MultiZebraLogic: A Multilingual Logical Reasoning Benchmark

Sofie Helene Bruun, Dan Saattrup Smart
The Alexandra Institute
Rued Langgaards Vej 7, 5D, 2300 Copenhagen S, Denmark
{sofie.bruun, dan.smart}@alexandra.dk

Abstract

We create high-quality datasets for LLM evaluation of logical reasoning skills across nine different languages, which have been manually checked by fluent speakers. The datasets consist of so-called zebra puzzles, and we analyse different ways of tuning the difficulty of the puzzles to fit modern LLMs. This includes the size of the puzzle (number of objects and number of clues), as well as a novel addition of red herring clues containing only irrelevant information. We show that presence of red herrings indeed makes the puzzles significantly harder for the models, and we find puzzle sizes 2×3 and 4×5 are sufficiently challenging for GPT-4o mini (a non-reasoning model) and o3-mini (a reasoning model), respectively. We analyse whether LLM performance of these are sensitive to the language, the cultural sensitivity of the puzzle theme, and the choice of clue types. These analyses are conducted with English and Danish, where we show that there is no significant difference for either of these three aspects, at least for the OpenAI models GPT-4o mini and o3-mini, chosen as representative non-reasoning and reasoning models, respectively. We publish the datasets for each of the nine languages for the identified sizes 2×3 and 4×5. We also publish the code used to generate the puzzles, which can be used to extend the benchmark into more languages.

Keywords: NLP evaluation, language resources, reasoning, LLM, logical reasoning

1. Introduction

With the advent of large language models (LLMs) with reasoning capabilities, evaluating their logical reasoning skills is essential. Existing reasoning datasets focus solely on *common-sense reasoning* (Zellers et al., 2019; Lin et al., 2021; Ponti et al., 2020) or English-only tasks (Lin et al., 2025; Chen et al., 2026; Patel et al., 2024; Wei et al., 2025).

To remedy this, we create *MultiZebraLogic*, a multilingual logical reasoning benchmark using zebra puzzles (Vassberg and Vassberg, 2009). First published in 1962, these puzzles require multi-step reasoning: the solver is given objects with attributes and clues describing relationships between them[1], finding a solution that satisfies all clues (see Figure 1).

This type of constraint satisfaction problem is easy to generate and requires multiple steps to solve. Simple algorithms for solving zebra puzzles exist, but to follow them, most humans would need to draw diagrams of excluded combinations. But assuming that such diagrams are allowed, humans can thus solve any zebra puzzle, albeit slowly.

In building the benchmark we identify appropriate sizes (number of objects and number of attributes per object) for LLM evaluation, and also examine other ways of increasing difficulty by adding red herrings (non-informative clues), more clue types, and a culture-specific theme: Danish smørrebrød (open sandwiches) with different ingredients.

Our main contributions are:

- A multilingual logical reasoning benchmark[2] designed for both reasoning and non-reasoning LLMs, covering 9 Germanic languages[3].
- Source code for puzzle generation built for scalability to more languages or themes [4].
- Analysis of effects on puzzle difficulty from red herrings (non-informative clues), a culture-specific theme, clue types, and a medium vs. high resource language.

2. Related Work

A wide range of benchmarks has been developed to systematically evaluate LLMs' logical reasoning skills across different reasoning types and complexities.

Lin et al. (2025) built a "ZebraLogic" benchmark to measure how logical reasoning performance of LLMs scale with zebra puzzle complexity in English. Our puzzle generation approach will be similar, but we attempt to increase difficulty for all puzzle sizes by adding more clue types, more languages as well as red herrings (non-informative clues).

[1] The original question was "Who owns the zebra?", which has named the puzzle genre.

[2] https://huggingface.co/datasets/alexandrainst/zebra_puzzles

[3] English, Danish, Swedish, Norwegian Bokmål, Norwegian Nynorsk, Faroese, Icelandic, German and Dutch.

[4] https://github.com/alexandrainst/zebra_puzzles

Proceedings of The Fourth Workshop on the Role of Resources in the Age of Large Language Models (RESOURCEFUL 2026), pages 119–130 May 11, 2026.

```
A row of houses have numbers 1 to 2 from left to
right.

In each house lives a person with unique attributes
in each of the following categories:

Jobs: nurse and police officer.
Favourite book genres: fantasy and romance.
Hobbies: bouldering and handball.

We also know the following:

1. The person who plays handball knows that snails
   are molluscs.
2. The police officer lives to the left of the nurse.
3. The person who plays handball does not live in
   house no. 2.
4. The romance reader lives in house no. 2.
5. The person with glasses does not live in house
   no. 1.

Who has which attributes and lives in which house?

Please submit your answer as a JSON dictionary in
the format below. Each row must begin with object_X
where X is the house number. Each column represents
a category, and they should be in the same order
as in the list of categories above.

{
    "object_1": [
        "jobs_1",
        "favourite book genres_1",
        "hobbies_1"
    ],
    "object_2": [
        "jobs_2",
        "favourite book genres_2",
        "hobbies_2"
    ]
}
```

Figure 1: A Zebra puzzle with 2 objects and 3 attributes for each object (2x3). Two red herrings are also included in the list of clues. See Table 1 for the solution.

Other benchmarks focus on specific reasoning domains. LogicBench (Parmar et al., 2024) targets 25 inference rules from propositional to non-monotonic logics, while JustLogic (Chen et al., 2025) emphasises deductive reasoning. SAT-Bench (Wei et al., 2025) challenges LLMs with search-based logical puzzles from Boolean satisfiability problems, and KOR-Bench (Ma et al., 2025) evaluates knowledge-orthogonal reasoning.

3. Methodology

3.1. Puzzle Generation

For a given theme and language, we generate puzzles with the following structure:

1. Introduction to the theme and rules including the number of objects, N_{objects}, and attributes per object, $N_{\text{attributes}}$.
2. A list of possible attributes and their categories.
3. A list of clues and red herrings.
4. Instructions on how to format the solution.

object_1	police officer	fantasy	handball
object_2	nurse	romance	bouldering

Table 1: Example of a $N_{\text{objects}} \times (N_{\text{attributes}} + 1)$ solution matrix for a 2×3 puzzle in the English houses theme. Each object represents a house and its row lists the attributes of the resident. See Figure 1 for the corresponding puzzle.

Objects could be houses, and attributes belong to categories such as jobs and pets. Multiple phrases[5] are included per attribute to fit different sentence structures without adding language-specific grammatical rules.

We start by generating solutions by randomly sampling categories and attributes within each category for each object, from a fixed list of categories and attributes. We also assign each row an object index. See Table 1 for an example of a solution.

To generate a clue, we sample a clue type from Table 2 and sample solution objects from the previous step along with attributes meeting the constraints of the clue. If the presented attribute order is irrelevant, attributes are sorted by category in the order that would typically sound the most natural[6]. Appendix B shows full clue examples.

Using the Python `constraint` package (Willemsen et al., 2025), we define a constraint satisfaction problem per puzzle and solve it. If a suggested clue changes the number of possible solutions, we keep it and iterate until a unique solution remains. Then, we remove each clue and only re-add it if the solution degenerates. This causes a bias towards including more informative clues, as illustrated in Appendix C.

Each red herring mentions either one of the attributes present in the solution, or none at all. We include 8 types; some follow the same templates as real clues, while others are new, such as random facts. We shuffle the order of clues and red herrings. See Figure 1 and Appendix A for examples of puzzles and all clue and red herring types.

Red herrings require less effort, as they contain no useful information. Some red herring types follow the same templates as real clues but with irrelevant attributes. Other types are new such as randomly chosen facts or statements about friendship related to objects. We end by shuffling the order of clues and red herrings.

3.1.1. Translation

The priorities for linguistic puzzle components are: 1) Correctness. Text must be linguistically acceptable. 2) Unambiguity. Clues must represent a

[5] E.g. “the baker”, “is a baker” and “is not a baker”.

[6] E.g., “The nurse loves oranges.” instead of “The person who loves oranges is a nurse.”

unique solution. 3) Naturalness. Phrases should sound typical of the chosen language. 4) Ease of generation. Puzzle generation should be simple. 5) Consistency. Text should be consistent in meaning and form across languages. 6) Diversity. A variety of properties and clue types should be included. There are tradeoffs between priorities[7].

Translation to new Germanic languages requires few changes to the puzzle generation algorithm itself, as we mostly avoid grammatical and social gender. The most important difference lies in the use of grammatical cases for attributes and clue types in Faroese, Icelandic and German. In German and Dutch, we add more forms of some clauses, to place the verb at the end of subordinate clauses. Some phrases are directly replaced after initial puzzle generation, such as the combination of "von dem" into "vom" in German.

All translations are drafted by the authors and reviewed by native/fluent speakers. For the drafts, we use Google Translate (Google), dictionaries (Svenska Akademien; Språkrådet and University of Bergen, a,b; Divvun.org), suggestions from GitHub Copilot with GPT-4.1 (GitHub; OpenAI) and Wikipedia (Wikipedia).

3.2. Evaluating LLM Performance

We explore puzzle difficulty for two LLMs. To represent a reasoning model, we choose o3-mini (OpenAI, 2025) with `max_completion_tokens` set to 100,000 and `reasoning_effort` set to "medium". As a non-reasoning model, we select GPT-4o mini (OpenAI, 2024) with `max_completion_tokens` set to 16,384 and `temperature` set to 0, to ensure reproducible evaluations[9]. They should output a JSON response for each puzzle, which is compared to the solution. See Appendix D for more details.

We use datasets of 100 puzzles per size with the smørrebrød theme, and evaluate using all sizes from 2×1 to 5×5, except 5×4 and 5×5 ($N_{\text{objects}} \times N_{\text{attributes}}$), as larger puzzles would take too many resources for both generation and evaluation. Puzzles with 1 object would require no clues. We generate 5 red herrings per puzzle and remove 4 or 5 to also create datasets with one or no red herring.

Performance is evaluated using the metrics of Lin et al. (2025): Puzzle-level accuracy, A_{puzzle}, which is 1 for a correct response and 0 otherwise; and cell-wise accuracy, A_{cell}, which is the fraction of correct cells in the response matrix.

We compute standard deviations assuming that A_{puzzle} follows a Bernoulli distribution and A_{cell} approximately follows a normal distribution. See Appendix E for more explanation of the use of standard deviations.

4. Results

4.1. Model Comparison

Fig. 2 shows the mean performance metrics of o3-mini and GPT-4o mini for different puzzle sizes and 5 red herrings. Based on the metrics, we see that 2×3 and 4×5 are suitably difficult sizes for GPT-4o mini and o3-mini, respectively, as their mean puzzle-level accuracies, $\overline{A_{\text{puzzle}}}$, are 0.36 ± 0.05 and 0.42 ± 0.05, respectively (with one σ uncertainties). $\overline{A_{\text{cell}}}$ for the two models is 0.70 ± 0.03 and 0.66 ± 0.04, respectively. An almost correct response that permutes the objects could get $A_{\text{cell}} = 0$. This rarely happens in practice, as shown in Appendix F.

To get an overall comparison score, we compute the t-statistic between scores for all puzzle sizes. We start by computing the difference in puzzle-level accuracy means, $\Delta\overline{A_{\text{puzzle}}}$, for each puzzle size evaluated by both LLMs (as illustrated in Appendix G). Then, we take the mean of all the differences across the puzzle sizes, $\overline{\Delta\overline{A_{\text{puzzle}}}} = 0.47 \pm 0.04$ and a t-statistic of 13. This shows that o3-mini performs significantly better than GPT-4o mini on these puzzles. Almost half the puzzles were only solved by o3-mini.

4.2. Red Herring Impact

To examine the effect of red herrings, we compare metrics with o3-mini for 0, 1 and 5 red herrings. For 0 vs. 1 red herring, we get $\overline{\Delta\overline{A_{\text{puzzle}}}} = 0.009 \pm 0.003$ and $t = 2.99$, and so, adding a red herring slightly increases difficulty (see Appendix H for more details).

If we add 5 red herrings instead, $\overline{\Delta\overline{A_{\text{puzzle}}}} = 0.032 \pm 0.007$ and $t = 4.77$. Going from 0 to 5 red herrings decreases $\overline{A_{\text{puzzle}}}$ by 4 ± 1 times as much as adding 1. Fig. 3 shows that the impact appears in large puzzles, with $\Delta\overline{A_{\text{puzzle}}} = 0.15 \pm 0.07$ for 4×5 with 5 red herrings.

Small puzzles are easy to o3-mini with or without red herrings. Using 5 red herrings has little impact on GPT-4o mini; $\overline{\Delta\overline{A_{\text{puzzle}}}} = 0.019 \pm 0.005$ and $\Delta\overline{A_{\text{puzzle}}} = 0.06 \pm 0.07$ for 2×3. Adding red herrings can be a simple alternative to increasing puzzle size for reasoning models.

[7] For unambiguity, we prefer "There are n houses between X and Y" although "X lives n houses away from Y" is slightly more natural. In Icelandic, for "X does not like H" we use "X elskar ekki H" instead of "X líkar ekki H" to avoid the dative case for X – this simplifies generation at a small cost to naturalness and consistency.

[9] We use a larger `max_completion_tokens` value for o3-mini to account for the reasoning trace.

[9] E.g. we assume a preference of `left_of` over `just_left_of` for $N_{\text{objects}} = 2$ across languages.

Clue type	Positional constraint	Requirement
found_at	$X = P$	
not_at	$X \neq P$	
same_object	$X = Y$	$N_{\text{attributes}} > 1$
not_same_object	$X \neq Y$	$N_{\text{attributes}} > 1$
next_to	$\lvert X - Y \rvert = 1$	$N_{\text{objects}} > 2$
not_next_to	$\lvert X - Y \rvert > 1$	$N_{\text{objects}} > 2$
just_left_of	$Y - X = 1$	$N_{\text{objects}} > 2$
just_right_of	$X - Y = 1$	$N_{\text{objects}} > 2$
left_of	$X < Y$	
right_of	$X > Y$	
between	$X < Y < Z \vee X > Y > Z$	$N_{\text{objects}} > 2$
not_between	$\neg(X < Y < Z \vee X > Y > Z) \wedge X \neq Y \wedge X \neq Z \wedge Y \neq Z$	$N_{\text{objects}} > 2$
one_between	$\lvert X - Y \rvert = 2$	$N_{\text{objects}} > 2$
multiple_between	$\lvert X - Y \rvert = N_{\text{between}} + 1$	$N_{\text{objects}} > 3$

Table 2: List of clue types and their positional constraints of objects X, Y and Z. P is a specific position, and N_{between} is the number of objects between A and B. Requirements are mentioned when they are stricter than the general puzzle generation requirements ($N_{\text{objects}} > 1, N_{\text{attributes}} > 0$). When multiple clue types would reveal the same information, the requirements exclude one for improved naturalness[8].

		Danish smørrebrød	Danish houses	English houses
A_{puzzle}	Mean	0.42±0.05	0.33±0.05	0.40±0.05
	Sample standard deviation	0.5	0.5	0.5
A_{cell}	Mean	0.66±0.04	0.66±0.04	0.67±0.04
	Sample standard deviation	0.4	0.4	0.4

Table 3: Comparison of o3-mini performance on 4×5 puzzles with 5 red herrings in the Danish smørrebrød, Danish houses and English houses themes (100 of each). Standard errors are included for mean values. Performance does not vary significantly by theme.

4.3. Language Comparison

We compare evaluation metrics in Table 3 between themes and two languages: English, a high resource language, and Danish, a medium resource language. $\overline{A_{\text{puzzle}}}$ and $\overline{A_{\text{cell}}}$ vary by $< 2\sigma$ – both for Danish vs. English house-themed puzzles and for the Danish houses vs. smørrebrød themes. The means and sample standard deviations are close to 0.5 for both metrics, indicating that individual puzzle metrics often vary wildly between the possible values from 0 to 1. Logical reasoning ability appears generalisable even for a culture-specific theme, and so, we use the houses theme for Multi-ZebraLogic, as it is easier to translate.

4.4. Clue Type Difficulty

To measure effect of clue and red herring types on difficulty, we compare their frequencies to A_{cell}. For each puzzle size, we fit to A_{cell} as a function of clue type frequencies using linear regression. The model coefficients show the importance of clue types. We normalise them, so their absolute values sum to 1, and flip the sign to arrive at the clue type difficulty. Thus, the higher the difficulty of a clue type, the more that clue type reduces the cell accuracy when present:

$$\text{difficulty}_{\text{clue type}} = -\frac{\text{coefficient}_{\text{clue type}}}{\sum |\text{coefficient}|}. \quad (1)$$

Section 4.2 shows that red herrings contribute negatively to accuracy, but if we keep the number of red herrings per puzzle constant, no red herring type particularly confuses o3-mini compared to the rest. There is also no clear pattern in clue type difficulties among the real clues across puzzle sizes when testing on 100 puzzles per size. See Appendix I for more details.

5. Discussion and Perspectives

For o3-mini with medium reasoning effort, ZebraLogicBench found an $\overline{A_{\text{puzzle}}}$ of 88 % and an $\overline{A_{\text{cell}}}$ of 90.4 % for large puzzles of sizes 4×5, 5×3, 4×6, 5×4 and 6×3. This is higher than our accuracies for 4×5 (42 % and 70 %) and 5×3 (73 % and 80 %) in Fig. 2. Our puzzles are more difficult, and Fig. 3 shows that this can be fully explained by red herrings as they decrease $\overline{A_{\text{puzzle}}}$ by 15±7 % for 4×5 puzzles.

Several corrections and adjustments have been applied since the analysis of this paper, which could slightly improve model performance. For example,

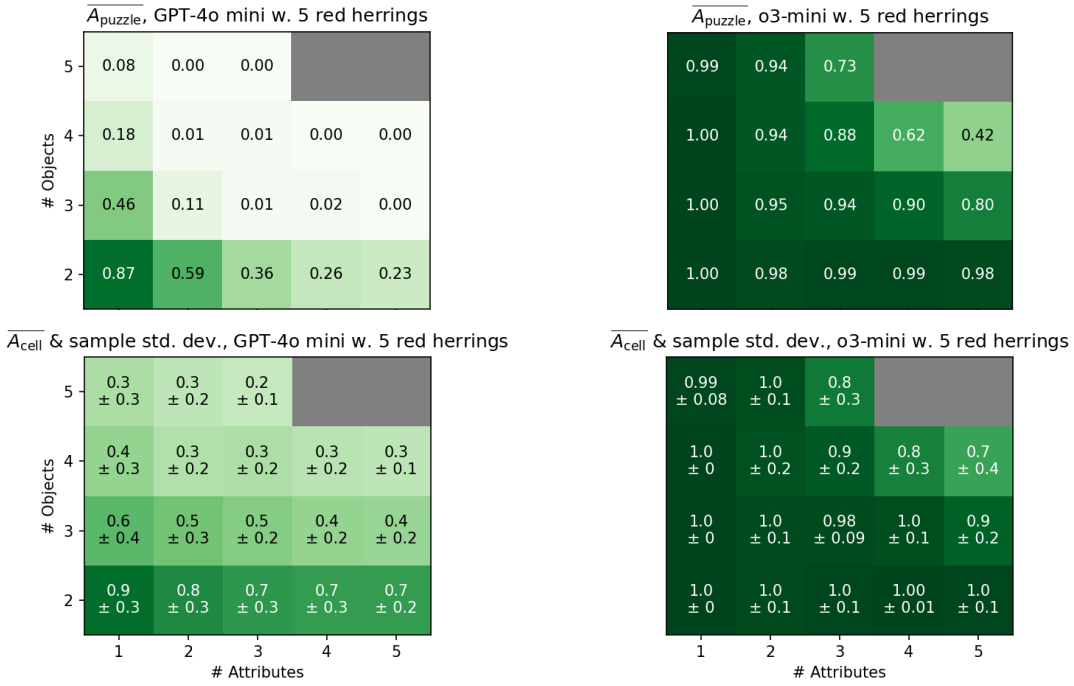

Figure 2: $\overline{A_{\text{puzzle}}}$ (upper row) and $\overline{A_{\text{cell}}}$ (lower row) for GPT-4o mini (left column) and o3-mini (right column) for 100 puzzles with 5 red herrings in the Danish smørrebrød theme. Sample standard deviations show the spread of A_{cell} (set to 0 for equal values). For A_{puzzle}, the mean values include all information. Sizes marked in grey are not evaluated. o3-mini performs better than GPT-4o mini for all evaluated sizes.

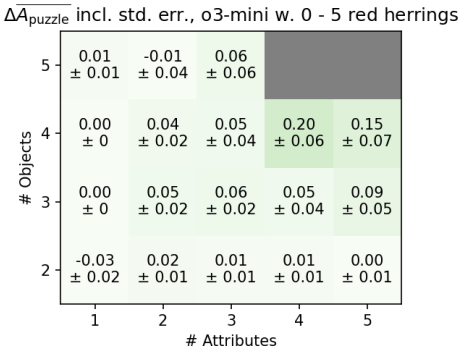

Figure 3: $\Delta\overline{A_{\text{puzzle}}}$ for o3-mini with 0 vs. 5 red herrings for 100 puzzles in the Danish smørrebrød theme. Using 5 red herrings gives a $> 2\sigma$ decrease in $\overline{A_{\text{puzzle}}}$ for sizes 3×2, 3×3, 3×5, 4×4, and 4×5.

only using the word football instead of soccer in English. We describe the changes in Appendix J. With more advanced LLMs, evaluating broader or more advanced reasoning skills could be useful. We suggest more puzzle and clue types in Appendix K.

6. Conclusion

We have published MultiZebraLogic datasets for benchmarking logical reasoning, and code for dataset generation. New languages or themes can be added as input for easy adaption. o3-mini can solve larger puzzles than GPT-4o mini, so for evaluation of reasoning models, we include 4×5 puzzles, and for other models, 2×3 puzzles. We always include 5 red herrings (and publish their indices), as this causes a $\overline{A_{\text{puzzle}}}$ drop of 15±7 % for o3-mini with 4×5 puzzles. Logical reasoning appears generalisable for o3-mini on 4×5 puzzles across Danish and English, and across the classic houses theme compared to the culture-specific smørrebrød theme. The puzzle generation algorithm prefers more informative clue types, but we find no clear correlation between included clue or red herring types and A_{cell}. The published dataset contains 128 puzzles for training (as few-shot examples) and 1024 for testing for sizes 2×3 and 4×5 in 9 languages.

7. Acknowledgements

We are very grateful to everyone who helped review the translations and language configuration files[10]. We thank the EU Horizon project TrustLLM (grant agreement number 101135671) and Danish Foundation Models[11] for funding this project.

[10] Annika Simonsen, Gardar Ingvarsson Juto, Lars Bungum, Mathias Stenlund, Jenny Kunz and Eike Güldenring.

[11] https://www.foundationmodels.dk/

8. Bibliographical References

Divvun.org. 2025. Divvun - Sámi language technology. [Online; accessed 28. Aug. 2025].

GitHub. 2025. GitHub Copilot · Your AI pair programmer. [Online; accessed 28. Aug. 2025].

Google. 2025. Google translate. [Online; accessed 28. Aug. 2025].

OpenAI. 2024. GPT-4o System Card.

OpenAI. 2025. Introducing GPT-4.1 in the API. [Online; accessed 8. Oct. 2025].

OpenAI. 2025. OpenAI o3-mini System Card.

Språkrådet and University of Bergen. 2025a. Bokmål til Nynorsk | Tekstoversetter | Wordify. [Online; accessed 28. Aug. 2025].

Språkrådet and University of Bergen. 2025b. Bokmålsordboka og Nynorskordboka - ordbøkene.no. [Online; accessed 28. Aug. 2025].

Svenska Akademien. 2025. svenska.se – Akademiens ordböcker. [Online; accessed 28. Aug. 2025].

Dylan Vassberg and J. Vassberg. 2009. Is einstein's puzzle over-specified?

Wikipedia. 2025. Wikipedia, the free encyclopedia. [Online; accessed 28. Aug. 2025].

Floris-Jan Willemsen, Sébastien Celles, and Gustavo Niemeyer. 2025. python-constraint. [Online; accessed 28. Aug. 2025].

9. Language Resource References

Chen, Jiangjie and He, Qianyu and Yuan, Siyu and Chen, Aili and Cai, Zhicheng and Dai, Weinan and Yu, Hongli and Chen, Jiaze and Li, Xuefeng and Yu, Qiying and others. 2026. *Enigmata: Scaling Logical Reasoning in Large Language Models with Synthetic Verifiable Puzzles*.

Michael K Chen, Xikun Zhang, and Dacheng Tao. 2025. Justlogic: A comprehensive benchmark for evaluating deductive reasoning in large language models. *arXiv preprint arXiv:2501.14851*.

Lin, Bill Yuchen and Le Bras, Ronan and Richardson, Kyle and Sabharwal, Ashish and Poovendran, Radha and Clark, Peter and Choi, Yejin. 2025. *ZebraLogic: On the Scaling Limits of LLMs for Logical Reasoning*. PID https://huggingface.co/datasets/WildEval/ZebraLogic.

Bill Yuchen Lin, Seyeon Lee, Xiaoyang Qiao, and Xiang Ren. 2021. Common sense beyond english: Evaluating and improving multilingual language models for commonsense reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1274–1287.

Kaijing Ma, Xeron Du, Yunran Wang, Haoran Zhang, Xingwei Qu, Jian Yang, Jiaheng Liu, Xiang Yue, Wenhao Huang, Ge Zhang, et al. 2025. Kor-bench: Benchmarking language models on knowledge-orthogonal reasoning tasks. In *The Thirteenth International Conference on Learning Representations*.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707.

Nisarg Patel, Mohith Kulkarni, Mihir Parmar, Aashna Budhiraja, Mutsumi Nakamura, Neeraj Varshney, and Chitta Baral. 2024. Multi-logieval: Towards evaluating multi-step logical reasoning ability of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20856–20879.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. Xcopa: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376.

Anjiang Wei, Yuheng Wu, Yingjia Wan, Tarun Suresh, Huanmi Tan, Zhanke Zhou, Sanmi Koyejo, Ke Wang, and Alex Aiken. 2025. Satbench: Benchmarking llms' logical reasoning via automated puzzle generation from sat formulas. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33820–33837.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800.

A. Advanced English Houses Example

A longer, more advanced Zebra puzzle example compared to Figure 1, can be found in Figure 4.

B. Clue Type Examples

Table 4 shows an example of each clue type and Table 5 shows an example of each red herring type.

Clue type	Example
found_at	The person who plays board games lives in house no. 2.
not_at	The science fiction reader does not live in house no. 1.
same_object	The police officer reads crime novels.
not_same_object	The dog owner does not like apples.
next_to	The zebra owner lives next to the person who loves strawberries.
not_next_to	The person who boulders does not live next to the person who loves blackcurrants, and they are different people.
just_left_of	The teacher lives to the immediate left of the rabbit owner.
just_right_of	The teacher lives to the immediate right of the coffee drinker.
left_of	The rabbit owner lives to the left of the person who plays board games.
right_of	The Brit lives to the right of the romance reader.
between	The person who loves blackcurrants lives between the police officer and the person who loves wild strawberries.
not_between	The rabbit owner does not live between the coffee drinker and the juice drinker, and they are three different people.
one_between	There is one house between the Norwegian and the police officer.
multiple_between	There are 2 houses between the nurse and the baker.

Table 4: An example clue for each clue type using the English houses theme.

C. Clue Type Frequency

Clues are randomly generated, but only included when useful, and this affects the frequencies of

```
A row of houses have numbers 1 to 4 from left to
right.

In each house lives a person with unique attributes
in each of the following categories:

Jobs: baker, nurse, shop assistant and teacher.
Pets: budgerigar, cat, dog and rabbit.
Drinks: coffee, juice, milk and tea.
Hobbies: board games, handball, soccer and tennis.
Favourite fruits: apple, blackcurrant, orange and
wild strawberry.

We also know the following:

1. The person with a master's degree in mathematics
   does not live in house no. 1.
2. The teacher lives to the immediate right of the
   coffee drinker.
3. The shop assistant lives to the immediate right
   of the budgie owner.
4. The rabbit owner does not live between the coffee
   drinker and the juice drinker, and they are three
   different people.
5. The dog owner does not like apples.
6. The person who owns a cactus often sails.
7. There are 2 houses between the nurse and the baker.
8. The tea drinker does not live next to the person
   who loves blackcurrants, and they are different
   people.
9. There is one house between the coffee drinker and
   the milk drinker.
10. There are many cars on the street.
11. There are 2 houses between the milk drinker and
    the tea drinker.
12. The nurse lives next to the dog owner.
13. There is one house between the person who plays
    board games and the person who plays handball.
14. The person who plays football lives next to the
    person who plays board games.
15. There are 2 houses between the person who plays
    football and the person who loves blackcurrants.
16. The person with a tattoo does not live in house
    no. 3.
17. The milk drinker is good friends with the person
    with a pet that is old for its species.
18. There is one house between the cat owner and the
    person who loves oranges.

Who has which attributes and lives in which house?

Please submit your answer as a JSON dictionary in the
format below. Each row must begin with object_X where
X is the house number. Each column represents a
category, and they should be in the same order as in
the list of categories above.

{
    "object_1": [
        "jobs_1",
        "pets_1",
        "drinks_1",
        "hobbies_1",
        "favourite fruits_1"
    ],
    "object_2": [
        "jobs_2",
        "pets_2",
        "drinks_2",
        "hobbies_2",
        "favourite fruits_2"
    ],
    "object_3": [
        "jobs_3",
        "pets_3",
        "drinks_3",
        "hobbies_3",
        "favourite fruits_3"
    ],
    "object_4": [
        "jobs_4",
        "pets_4",
        "drinks_4",
        "hobbies_4",
        "favourite fruits_4"
    ]
}
```

Figure 4: A Zebra puzzle with 4 objects and 5 attributes for each object (4x5). Five red herrings are also included in the list of clues.

Red herring type	Example
`same_herring`	The person who loves wild strawberries loves physics.
`next_to_herring`	The Dutchman lives next to the person with a bike.
`double_herring`	The person who owns a cactus often sails.
`fact`	Snails are molluscs.
`object_fact`	The shop assistant knows that several of the houses have a green door.
`friends`	The person who boulders is good friends with the person who plays video games.
`herring_found_at`	The person who has been to Canada lives in house no. 3.
`herring_not_at`	The person with a master's degree in mathematics does not live in house no. 1.

Table 5: An example of each red herring type in the English houses theme. Some red herrings may sound informative, but they are all irrelevant to the solving process.

clue types. The number of clues may also vary between puzzles generated with the same inputs. To compare clue type frequencies, we count and normalise them in each puzzle, so the frequencies sum to 1. Then, we take the mean across puzzles of the same size (same N_{objects} and $N_{\text{attributes}}$).

Fig. 5 shows the mean normalised frequencies for 100 puzzles with 5 red herrings. Naturally, the herrings are relatively frequent for small puzzles that require few real clues. For real clues, the frequencies are connected to their usefulness. For example, `not_same_object` is relatively rare for most puzzle sizes, as it only excludes one link between attributes. `not_between`-clues connect 3 objects and fully include the `not_same_object`-clue – this makes them more informative and more common.

To change frequencies of clue types or red herring types, selection weights can be adjusted. These are equal per default.

D. Evaluation Details

When evaluating the models, if the API returns an `InternalServerError`, `APIError`, `APIConnectionError`, `RateLimitError`, `RateLimitError`, we wait 5 seconds and try again up to 4 more times, as these errors do not depend on puzzle difficulty, unlike, e.g., `APITimeoutError`. For continued errors or other error types, we treat them as a wrong solution.

E. Uncertainty Calculation

We will generally propagate uncertainties σ for a function $f(a, b, ...)$ using

$$\sigma_{f(a,b,...)} = \sqrt{\sigma_a^2 \left(\frac{\partial f}{\partial a}\right)^2 + \sigma_b^2 \left(\frac{\partial f}{\partial b}\right)^2 + ...} \quad (2)$$

One standard deviation corresponds to a confidence interval of 68 % and two corresponds to 95 %. The sample standard deviation of the Bernoulli-distributed puzzle-level accuracies, A_{puzzle}, is:

$$\sigma_{A_{\text{puzzle}}} = \sqrt{\overline{A_{\text{puzzle}}} * (1 - \overline{A_{\text{puzzle}}})}. \quad (3)$$

The sample standard deviation of cell-wise accuracies, A_{cell} is computed as:

$$\sigma_{A_{\text{cell}}} = \sqrt{\frac{\sum_i |A_{\text{cell},\, i} - \overline{A_{\text{cell}}}|^2}{N_{\text{puzzles}} - 1}}. \quad (4)$$

To get the standard deviation of the mean scores (standard error of the mean), we divide by $\sqrt{N_{\text{puzzles}}}$:

$$\sigma_{\overline{A}} = \frac{\sigma_A}{\sqrt{N_{\text{puzzles}}}}. \quad (5)$$

The standard deviation of the difference in means, $\Delta\overline{A}$, is computed as

$$\sigma_{\overline{A}} = \sqrt{\sigma_{A_i}^2 + \sigma_{A_j}^2} \quad (6)$$

for models i and j. To do this, we assume that scores can be treated as independent, although the models can actually be evaluated on the same puzzles. The standard deviation of the mean difference in means, $\overline{\Delta\overline{A}}$, is

$$\sigma_{\overline{\Delta\overline{A}}} = \sqrt{\frac{\sum_i |(\Delta\overline{A})_i - \overline{\Delta\overline{A}}|^2}{N_{\text{evaluated sizes}} - 1}}. \quad (7)$$

The t-statistic (difference in units of standard deviations) is then

$$t = \frac{\overline{\Delta\overline{A}}}{\sigma_{\overline{\Delta\overline{A}}}}. \quad (8)$$

F. Best Permuted Cell-Wise Accuracies

If a model correctly connects attributes, but switches the object numbers, this is punished

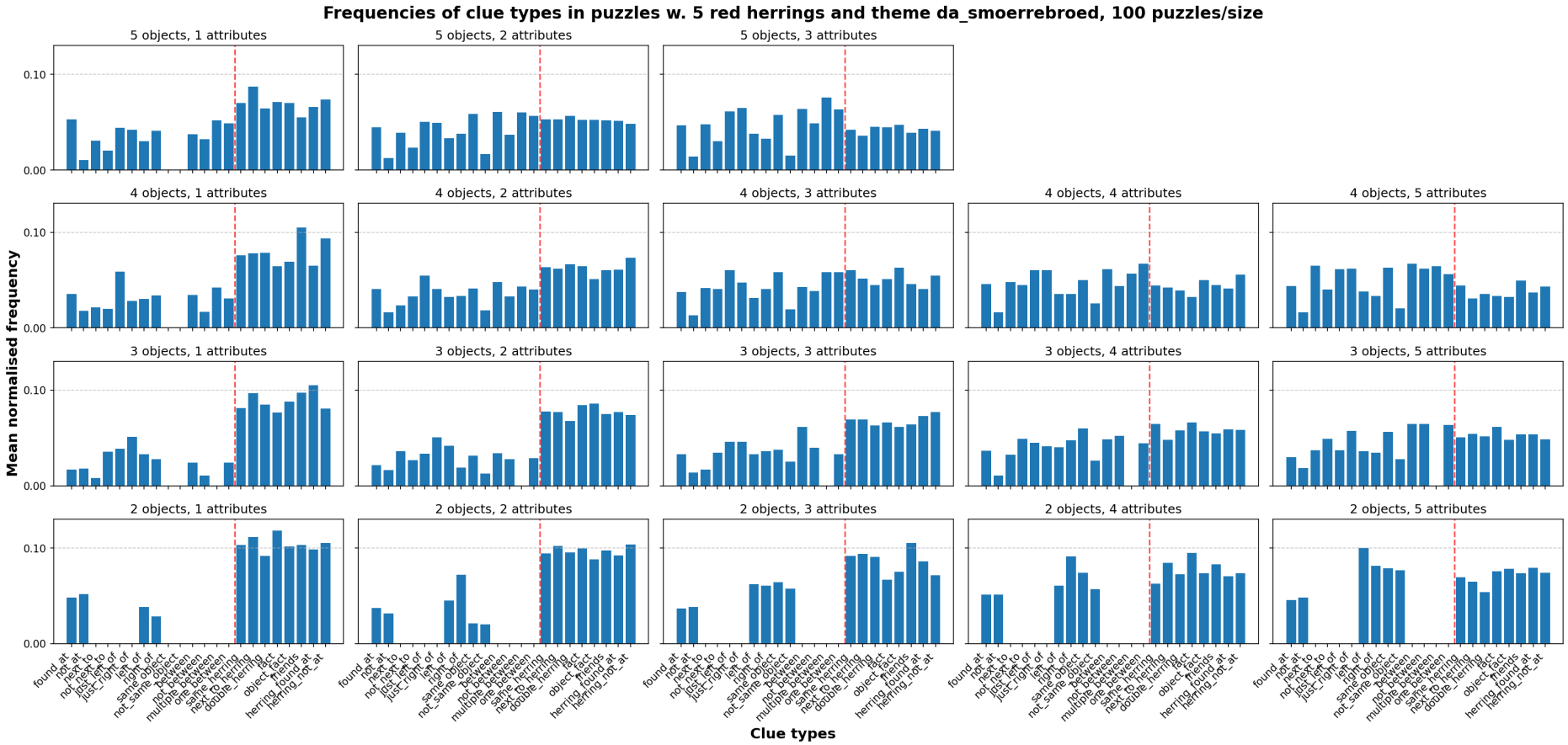

Figure 5: Mean normalised frequencies of all clue types in puzzles with the Danish smørrebrød theme and 5 red herrings. To the right of the red line, all 'clues' are red herrings. Some clue types are only used above certain puzzle sizes – see Table 2. Frequently selected clues are typically more informative.

harder by A_{cell} than if attributes were switched within a category. To notice if this happens, we check the best permuted cell-wise accuracy, $A_{\text{best cell}}$, which is the maximum cell-wise accuracy for all object permutations. This is always equal to or higher than A_{cell}.

The difference is not significant for responses from o3-mini on 4×5 puzzles with 5 red herrings in the Danish smørrebrød theme. $\overline{A_{\text{best cell}}}$ values are generally a bit higher for GPT-4o mini with $\overline{A_{\text{best cell}}} - \overline{A_{\text{cell}}} = 0.11 \pm 0.4$ for 2×3 puzzles. If the effect is major for some LLMs, $A_{\text{best cell}}$ could be considered as an extra metric for comparison.

G. Model comparison

In Fig. 6, for each puzzle size evaluated by both models, we take $\Delta\overline{A_{\text{puzzle}}}$ and $\Delta\overline{A_{\text{cell}}}$. The figure shows that o3-mini performs better than GPT-4o mini, especially for medium sizes such as 4×2, which are hard for GPT-4o mini but still easy for o3-mini.

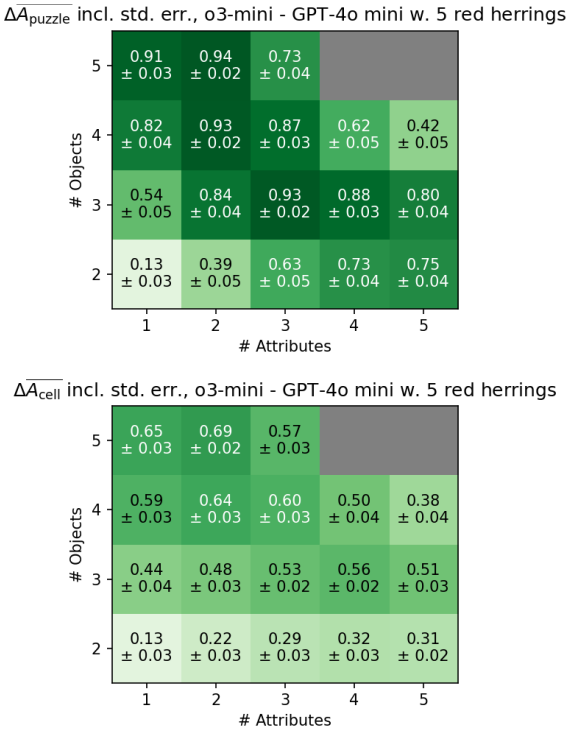

Figure 6: Difference in mean score between o3-mini and GPT-4o mini for 100 puzzles with 5 red herrings in the Danish smørrebrød theme. The upper plot shows puzzle-level accuracies and the lower shows cell-wise accuracies. The uncertainties show the standard deviations of the differences in mean scores.

H. The Impact of One Red Herring

Fig. 7 shows that adding a single red herring typically decreases $\overline{A_{\text{puzzle}}}$, but the effect is very small and not significant for most puzzle sizes – even the largest ones, where we see the greatest effect of adding 5 red herrings in Fig. 3.

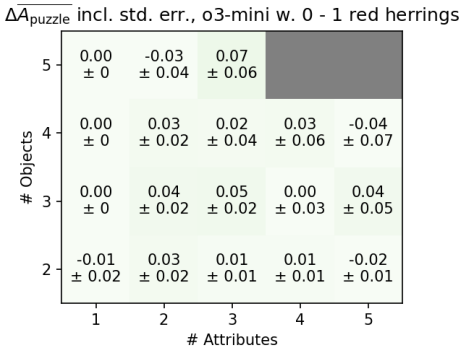

Figure 7: $\Delta\overline{A_{\text{puzzle}}}$ for o3-mini with 0 vs. 1 red herrings for 100 puzzles in the Danish smørrebrød theme. Including 1 red herring slightly decreases $\overline{A_{\text{puzzle}}}$, but the effect is not consistent across puzzle sizes.

I. Clue type difficulties

In Fig. 8, clue type difficulties are shown for o3-mini. They show no consistent pattern across the puzzle sizes. Clue type difficulties for o3-mini are more accurate for large puzzles, as A_{cell} values are more diverse (see Fig. 2).

J. Adjustments and Corrections

Multiple linguistic adjustments have been made since the results of this paper were computed. Below we mention the most important changes.

For red herring generation, we have replaced the interest in watching football, as this could be confused with the hobby of playing football, which is an attribute in some puzzles. These occur together in about 11 % of 4×5 puzzles and 3 % of 2×3 puzzles – both with 5 red herrings. We have replaced watching football with watching ski jumping. We were also using the words 'soccer' and 'football' interchangeably in English, and are now only using 'football'.

We are testing a different puzzle template including a new description of the desired JSON format in which sorting the attributes by category is not required. If this works well for most LLMs on Danish houses in EuroEval, it will be translated to all included languages. Otherwise, we will consider further clarification of the rules etc.

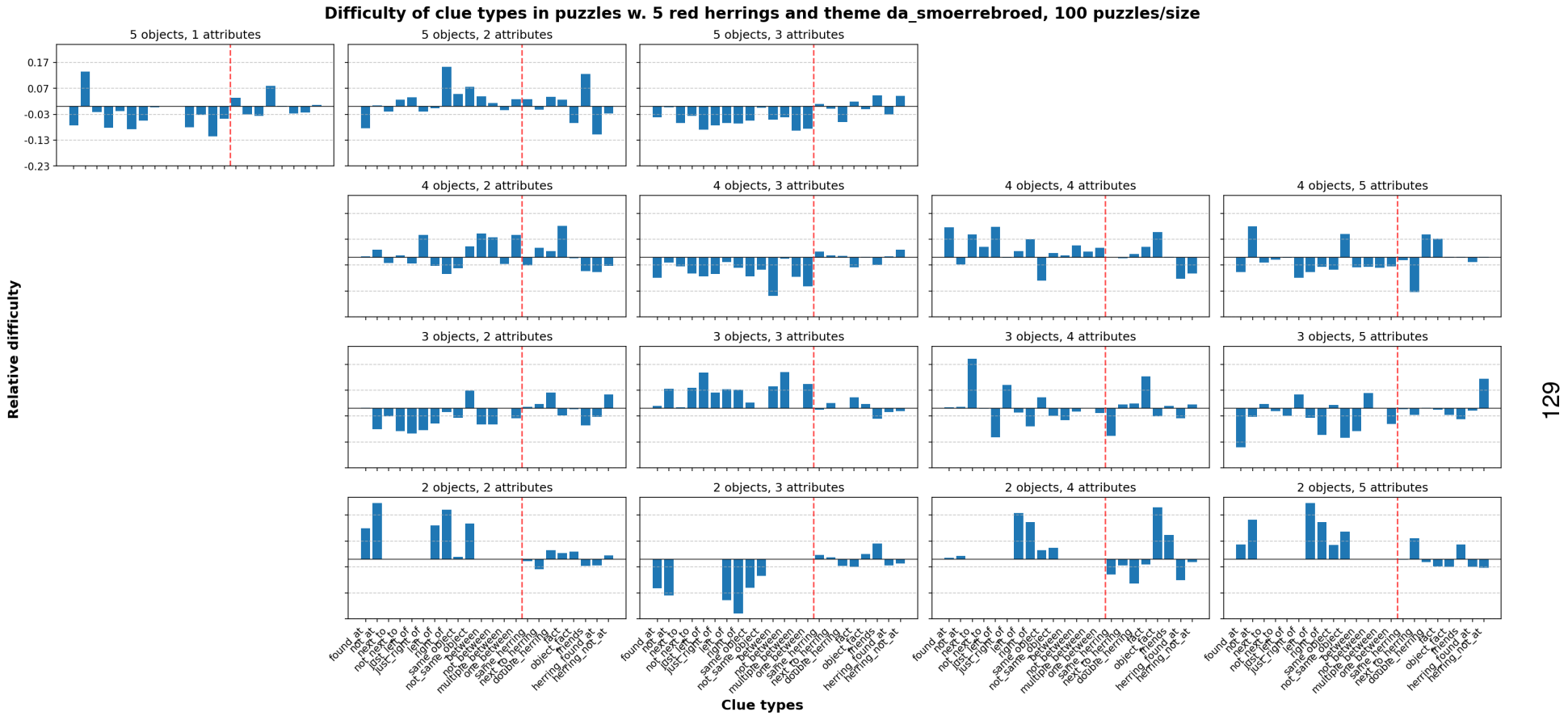

Figure 8: Clue type difficulties as predicted contributions of clue type frequencies to A_{cell} values for o3-mini on puzzles in the Danish smørrebrød theme with 5 red herrings. Red herrings are on the right side of the red line. Some small puzzle sizes are not included, as difficulties cannot be estimated for constant A_{cell}.

K. Suggested Expansions

To expand how logical reasoning is evaluated, an approach would be to use more puzzle types. A variation of zebra puzzles could be houses on a grid instead of a linear street. Attributes could also be non-unique or described by super-attributes (e.g. “The Latvian owns an animal larger than a cat” which could be a zebra or a dog) or ordinal attributes (e.g. “The poetry reader owns a larger animal than the Latvian does”). Some houses could be empty or house multiple people. One person could also have multiple attributes in the same category.

For the current puzzle type, different clue types could be introduced, such as “half-herrings” that provide some useful and some useless information. For example, “The minister’s sister likes to make paintings of the baker’s cat” reveals that the baker is the cat owner, but not which resident likes to paint, as the sister might not live on the same street.

Other types of clues could be added for variety, such as “The baker is either Norwegian or has a dog”, and for all real clue types, a red herring type of a similar structure could be created.