::852>, <0.00::487,0.99::488,0.01::489>,
<0.08::874,0.85::875,0.07::876>)
```

```
Complete the RPM …
row 1: (3,5,5), (6,5,5), (2,5,5), (3,5,5),
(4,5,5), (5,5,5), (6,5,5), (7,5,5), (8,5,5),
(9,5,5), (10,5,5);
row 2: (4,3,1), (3,3,1), …, (6,3,1);
row 3: (6,1,7), (4,1,7), …,
Answer set:
Answer #0: (8,1,7)
…
Answer #7: (3,1,5)
```

```
Complete the RPM …
row 1: (13,85,15), (46,85,15), (44,85,15);
row 2: (44,3,11), (13,3,11), (46,3,11);
row 3: (6,1,77), (4,1,77),
Answer set:
Answer #0: (7,1,77)
…
Answer #7: (83,55,52)
```

```
Complete the RPM …
row 1: (3,5,5,9,4), (6,5,5,7,3), (4,5,5,1,8);
row 2: (4,3,1,4,3), (3,3,1,2,2), (6,3,1,4,6);
row 3: (6,1,7,5,5), (4,1,7,2,6),
Answer set:
Answer #0: (3,2,7)
…
Answer #7: (3,1,5)
```

```
Complete the RPM …
row 1: (<0.01::2,0.78::3,0.21::4>,
<0.04::5,0.65::6,0.31::7>, <0.22::5,0.77:6,0.01:7>), …;
row 2: …;
row 3: ...,
Answer set:
Answer #0: (3,2,7)
…
Answer #7: (3,1,5)
```

Figure 1: This figure highlights all the different axes of generalization and robustness to uncertainty, which I-RAVEN-X stresses. Compared to standard I-RAVEN **(a)**, I-RAVEN-X **(b)** involves more panels per row (10 vs. 3) and larger attribute dynamic ranges (up to 100× more values per attribute). In addition, it is possible to introduce uncertainty in the reasoning process through confounders (such as panels' background and color patterns within objects) and smoothening the attribute values' distributions (displayed on the right for the panel in position $(1,10)$) **(c)**. We adopt a visual representation of the panels and their attributes for clarity of explanation; in practice, however, our dataset is purely symbolic and has not been extended yet to the visual domain.

2.2 Models and prompting techniques

We focus our study on two state-of-the-art (SOTA) LRMs (the closed-source OpenAI o3-mini model and the open-source DeepSeek R1 model [16], together with its distilled version based on Llama 70B) and LLMs (the proprietary GPT-4 [17] and the open-source Llama-3 70B [18]). A more precise accounting of the version of the model used, along with additional details on the prompting engineering techniques adopted in our experiments, is presented in Appendix C. In Appendix D, we include an additional comparison between o3-mini and its predecessor, o1.

3 Results

In this section, we evaluate the generalization and robustness of the analogical and arithmetic reasoning capabilities of LRMs and LLMs using I-RAVEN and I-RAVEN-X. In particular, we aim to answer the following research questions: How well do the analogical and mathematical capabilities of LLMs and LRMs generalize in terms of productivity and systematicity **(Q1)**? How robust are LLMs and LRMs when confronted with reasoning under uncertainty **(Q2)**?

3.1 LRMs are stronger analogical and mathematical reasoners than LLMs

To answer Q1, we benchmark the productivity and systematicity of the models introduced in Section 2.2 on I-RAVEN and I-RAVEN-X. Table 1 reports the results of the evaluation. We observe that LRMs are much stronger reasoners than LLMs when challenged with the longer reasoning rules and attribute ranges in I-RAVEN-X, especially when we consider mathematical reasoning (additive relations). While LLMs show a massive drop in arithmetic accuracy on I-RAVEN-X, nearing 0% for comparable prompt complexity, LRMs are affected by a much smaller arithmetic degradation on average. These marked gains in arithmetic reasoning performance, with improvements reaching up to 65.4% in certain settings, suggest that LRMs can more comfortably identify and generalize (in productivity and systematicity) mathematical rules compared to LLMs.

Moreover, we can also see that LRMs achieve reasoning accuracy on par with LLMs despite using significantly less engineered prompts; when the prompt complexity is comparable, on the other hand, LRMs consistently outperform LLMs on the investigated benchmarks. o3-mini, for instance, shows no drops in accuracy on I-RAVEN-X and a 6% drop on I-RAVEN compared to GPT-4 while using only 1⁄21 of the prompts. When we compare the same two models on similar prompt complexities (that is, using entangled prompting in both settings, but still retaining a 1:7 ratio between LRMs and LLMs due to self-consistency), o3-mini emerges as a clear winner, showing an average 6.5% increase in accuracy. These results suggest that LRMs do not require incorporating as much inductive bias (through prompt engineering) as LLMs do, and that the production of "thinking" tokens generally makes the reasoning process more robust.

			I-RAVEN (3×3)			**I-RAVEN-X** (3×10)					
Model	**ICL**	**Prompts**	**Range 10**			**Range 100**			**Range 1000**		
			Task	Arithm.	Tok.	Task	Arithm.	Tok.	Task	Arithm.	Tok.
Llama-3 70B	✓	21	85.0	45.0	21	73.0	2.6	21	74.2	0.4	21
GPT-4	✗	21	93.2	73.6	21	79.6	25.1	21	76.6	8.4	21
Llama-3 70B	✓	7	79.0	31.0	21	72.6	0.0	21	74.0	0.4	21
GPT-4	✗	7	74.8	27.2	21	72.8	2.7	21	74.0	1.1	21
OpenAI o3-mini (med.)	✗	1	86.6	74.4	5445	77.6	53.2	7884	81.0	60.8	7209
OpenAI o3-mini (high)	✗	1	92.6	86.1	9867	82.4	63.5	19041	80.6	60.1	19449
DeepSeek R1	✗	1	80.6	74.8	4486	84.0	67.7	5550	82.8	65.8	5505
DeepSeek R1 dist.	✗	1	78.4	69.4	5192	67.0	52.9	6690	72.0	54.4	6324

Table 1: Full task accuracy (% of test examples correctly predicted) and arithmetic accuracy (% of the attributes in the test examples governed by an arithmetic relation correctly predicted) of LLMs and LRMs on I-RAVEN and I-RAVEN-X. We report if In-Context Learning (ICL) examples of the task were added to the prompt, the number of total prompts fed into the model (some techniques, such as self-consistency and disentangled prompting require querying the model multiple times), and the number of tokens generated by the model. "Range" indicates the dynamic range of the attributes' values. "Tok." indicates the average number of output tokens of the model.

3.2 LRMs are significantly challenged by reasoning under uncertainty

The results in Section 3.1 show that LRMs can solve analogical and mathematical reasoning tasks more accurately than LLMs. However, would they be capable of retaining the same robustness in scenarios where uncertainty is introduced (Q2)? To answer this question, we benchmark two LRMs with I-RAVEN-X with uncertainty as proposed in Section 2.1. Due to the failures shown in the previous section, we do not consider LLMs for these experiments. The empirical results of this study are reported in Table 2.

Firstly, we observe that LRMs perform significantly worse when noise factors that simulate perceptual uncertainty are integrated into the experiments. For instance, o3-mini's accuracy dropped by 11.2% and 15.2% on task and arithmetic accuracy, respectively, when evaluated with 10 additional confounding attributes. R1, on the other hand, is more robust to confounders (5.8% and 12.2% drops on task and arithmetic accuracy), but it performs much worse when the attribute values' distributions are smoothened, losing up to 19.8% of task accuracy in the harshest scenario. o3-mini shows a much smaller degradation (5.4%) in this setting.

When both the confounders and distribution smoothening are evaluated together at their maximum level, we observe a sharp drop in task accuracy for both o3-mini (to 17.0%) and DeepSeek R1 (to 22.8%), bringing them close to random chance (12.5%). Different causes could play a role in this significant degradation: an increasing complexity of the prompts, which might impair the model's ability to detect and mimic patterns in the input, or a more general limitation in modeling probabilistic reasoning that requires maintaining coherence across multiple uncertain variables.

			OpenAI o3-mini			**DeepSeek R1**		
Experiment	**Confounders (SNR)**	$\mathbf{p_L}$	Task	Arith.	Tokens	Task	Arith.	Tokens
	0 (∞)	1.00	81.0	60.8	7209	82.8	65.8	6324
(a)	1 (4.77)	1.00	76.0	53.2	11521	78.2	55.2	8919
	3 (0.00)	1.00	75.6	51.7	11669	80.2	58.2	8429
	5 (-2.22)	1.00	71.2	48.3	12640	78.6	55.9	8681
	10 (-5.23)	1.00	69.8	45.6	13709	77.0	53.6	8912
(b)	0 (∞)	0.70	75.0	51.7	13112	67.4	44.9	6995
	0 (∞)	0.51	75.6	53.2	13028	63.0	46.4	7518
(c)	10 (-5.23)	0.51	17.0	41.1	18482	23.2	45.3	7147

Table 2: Task and arithmetic accuracy (%) on I-RAVEN-X (range [0,1000]) with different numbers of confounders, from 0 (no confounders, SNR=∞) to 10 (SNR=-5.23 dB), and different attributes' distribution smoothening (bin-smoothening strategy, with different probabilities assigned to the correct value bin p_L). We show experiments with: a) only confounders; b) only the attributes' distribution smoothing; c) both confounders and distribution smoothing. We report the number of output tokens to quantify the reasoning effort adopted on average by the model to find a solution.

4 Conclusion

This work presents I-RAVEN-X, a novel, symbolic benchmark for testing the generalization and robustness of analogical and mathematical reasoning. I-RAVEN-X is then used to evaluate these capabilities in LLMs and LRMs. Compared to LLMs, LRMs achieve improved productivity and systematicity on longer reasoning relations and wider attribute ranges, respectively. For instance, LRMs experience a significantly smaller degradation on arithmetic accuracy ($80.5\% \rightarrow 63.0\%$) compared to LLMs ($59.3\% \rightarrow 4.4\%$). However, LRMs are still significantly challenged by reasoning under uncertainty (-61.8% in task accuracy) and cannot explore multiple probabilistic outcomes at the same time. One limitation of our work consists of exploring the causal relationship between reasoning under uncertainty, prompt efficiency, and reasoning accuracy; further investigating this relation is left for future work. The dataset and the experiments' code will be released upon acceptance.

References

[1] Warren B. Bilker, John A. Hansen, Colleen M. Brensinger, Jan Richard, Raquel E. Gur, and Ruben C. Gur. Development of abbreviated nine-item forms of the Raven's standard progressive matrices test. *Assessment*, 19(3):354–369, September 2012. ISSN 1552-3489. doi: 10.1177/1073191112446655.

[2] Anoop Cherian, Kuan-Chuan Peng, Suhas Lohit, Kevin A. Smith, and Joshua B. Tenenbaum. Are Deep Neural Networks SMARTer Than Second Graders? In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10834–10844, Vancouver, BC, Canada, June 2023. IEEE. ISBN 9798350301298. doi: 10.1109/CVPR52729.2023.01043. URL `https://ieeexplore.ieee.org/document/10204961/`.

[3] François Chollet. On the Measure of Intelligence. *arXiv preprint arXiv.1911.01547*, November 2019. URL `http://arxiv.org/abs/1911.01547`.

[4] Yannick Niedermayr, Luca A. Lanzendörfer, Benjamin Estermann, and Roger Wattenhofer. RLP: A reinforcement learning benchmark for neural algorithmic reasoning, 2024. URL `https://openreview.net/forum?id=pYmQId95iR`.

[5] J.C. Raven, J.H. Court, and J. Raven. *Raven's progressive matrices*. Oxford Psychologists Press, 1938.

[6] Patricia A Carpenter, Marcel Adam Just, and Peter Shell. What One Intelligence Test Measures: A Theoretical Account of the Processing in the Raven Progressive Matrices Test. *Psychological review*, 97(3):404, 1990.

[7] Yaniv Benny, Niv Pekar, and Lior Wolf. Scale-Localized Abstract Reasoning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12552–12560, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.01237. URL `https://ieeexplore.ieee.org/document/9577474/`.

[8] Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. Stratified Rule-Aware Network for Abstract Visual Reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16248`.

[9] Mikołaj Małkiński and Jacek Mańdziuk. Deep Learning Methods for Abstract Visual Reasoning: A Survey on Raven's Progressive Matrices. *ACM Comput. Surv.*, 57(7):166:1–166:36, February 2025. ISSN 0360-0300. doi: 10.1145/3715093. URL `https://dl.acm.org/doi/10.1145/3715093`.

[10] Melanie Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021. ISSN 1749-6632. doi: 10.1111/nyas.14619. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/nyas.14619`.

[11] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. RAVEN: A Dataset for Relational and Analogical Visual REasoNing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. URL `https://ieeexplore.ieee.org/document/8953364/`.

[12] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2025. URL `https://openreview.net/forum?id=AjXkRZIvjB`.

[13] Taylor Webb, Keith J. Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541, July 2023. ISSN 2397-3374. doi: 10.1038/s41562-023-01659-w. URL `https://www.nature.com/articles/s41562-023-01659-w`.

[14] Xiaoyang Hu, Shane Storks, Richard Lewis, and Joyce Chai. In-Context Analogical Reasoning with Pre-Trained Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1953–1969, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.109. URL `https://aclanthology.org/2023.acl-long.109`.

[15] Michael Hersche, Giacomo Camposampiero, Roger Wattenhofer, Abu Sebastian, and Abbas Rahimi. Towards Learning to Reason: Comparing LLMs with Neuro-Symbolic on Arithmetic Relations in Abstract Reasoning. In *AAAI Workshop on Neural Reasoning and Mathematical Discovery – An Interdisciplinary Two-Way Street (NEURMAD)*, March 2025. URL `https://openreview.net/pdf?id=F90YOOMacL`.

[16] DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv.2501.12948*, January 2025. doi: 10.48550/arXiv.2501.12948. URL `http://arxiv.org/abs/2501.12948`.

[17] OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, March 2024. URL `http://arxiv.org/abs/2303.08774`.

[18] Meta. The Llama 3 Herd of Models. *arxiv preprint arXiv:2407.21783*, August 2024. doi: 10.48550/arXiv.2407.21783. URL `http://arxiv.org/abs/2407.21783`.

[19] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114v1*, 2013. doi: 10.48550/arXiv1312.6114v1. URL `https://arxiv.org/abs/1312.6114v1`.

[20] Melanie Mitchell, Alessandro B. Palmarini, and Arseny Moskvichev. Comparing Humans, GPT-4, and GPT-4V On Abstraction and Reasoning Tasks. In *AAAI 2024 Workshop on ”Are Large Language Models Simply Causal Parrots?”*, 2024. URL `https://openreview.net/forum?id=3rGT5OkzpC`.

[21] Yifan Jiang, Jiarui Zhang, Kexuan Sun, Zhivar Sourati, Kian Ahrabian, Kaixin Ma, Filip Ilievski, and Jay Pujara. MARVEL: Multidimensional Abstraction and Reasoning through Visual Evaluation and Learning. In *The Thirty-eight Conference on Neural Information Processing Systems (NeurIPS)*, 2024. URL `https://openreview.net/pdf?id=vecFROHnL4`.

[22] Xu Cao, Bolin Lai, Wenqian Ye, Yunsheng Ma, Joerg Heintz, Jintai Chen, Jianguo Cao, and James M. Rehg. What is the Visual Cognition Gap between Humans and Multimodal LLMs? *arXiv preprint arXiv:2406.10424*, June 2024. URL `http://arxiv.org/abs/2406.10424`.

[23] Kian Ahrabian, Zhivar Sourati, Kexuan Sun, Jiarui Zhang, Yifan Jiang, Fred Morstatter, and Jay Pujara. The Curious Case of Nonverbal Abstract Reasoning with Multi-Modal Large Language Models. In *First Conference on Language Modeling (COLM)*, 2024. URL `https://openreview.net/forum?id=eDWcNqiQWW`.

[24] Yizhe Zhang, He Bai, Ruixiang Zhang, Jiatao Gu, Shuangfei Zhai, Joshua M. Susskind, and Navdeep Jaitly. How Far Are We from Intelligent Visual Deductive Reasoning? In *ICLR 2024 Workshop: How Far Are We From AGI*, May 2024. URL `https://openreview.net/forum?id=AMrYF9F3J6`.

[25] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[26] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving Quantitative Reasoning Problems With Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 3843–3857, 2022. URL `https://openreview.net/forum?id=IFXTZERXdM7`.

[27] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler,

Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

[28] Antonia Wüst, Tim Tobiasch, Lukas Helff, Devendra S. Dhami, Constantin A. Rothkopf, and Kristian Kersting. Bongard in Wonderland: Visual Puzzles that Still Make AI Go Mad? In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*, October 2024. URL `https://openreview.net/pdf?id=4Yv9tFHDwX`.

A I-RAVEN dataset

Raven's progressive matrices (RPM) is a visual task that involves perceiving pattern continuation and elemental abstraction as well as deducing relations based on a restricted set of underlying rules in a process that mirrors the attributes of advanced human intelligence. In this work, we focus on the I-RAVEN dataset. Each RPM test in I-RAVEN is an analogy problem presented as a 3×3 pictorial matrix of context panels. Every panel in the matrix is filled with several geometric objects based on a certain rule, except the bottom-right panel, which is left blank. Figure 2 includes an I-RAVEN example test. The task is to complete the missing panel by picking the correct answer from a set of (eight) candidate answer panels that match the implicit generation rule on every attribute. The object's attributes (color, size, shape, number, position) are governed by individual underlying rules:

- *constant*, the attribute value does not change per row;
- *arithmetic*, the attribute value of the third panel corresponds to either the sum or the difference of the first two panels of the row;
- *progression*, the attribute value monotonically increases or decreases in a row by 1 or 2;
- *distribute three*, the set of the three different values remains constant across rows, but the individual attribute values get shifted to the left or to the right by one position at every row; it also holds column-wise.

Each panel contains a variable number of objects (minimum one, maximum nine) arranged according to one of seven different constellations (center, distribute-four, distribute-nine, left-right, up-down, in-out-center, and in-out-four).

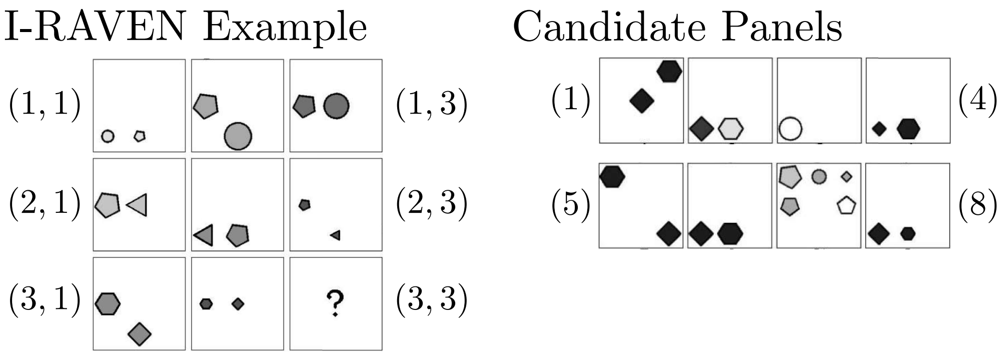


Figure 2: RPM example from I-RAVEN.

B I-RAVEN-X implementation details

B.1 Productivity and Systematicity

Our new benchmark maintains I-RAVEN's four rules and three attributes but allows for a parameterizable number of columns (g) and a dynamic range of attribute values (m). When generating a new RPM example, we uniformly sample from one of the available rules (`constant`, `progression`, `arithmetic`, and `distribute three`). Note that the attribute `shape` does not incur the `arithmetic` rule.

In the following, we describe the generation process of the RPM context matrix of size $3 \times g$ for the individual rules. The overall goal is that the values stay in the range $[0, m-1]$.

- `constant`: For each row, we uniformly sample an integer from the set $\{0, 1, ..., m-1\}$, and duplicate along the row.

- `progression`: First, we uniformly sample the progressive increment/decrement (δ) from the set $\{-2, -1, +1, +2\}$. In case of a positive increment, we first define the values of the right-most columns, by uniformly sampling from the set $\{(g-1) \cdot \delta, ..., m-1\}$ for each row. Then, the rest of the matrix is completed by applying the progression rule. The sampling for a negative δ is done specularly from the first column.
- `arithmetic`: The attribute values of the first $g-1$ panels are either added (`arithmetic plus`) or subtracted (`arithmetic minus`), yielding the attribute value of the last panel in the row. In `arithmetic plus`, we sequentially sample the values from the first $g-1$ panels in the row. For each panel, we set the sampling range to $\{0, ..., m-s\}$, where s is the sum of the already sampled panels in the row. Afterward, the first $g-1$ panels are shuffled. Finally, the values of the last panels are the sum of the first $g-1$ ones, applied row-wise. For `arithmetic minus`, we apply the same sampling strategy but leave the first column empty. The value of the first column is then defined as the sum of the other columns.
- `distribute-n`: We uniformly sample distinct values for the first row from $\{0, ..., m-1\}$. The content of the remaining rows is defined by applying a circular shift per row (either right or left).

Finally, we generate the candidate answers using I-RAVEN's attribute bisection tree [8]. The original RAVEN dataset had a flaw in the generation of the answer set. Each distractor in the answer set (i.e., a wrong answer candidate) was generated by randomly altering one attribute of the correct answer. As a result, one could predict the correct answer by taking the mode of the answer candidates without looking at the context matrix, therefore bypassing the actual reasoning task. As a remedy, the attribute bisection tree generates unbiased answers that are well-balanced.

B.2 Confounding attributes

Confounding attributes represent properties and patterns that can be extracted from the visual inputs by a front-end perception module but are not relevant to the reasoning process. This could be the case, for instance, when the attributes are extracted by unsupervised vision models such as Variational Autoencoders [19] or even a multi-modal LLM that is prompted to extract the attributes. In Figure 1, confounding attributes are represented by the background of the input panels and the color patterns, which sometimes appear inside the objects. In I-RAVEN-X, we integrate confounders by extending the set of original attributes of each panel with an arbitrary number of additional attributes uniformly sampled in the interval $[0, m-1]$, where m is the range of the attributes' values. For large enough m, the probability of sampling values that fit a valid rule is negligible, and hence, confounders do not introduce ambiguities in the choice of the answer panel. However, they linearly reduce the signal-to-noise ratio (SNR) in the reasoning process, requiring models to implement strategies to filter out noisy input components.

B.3 Smooth attribute values' distributions

We deviate from the original I-RAVEN degenerate attributes' distributions and introduce variance, which allows us to test the robustness of the models when reasoning with uncertain attribute values. In practice, we smoothen the original attributes' distributions using a three-bins strategy, where the probability of the true value T is $p(T) \sim \mathcal{U}(p_L, 1), p_L > 0.5$ and the probabilities of its two neighboring values are $p(N_1) \sim \mathcal{U}(0, 1-p(T))$ and $p(N_2) = 1 - p(T) - p(N_1)$. Note that the motivation behind the three-bins strategy is to introduce variance with minimal additional cost for LRMs' prompt complexity.

C Models and prompting details

LLMs We focused our evaluations on text-only LLMs. There exist attempts [20–24] that leverage vision support of some multi-modal LLMs (e.g., GPT-4V) directly feeding the models with visual RPM data; however, they achieve consistently lower reasoning performance than with text-only prompting. The SOTA LLM-based abstract reasoning approach [14] relied on reading out GPT-3's (`text-davinci-002`) token probabilities. However, this model is no longer accessible to users and its successive iterations do not allow the retrieval of prediction logits. Hence, we considered discrete classification approaches that are based on output strings rather than distribution over tokens. In particular, we investigated two SOTA LLMs: the proprietary GPT-4 [17][1] (`gpt-4-0613`) and the open-source Llama-3 70B [18][2]. During initial tests, GPT-4o yielded worse results than GPT-4, hence we focused on GPT-4. Different prompting engineering techniques were integrated to improve the overall accuracy of these models:

- **Disentangled prompting**, a compositionally structured approach that queries the LLM for individual attribute prediction. Disentangled prompting simplifies the task, but increases the number of queries by $3\times$ (where 3 is, in this case, the number of attributes). In our experiments, disentangled prompts were only used for some experiments in LLMs, increasing the number of prompts from 7 to 21.
- **Self-consistency** [25, 26], which consists in querying the model multiple times ($n = 7$ times), sampling the next token from the distribution with a non-zero soft-max temperature. We find the optimal soft-max temperature for GPT-4 ($T = 0.5$) and Llama-3 70 B ($T = 0.4$) via a grid search on a subset of 50 I-RAVEN problems. We did not explore the effect of other parameters, such as top-k or top-p, and set them to the default values. The final prediction is determined by a majority vote over the sampled outputs. The selection of an odd number of samples (i.e., $n = 7$) helps to prevent potential ties.
- **In-context learning**: for a better understanding of the RPM task, we optionally prefix 16 in-context examples to the prompt [27]. In the predictive classification approach (where no answer candidates are provided), we simply provide complete example RPM matrices. The in-context samples are randomly selected from I-RAVEN's training set. Examples that had the same context matrix as the actual task are discarded and re-sampled to prevent shortcut solutions.

LRMs For the OpenAI o3-mini model, we use the `o3-mini-2025-01-31` via the OpenAI API. By default, reasoning efforts were set to `medium` and the number of reasoning tokens to 25,000. For DeepSeek R1 model, the full model with 671B parameters was serviced by `www.together.ai`, whereas the distilled version was run locally on 8 NVIDIA A100 GPUs. The maximum number of reasoning tokens was set to 25,000, the temperature to 0.6, and top-p to 0.7. Self-consistency [25, 26] and attributes' scaling [14] were not used in experiments with LRMs. Moreover, no in-context examples of the tasks [27] were provided since they were previously observed to be hurtful for LRMs [16]. We also restrict the investigation to a subset of 500 randomly sampled RPM tests in both I-RAVEN and I-RAVEN-X (due to budget constraints), which we observed to be representative enough of the entire test set [14]. We report some examples of the prompts used in our experiments in Tables 3, 4, 5, and 6. The prompting style for embracing CoT was inspired by [28]. For automatic retrieval of the model's answer, we prompt it to provide its answer in the format "My Answer: Answer #<your answer>". By default, answer panel #0 is predicted if no answer can be retrieved. Contrary to LLMs, all the empirical results reported for LRMs are obtained using entangled prompts.

[1] GPT-4 was accessed between 07/03/2024–10/30/2024.

[2] The model weights were downloaded and evaluated locally. Unless stated otherwise, we use the base model without instruction tuning.

Some examples of I-RAVEN and I-RAVEN-X prompts used for LRMs are reported in the following Tables.

Complete the Raven’s progressive matrix. Your task is to select the correct Answer from the Answer set. Please decide carefully. Take a deep breath and think step-by-step. Finally, give your answer in the following format: My Answer: Answer #<your answer>

row 1: (3,5,5), (6,5,5), (4,5,5);
row 2: (4,3,1), (3,3,1), (6,3,1);
row 3: (6,1,7), (4,1,7),

Answer set:
Answer #0: (3,2,7)
Answer #1: (7,1,5)
Answer #2: (7,2,5)
Answer #3: (7,2,7)
Answer #4: (7,1,7)
Answer #5: (3,1,7)
Answer #6: (3,2,5)
Answer #7: (3,1,5)

Table 3: Example prompt for an I-RAVEN task.

Complete the Raven’s progressive matrix. Your task is to select the correct Answer from the Answer set. Please decide carefully. Take a deep breath and think step-by-step. Finally, give your answer in the following format: My Answer: Answer #<your answer>

row 1: (6,16,9), (7,15,9), (70,14,9), (93,13,9), (88,12,9), (77,11,9), (83,10,9), (22,9,9), (39,8,9), (27,7,9);
row 2: (7,12,24), (70,11,24), (93,10,24), (88,9,24), (77,8,24), (83,7,24), (22,6,24), (39,5,24), (27,4,24), (6,3,24);
row 3: (70,35,52), (93,34,52), (88,33,52), (77,32,52), (83,31,52), (22,30,52), (39,29,52), (27,28,52), (6,27,52),

Answer set:
Answer #0: (7,26,52)
Answer #1: (83,55,52)
Answer #2: (7,26,37)
Answer #3: (83,55,37)
Answer #4: (7,55,52)
Answer #5: (83,26,37)
Answer #6: (7,55,37)
Answer #7: (83,26,52)

Table 4: Example prompt for an I-RAVEN-X task.

Complete the Raven’s progressive matrix. Your task is to select the best matching Answer from the Answer set. Please decide carefully. Take a deep breath and think step-by-step. Finally, give your answer in the following format: My Answer: Answer #<your answer>

row 1: (917,854,889,837,449,40,616,988,225,603,813,154,860), (290,853,889,310,920,885,291,416,926,503,379,786,859), (532,852,889,336,540,95,33,182,41,215,990,859,625), (25,851,889,948,465,970,253,795,956,622,323,735,535), (31,850,889,846,149,643,802,187,413,101,300,378,181), (43,849,889,700,975,580,488,662,820,977,189,160,955), (574,848,889,484,18,951,173,279,247,567,639,939,730), (761,847,889,971,245,547,175,991,94,306,976,778,188), (576,846,889,547,182,955,995,410,545,537,859,368,146), (291,845,889,544,515,965,647,155,660,835,167,363,578);
row 2: (290,898,875,416,729,621,255,121,775,992,332,824,69), (532,897,875,617,602,91,626,959,328,566,572,496,129), (25,896,875,507,14,482,3,638,723,822,326,152,311), (31,895,875,551,141,165,894,867,142,856,245,396,325), (43,894,875,645,712,987,788,382,795,149,295,457,63), (574,893,875,269,762,290,698,804,252,56,328,850,702), (761,892,875,621,590,319,785,4,122,627,517,924,88), (576,891,875,268,299,764,678,718,860,626,845,523,1), (291,890,875,860,69,712,754,590,214,674,171,773,227), (917,889,875,802,908,433,515,585,256,102,529,939,585);
row 3: (532,497,831,73,406,82,149,646,932,466,196,966,172), (25,496,831,76,880,109,467,76,845,392,673,736,51), (31,495,831,79,825,847,494,174,270,472,649,164,234), (43,494,831,39,960,182,917,180,643,977,698,321,467), (574,493,831,553,583,258,422,840,680,109,870,539,289), (761,492,831,481,548,81,43,180,359,410,733,702,708), (576,491,831,882,329,883,287,624,816,453,120,316,349), (291,490,831,398,434,521,426,600,224,181,827,281,512), (917,489,831,611,791,841,260,28,125,408,122,577,903),

Answer set:
Answer #0: (290,488,875,657,175,669,825,660,980,305,71,297,764)
Answer #1: (851,488,875,785,95,663,714,937,607,543,958,80,215)
Answer #2: (290,451,831,808,72,151,7,665,312,920,665,806,177)
Answer #3: (290,488,831,340,114,819,129,10,922,744,948,540,925)
Answer #4: (851,451,875,714,337,713,987,115,520,218,644,222,463)
Answer #5: (851,488,831,948,251,490,394,977,846,124,951,827,501)
Answer #6: (290,451,875,761,816,59,950,670,732,542,237,552,272)
Answer #7: (851,451,831,9,552,304,979,949,86,118,847,82,575)

Table 5: Example prompt for the I-RAVEN-X task with confounders.

Complete the Raven’s progressive matrix. You are given a context matrix of 3 rows and 10 colums. Each element in the matrix has multiply attributes, embedded in round brackets (). Each attribute is described with a probability distribution, e.g., <p_a::v_a, p_b::v_b> describes that the attribute has value v_a with probability p_a and value v_b with probability p_b. Your task is to select the best matching Answer from the Answer set. Please decide carefully. Take a deep breath and think step-by-step. Finally, give your answer in the following format: My Answer: Answer #<your answer>

row 1: (<0.21::916,0.53::917,0.26::918>, <0.02::853,0.62::854,0.36::855>, <0.24::888,0.64::889,0.12::890>), (<0.09::289,0.75::290,0.16::291>, <0.12::852,0.74::853,0.14::854>, <0.11::888,0.85::889,0.04::890>), (<0.44::531,0.55::532,0.01::533>, <0.36::851,0.63::852,0.01::853>, <0.24::888,0.74::889,0.02::890>), (<0.09::24,0.88::25,0.03::26>, <0.03::850,0.97::851,0.00::852>, <0.04::888,0.76::889,0.20::890>), (<0.08::30,0.58::31,0.34::32>, <0.02::849,0.97::850,0.01::851>, <-0.00::888,0.91::889,0.09::890>), (<0.20::42,0.51::43,0.29::44>, <0.01::848,0.97::849,0.02::850>, <0.25::888,0.70::889,0.05::890>), (<0.12::573,0.87::574,0.01::575>, <0.06::847,0.78::848,0.16::849>, <0.01::888,0.99::889,0.00::890>), (<0.04::760,0.82::761,0.14::762>, <0.08::846,0.70::847,0.22::848>, <0.04::888,0.77::889,0.19::890>), (<0.04::575,0.54::576,0.42::577>, <0.46::845,0.54::846,-0.00::847>, <0.01::888,0.91::889,0.08::890>), (<0.15::290,0.85::291,0.00::292>, <0.04::844,0.78::845,0.18::846>, <0.30::888,0.66::889,0.04::890>);
row 2: (<0.01::289,0.81::290,0.18::291>, <0.19::897,0.59::898,0.22::899>, <0.20::874,0.72::875,0.08::876>), (<0.07::531,0.82::532,0.11::533>, <0.37::896,0.54::897,0.09::898>, <-0.00::874,0.77::875,0.23::876>), (<0.12::24,0.72::25,0.16::26>, <0.01::895,0.78::896,0.21::897>, <0.34::874,0.66::875,-0.00::876>), (<0.19::30,0.74::31,0.07::32>, <0.20::894,0.61::895,0.19::896>, <0.00::874,0.99::875,0.01::876>), (<0.20::42,0.77::43,0.03::44>, <0.02::893,0.95::894,0.03::895>, <0.08::874,0.73::875,0.19::876>), (<0.05::573,0.85::574,0.10::575>, <0.08::892,0.91::893,0.01::894>, <0.06::874,0.81::875,0.13::876>), (<0.14::760,0.53::761,0.33::762>, <0.15::891,0.65::892,0.20::893>, <0.13::874,0.66::875,0.21::876>), (<0.05::575,0.65::576,0.30::577>, <0.01::890,0.82::891,0.17::892>, <0.12::874,0.66::875,0.22::876>), (<0.00::290,0.94::291,0.06::292>, <0.02::889,0.95::890,0.03::891>, <0.12::874,0.86::875,0.02::876>), (<0.14::916,0.84::917,0.02::918>, <0.02::888,0.95::889,0.03::890>, <0.01::874,0.54::875,0.45::876>);
row 3: (<0.21::531,0.77::532,0.02::533>, <0.01::496,0.88::497,0.11::498>, <0.07::830,0.62::831,0.31::832>), (<0.20::24,0.79::25,0.01::26>, <0.19::495,0.62::496,0.19::497>, <0.06::830,0.92::831,0.02::832>), (<0.17::30,0.56::31,0.27::32>, <0.27::494,0.64::495,0.09::496>, <0.02::830,0.98::831,0.00::832>), (<0.00::42,0.98::43,0.02::44>, <0.38::493,0.58::494,0.04::495>, <0.19::830,0.53::831,0.28::832>), (<0.07::573,0.52::574,0.41::575>, <0.01::492,0.99::493,0.00::494>, <0.01::830,0.81::831,0.18::832>), (<0.26::760,0.55::761,0.19::762>, <0.13::491,0.83::492,0.04::493>, <0.05::830,0.82::831,0.13::832>), (<0.47::575,0.52::576,0.01::577>, <0.15::490,0.59::491,0.26::492>, <0.16::830,0.81::831,0.03::832>), (<0.03::290,0.82::291,0.15::292>, <0.29::489,0.52::490,0.19::491>, <0.03::830,0.85::831,0.12::832>), (<0.08::916,0.81::917,0.11::918>, <0.05::488,0.83::489,0.12::490>, <0.09::830,0.64::831,0.27::832>),

Answer set:
Answer #0: (<0.06::289,0.83::290,0.11::291>, <0.00::487,1.00::488,0.00::489>, <0.03::874,0.82::875,0.15::876>)
Answer #1: (<0.01::850,0.78::851,0.21::852>, <0.00::487,0.99::488,0.01::489>, <0.08::874,0.85::875,0.07::876>)
Answer #2: (<0.03::289,0.57::290,0.40::291>, <0.15::450,0.75::451,0.10::452>, <0.15::830,0.62::831,0.23::832>)
Answer #3: (<0.06::289,0.52::290,0.42::291>, <0.03::487,0.92::488,0.05::489>, <0.31::830,0.61::831,0.08::832>)
Answer #4: (<0.02::850,0.95::851,0.03::852>, <0.16::450,0.63::451,0.21::452>, <0.20::874,0.52::875,0.28::876>)
Answer #5: (<0.02::850,0.86::851,0.12::852>, <0.18::487,0.80::488,0.02::489>, <0.14::830,0.79::831,0.07::832>)
Answer #6: (<0.01::289,0.96::290,0.03::291>, <0.38::450,0.59::451,0.03::452>, <0.08::874,0.68::875,0.24::876>)
Answer #7: (<0.08::850,0.62::851,0.30::852>, <0.15::450,0.82::451,0.03::452>, <0.09::830,0.87::831,0.04::832>)

Table 6: Example prompt for the I-RAVEN-X task with smooth distributions.

D Comparison between OpenAI o3-mini and o1

This Appendix presents a small ablation study on two different closed-source LRMs, OpenAI o1 and OpenAI o3-mini. The goal of these experiments was to measure the difference, if any, in the reasoning capabilities of the o3-mini model compared to its bigger, more expensive predecessor. We restricted the size of the test set to 100 test examples for both I-RAVEN and I-RAVEN-X. The results, presented in Table 7, show that the two models achieve roughly comparable performance on both I-RAVEN and I-RAVEN-X, with o3-mini being consistently slightly less accurate than o1. However, o1 is also considerably more expensive compared to o3: o1 is priced at $15 and $60 per million input and output tokens, respectively, while o3-mini costs only $1.1 and $4.4 per million input and output tokens (approximately $14\times$ less expensive). Hence, we opt to use only o3-mini in the full evaluation.

		I-RAVEN		**I-RAVEN-X**			
Model	**Setting**	**Range 10**		**Range 100**		**Range 1000**	
		Task	Arithm.	Task	Arithm.	Task	Arithm.
OpenAI o1	Entangled	88.0	79.7	86.0	68.2	86.0	68.2
OpenAI o3-mini	Entangled	86.6	81.4	84.0	63.6	81.0	60.8

Table 7: Task and arithmetic accuracy (%) comparison of two different LRMs on a subset of 100 test examples of I-RAVEN and I-RAVEN-X.