Dear editor and reviewers,

Thank you for your positive feedback on our rebuttal. We are especially pleased that both reviewers have recommended acceptance of our paper.

We have structured this second-round response in the same format as before: the reviewers' comments appear in black, our responses are in blue, and references to changes in the manuscript are in red. In the revised main text, significant modifications of this second revision are highlighted in green.

Review #1

I omit the summary of the paper, as this has not changed with respect to my prior review. In general, this new version of the paper conveys its contribution in a clearer and more understandable way; the purpose of the paper is also more clearly motivated. I have a few further comments (specially about the shape of the rules), but in general, I believe the paper can now be recommended for acceptance – but please clarify the exact form of the rules.

Thank you very much for your positive evaluation and for recommending our paper for acceptance. We appreciate your feedback and aim to address the remaining questions, particularly regarding the shape of the rules, in the detailed responses below.

--I find it still somewhat strange that a textual version of the (originally visual) RPM task is used. I understand that other approaches leverage vision support and achieve consistently lower performance. However, I miss an argument explaining using text-only data is still an interesting problem to consider, given that (as the paper already says) there exist verbal abstract reasoning benchmarks already. This could simply be a matter of adding one or two sentences in the Introduction.

Our motivation for using a textual version of the RPM task is to evaluate the reasoning capabilities of LLMs under conditions that play to their strengths, namely, language understanding. While prior work has shown that multimodal models struggle with visual abstract reasoning, we aimed to isolate the reasoning component by translating the task into a purely textual format. **Interestingly, even in their optimal setting, LLMs exhibit weaknesses in handling certain rule types (e.g., arithmetic), highlighting fundamental limitations in their reasoning abilities**. We believe these insights can inform future model design and, ultimately, contribute to improving multimodal reasoning performance as well.

We adjusted and extended the following part to the Introduction:

Circumventing the perception by providing ground-truth attribute labels to the models allows us to measure their analogical and mathematical reasoning capabilities in isolation. Hence, we evaluate the reasoning capabilities of LLMs under conditions that

play to their strengths, namely, language understanding, when such compositionally structured (i.e., disentangled) representations are provided.

--I am still not fully clear on what nxn constellations are. So, if I take what the paper says literally, there are 8 candidate answer panels; in the 2x2 constellation, each panel has 4 objects, is that correct? How are those objects related to the 3x3 context matrix?

Thank you for your question. It seems we have not sufficiently explained the overall task setup and the concept of constellations.

To clarify: the task consists of a 3x3 context matrix with 9 panels, where the bottom-right panel is missing. The task is to select a correct answer from 8 candidate panels to complete the 3x3 context matrix.

Independent of this setup is the **constellation**, which defines how objects are arranged within each panel. For example:

- In a 2x2 constellation, each panel can contain up to 4 objects arranged in a 2x2 grid.
- In a 3x3 constellation, each panel can contain up to 9 objects.
- In our work, we focus on the Center constellation, where each panel contains only a single object centered in the panel.

We chose the Center constellation because it already provides sufficient complexity to stress-test the reasoning capabilities of LLMs, while avoiding confounding factors introduced by more complex visual layouts.

We added the following sentences in Section 2.1 I-RAVEN:

The task is then to select the correct answer from eight candidate panels to complete the matrix. Independent of this setup, each panel contains a number of objects arranged according to a specific constellation.

--Section 4.1: it is still unclear what the Binding and Unbinding operators do. Could you provide some intuition (for example, as you already do with the Bundling operator?)

Thank you for pointing this out.

- Binding associates two elements, effectively encoding a relationship between two vectors. For example, binding the attribute “color” with the value “red” produces a new vector that represents this pair. Importantly, this operation destroys similarity: the result is dissimilar to both operands, which helps us to prevent interference between different bindings.
- Unbinding is the inverse operation. Given a bound pair and one of its components (e.g., the attribute), unbinding retrieves the other (e.g., the value). This allows for structured information retrieval.

We added this more explicit explanation of the binding and unbinding in Section 4.1.

--Page 8, line 34: should X and O be switched around? If I understood correctly, O should be the first two rows, and X should be the incomplete row.

Thank you for catching this typo. You are absolutely right: O should correspond to the first two rows and X to the incomplete third row.

We have corrected the typo in Section 4.2.

--I am still confused about the shape of the rules. What is equation (4) and why are there 12 distinct c_i's? Is this the most general form of the rule that the model learns? Why does it only mention two operators, even though 3 were defined?

--I am confused by line 14 in page 8 and Equation (5). It says "c_i either represents a context panel v_a^(i,j) or the identity". However, the definition of c_k makes no reference to the row i and column j, as far as I can see, so how to match c_i to a specific position in the context matrix?

Thank you for your questions. We address both points jointly below.

Equation (4) provides a generalized template for representing RPM rule application in the VSA space. It abstracts the reasoning process as a combination of binding and unbinding operations over a set of panel representations.

- The 12 distinct c_i terms correspond to 6 panels involved in the "positive" part of the rule (numerator) and the 6 panels in "negative" part (denominator). These panels are selected during learning and depend on the specific rule.
- Each c_i can either be a panel representation (e.g., x^k or o^l) or the identity vector e, depending on whether that position contributes to the rule.

While the notation c_i does not explicitly reference the row i and column j of the context matrix, **the mapping from c_i to specific positions is learned and stored during training**. For example, in the arithmetic plus rule, the model learns the following assignment:

c_1 = x^1
c_2 = x^2
c_i = e for all other i

This expresses that the two panels in the row are combined via binding.

Regarding the operators: Equation (4) uses only binding and unbinding because these are sufficient to express the arithmetic-like transformations underlying most RPM rules. Bundling is used separately to represent probability mass functions (PMFs) in distributed representation via weighted superpositions and is not required in the rule execution template.

For further clarity, Table D.7 in the appendix provides all the learned assignments for different rule types.

Table D.7

I-RAVEN rules programmed in $\text{ARLC}_{\text{progr}}$.

RPM Rule	Programmed rule
`constant`	$(\mathbf{x}_a^1) \oslash (\mathbf{e})$
`progression`	$(\mathbf{x}_a^2 \otimes \mathbf{x}_a^2) \oslash (\mathbf{x}_a^1)$
`arithmetic plus`	$(\mathbf{x}_a^1 \otimes \mathbf{x}_a^2) \oslash (\mathbf{e})$
`arithmetic minus`	$(\mathbf{x}_a^1) \oslash (\mathbf{x}_a^2)$
`distribute three`	$(\mathbf{o}_a^1 \otimes \mathbf{o}_a^2 \otimes \mathbf{o}_a^3) \oslash (\mathbf{x}_a^1 \otimes \mathbf{x}_a^2)$

In summary, Equation (4) offers a flexible and expressive framework for modeling RPM rules, with the c_i terms serving as placeholders that are instantiated/learned based on the learned rule structure.

We have added a more elaborate explanation with an example to Section 4.2.

Review #2

Detail Comments: After reading the authors' responses to the reviewers' comments, I have no further remarks and recommend acceptance. I believe the added explanations and experiments made the manuscript clearer and more self-contained.

Thank you very much for your recommendation and for recognizing the added clarity and completeness of the manuscript.