

GlyphPattern: An Abstract Pattern Recognition for Vision-Language Models

Zixuan Wu
Northeastern University
zi.wu@northeastern.edu

Yoolim Kim
Wellesley College
ykim6@wellesley.edu

Carolyn Jane Anderson
Wellesley College
carolyn.anderson@wellesley.edu

Abstract

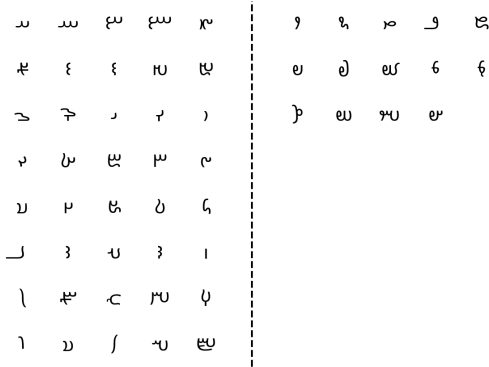
Vision-Language Models (VLMs) have made rapid progress in reasoning across visual and textual data. While VLMs perform well on vision tasks that they are trained on, our results highlight key challenges in abstract pattern recognition. We present GlyphPattern, a 954 item dataset that pairs 318 human-written descriptions of visual patterns from 40 writing systems with three visual presentation styles.

GlyphPattern evaluates abstract pattern recognition in VLMs, requiring models to understand and judge natural language descriptions of visual patterns. GlyphPattern patterns are drawn from a large-scale cognitive science investigation of human writing systems; as a result, they are rich in spatial reference and compositionality. Our experiments show that GlyphPattern is challenging for state-of-the-art VLMs (GPT-4o achieves only 55% accuracy), with marginal gains from few-shot prompting. Our detailed analysis reveals errors at multiple levels, including visual processing, natural language understanding, and pattern generalization.

1 Introduction

A key aspect of intelligence is the ability to recognize abstract patterns. Bongard (1967) proposed a task to measure abstract pattern recognition in artificial intelligence using a kind of partitioned diagram, now known as a *Bongard problem*. Each diagram shows two sets of items; the goal is to describe the pattern that distinguishes them. Bongard problems have inspired a number of visual benchmarks (Nie et al., 2020; Jiang et al., 2022; Spratley et al., 2023; Wu et al., 2024b). However, existing benchmarks are easier than classical Bongard problems because they do not involve a natural language description of the pattern.

We present GlyphPattern, an abstract pattern recognition benchmark for vision-language models (VLMs) that challenges models to understand



Characters on the right side in the image are characters that contain a loop or circular shape.

Figure 1: Example GlyphPattern item in the left-right visual style. The writing system is Avestan.

natural language pattern descriptions. GlyphPattern tests the ability of VLMs to recognize visual properties that distinguish two groups of symbols, pairing together human-written and human-verified patterns with three different visual styles designed to test the robustness of abstract pattern recognition in VLMs, for a benchmark of 954 items.

A key feature of GlyphPattern is that it builds upon the Glyph project (Kim et al., 2024), a cognitive science investigation of cross-linguistic universals in human writing systems that collects human-written patterns in a gamified web-based experiment. Glyph provides a high quality data source since each rule is validated by multiple players.

Figure 1 shows an example item from GlyphPattern. Each human-written rule describes a visual property of a subset of the characters in the writing system: in this case, the characters on the right of the image contain a loop, while the characters on the left do not. We test three different visual styles: a left-right partition, a circular boundary, and a color contrast (Figure 2).

We use GlyphPattern to test the pattern recognition ability of eight VLMs in a multiple choice task.

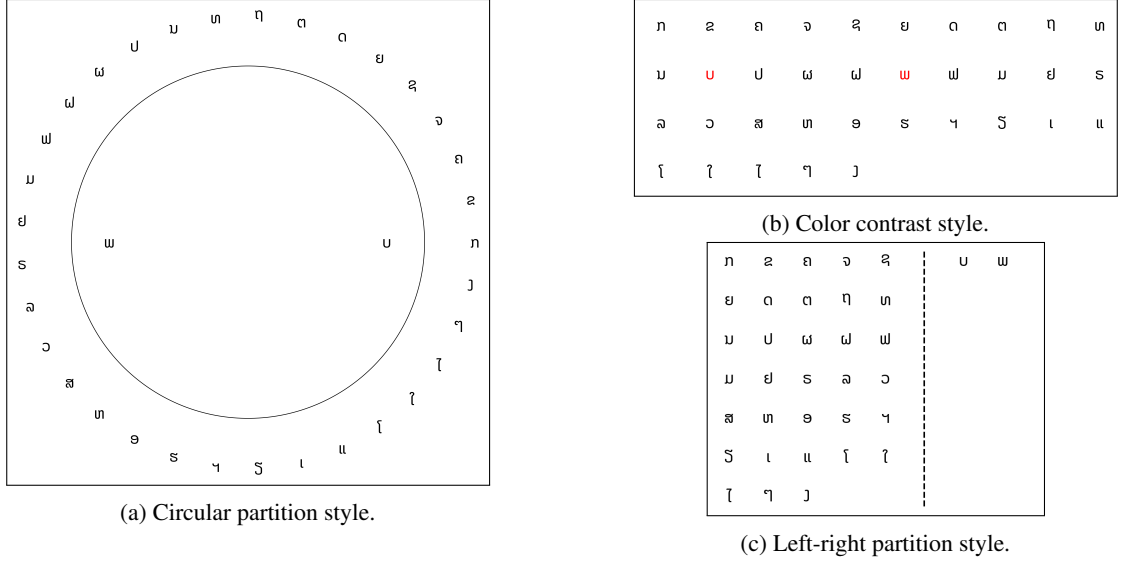


Figure 2: GlyphPattern’s three visual styles for the Lao pattern *Characters that are symmetrical around a vertical axis*.

Our zero-shot results show that this is a challenging task for all models: the state-of-the-art GPT-4o model achieves only 55.6% accuracy. We find that although few-shot prompting does not help GPT-4o, it enables Gemini-1.5 to match GPT-4o’s performance. For the best-performing models, we also explore a free response format and find that models generate valid rules less than 16% of the time.

GlyphPattern’s rich human-written pattern set and multiple visual styles allow us to explore the impact of different aspects of the task, including the length and syntactic complexity of the rules, the visual saliency of different writing systems, and the visual presentation style. Our fine-grained error analysis reveals that models “fail to see” key portions of the image and fail to understand certain terms in the natural language descriptions. We hope that GlyphPattern will guide improvements in VLMs by highlighting weaknesses in both the vision and language capabilities of current models. Our key contributions are as follows:

- A 954 item abstract pattern recognition benchmark for VLMs based on human-written descriptions of patterns in visual properties of writing system characters.
- Benchmarking results for eight VLMs, showing that GlyphPattern is challenging for state-of-the-art models: Gemini-1.5 and GPT-4o achieve accuracy around 55%.
- A comparison of zero-shot, few-shot, chain-

of-thought, and free response results for the two best-performing models.

- A detailed analysis of the impact of image style, pattern complexity, and human-identified cross-linguistic pattern types.
- An exploration of error sources, revealing mistakes that arise at multiple stages, including image-processing, language understanding, and abstract pattern reasoning.

2 Related Work

There is a long history of assessing intelligence through the ability to recognize abstract patterns, both in non-verbal tests of human intelligence and in AI. [Bongard \(1967\)](#)’s influential work on pattern recognition in AI proposed a test using a diagram that partitions items into two groups. So-called *Bongard problems* evaluate an agent’s few-shot pattern recognition ability: the goal is to recognize and describe the pattern from a handful of examples.

2.1 Abstract Pattern Recognition Benchmarks

General-purpose visual understanding requires the ability to recognize patterns from a handful of examples. Consequently, it is important to understand the abstract reasoning abilities of VLMs alongside their real-world, domain-specific performance ([Hudson and Manning, 2019](#); [Yue et al., 2024](#)).

A number of abstract pattern recognition benchmarks for few-shot image classification have been

inspired by intelligence tests. [Barrett et al. \(2018\)](#) adapt IQ test items to measure few-shot pattern recognition in computer vision models. The model is prompted with a grid of example images, and asked to select the next member of the sequence. [Zhang et al. \(2019\)](#) and [Teney et al. \(2020\)](#) use a similar format inspired by Raven’s Progressive Matrices ([Raven, 1936](#)). [Zerroug et al. \(2022\)](#), by contrast, use an odd-one-out classification task.

There are several Bongard-inspired image classification benchmarks. [Nie et al. \(2020\)](#) propose Bongard-Logo, a benchmark where models classify images as members or non-members of a pattern. Bongard-HOI ([Jiang et al., 2022](#)) and Bongard-OpenWorld ([Wu et al., 2024b](#)) use real-world images. Most similar to our work, [Spratley et al. \(2023\)](#) present a dataset of Unicode character patterns; unlike our patterns, which are easily solved by humans, their automatically-identified patterns have only a 55% human solve rate.

These benchmarks are a simplification of Bongard’s original task, because they test whether the model can *continue* a pattern rather than *describe* it. Our GlyphPattern benchmark comes closer by challenging VLMs to understand and even produce natural language descriptions of abstract patterns.

2.2 Benchmarking Vision-Language Models

Our focus is on benchmarking VLMs, which may struggle with image processing, natural language understanding, or relating the two modalities.

Compositionality in VLMs An important feature of GlyphPattern is that its human-written patterns are linguistically rich in compositionality and spatial relations. Compositionality is a key evaluation area for VLMs. Existing benchmarks explore compositionality in object attributes ([Ray et al., 2023](#)), spatial relations ([Liu et al., 2023](#); [Kamath et al., 2023](#)), and visual concepts ([Suhr et al., 2019](#); [Ma et al., 2023](#)). A popular benchmark is Winoground ([Thrush et al., 2022](#)), which targets linguistic compositionality with pairs of captions containing the same words in different syntactic structures. [Diwan et al. \(2022\)](#) argue that it is text-image linking that makes Winoground challenging.

Abstract Pattern Recognition in VLMs The increasing prominence of LLM-based systems has led to widespread interest in multimodality. However, recent work on abstract pattern recognition reveals surprising weaknesses of VLMs. [Ji et al.](#)

(2022) find poor zero-shot performance on tangram puzzle-based tasks. [Rahmanzadehgervi et al. \(2024\)](#) show that VLMs struggle with extremely easy visual reasoning tasks, like judging if circles overlap, which they attribute to visual processing issues. [Wu et al. \(2024a\)](#) ask VLMs to verify captions for displays of geometric shapes from the NLVR dataset ([Suhr et al., 2017](#)), and find that even state-of-the-art models perform little better than randomly.

Our findings converge with and expand on [Wu et al. \(2024a\)](#)’s findings: using a different task and data domain, we find similar weaknesses, including inconsistent gains from few-shot prompting. However, our fine-grained analysis allows us to dig deeper into the sources of errors.

Computer Vision and Writing Systems GlyphPattern items come from a diverse set of human writing systems. There is previous work on computer vision systems for recognizing characters in writing systems, including [Magre and Brown \(2022\)](#)’s MNIST ([Deng, 2012](#)) inspired dataset and [Lake et al. \(2015\)](#)’s work on discovering schemas for recognizing characters. Our goal is to probe abstract pattern recognition in VLMs, rather than to improve their ability to transcribe writing systems.

3 Dataset

GlyphPattern uses patterns from human writing systems to benchmark abstract pattern recognition in VLMs. It pairs 318 human-written descriptions of patterns that partition characters in 40 human writing systems with three different visual presentation styles, for a total of 954 items.¹

3.1 Online Data Collection

We draw on human-written patterns collected as part of the online applet *Glyph* created by the Minds and Traditions Max Planck Independent Research Group, which explores cross-linguistic universals in the visual properties of human writing systems. We use the publicly-available data collected from the online applet.

In *Glyph*, players select a subset of characters in an unfamiliar writing system and write a description of the pattern that separates the selected characters (the *included characters*) from the rest of the script (the *excluded characters*). Players must verify their rule by re-selecting the included

¹The dataset and support code are available at <https://github.com/Wellesley-EASEL-lab/GlyphPattern>.

characters after at least three minutes using their previously written rule.

3.2 Data Cleaning

We use a subset of 318 filtered, validated, and standardized player-written rules produced by the *Glyph* team. Rules were independently validated by five researchers, who identified redundant rules and grouped the rules into categories. Rules were then summarized by ChatGPT in order to standardize wording. We use this publicly-available subset of rules, but have an additional rewording step so that all rules are compatible with our prompt prefixes.

3.3 Image Construction

The *Glyph* dataset associates each rule with a script and a partition of characters. We retrieve the Unicode symbols for the included and excluded characters, and generate images in three visual styles (Figure 2). In the *color* style, the included characters are shown in red, while the excluded are black. In the *left-right* style, a dashed line partitions the image into two halves. The included characters are on the right side and the excluded on the left. In the *circle* style, the characters are arranged along the circumference of a circle: the included characters are inside and the excluded outside.

Exploring three ways of visualizing the partition allows us to measure how sensitive VLMs are to visual properties of the images. If a model performs well on one style but not the others, it would indicate that the model understands the pattern, but fails to apply it to certain kinds of images.

3.4 Benchmark Construction

We present *GlyphPattern* as a multiple choice Visual Question Answering task. Given an image and a multiple choice question with four pattern options, the goal is to pick the pattern that explains which characters are included and excluded. We randomly sample the three other answer options and manually verify that they are not correct patterns for the image, replacing them if so.

4 Experiments

We use *GlyphPattern* to evaluate eight VLMs. All experiments are run with greedy decoding on a single A100 GPU. We evaluate eight models,² both open and closed-source state-of-the-art VLMs:

- **GPT-4o** (Achiam et al., 2023) is a closed-source multimodal chat model.
- **Gemini-1.5 Pro** (Reid et al., 2024) is a 1.5T closed-source multimodal chat model.
- **Idefics2** (Laurençon et al., 2024) is an 8B open-source multimodal chat model.
- **Idefics3** is an improved version of Idefics2.
- **LLaVA-NeXT** (Liu et al., 2024) is an open-source multimodal LLaVA model with Mistral-7B-Instruct-v0.2.
- **InstructBLIP** (Dai et al., 2024) is an open-source vision-language model that uses BLIP-2 with Vicuna-7b.
- **Molmo** (Deitke et al., 2024) is a family of open-source multimodal language models; we benchmark Molmo-7B-O and Molmo-7B-D.

4.1 Multiple Choice Experiments

We benchmark all models using the multiple choice format of *GlyphPattern*. We use a chat template for chat models, with a system prompt explaining the task. For non-chat models, we suffix the instructions to the prompt.

Answers-Only Baseline In multiple choice tasks, information about the correct answer may be gleaned from the answer options. We measure this in a pre-experiment: we prompt GPT-4o with the answer choices alone, excluding the question and using an empty image. Accuracy is close to random. We call this the *Answers-Only* baseline.

Zero-Shot Prompting We conduct zero-shot experiments by presenting the model with question and answer options, followed by the image.

Few-Shot Prompting For models that support few-shot prompting, we explore prompting with one, three, and five examples. We format the few-shot prompt as a multi-turn message series. To avoid biasing the model, we use as examples patterns that do not reoccur in other writing systems. For example, we use a rule from the Zanabazar Square script, *Characters that contain at least one enclosed rectangle*, rather than the commonly-occurring rule *Characters with enclosed loop or circle*.

Chain-of-Thought Reasoning We also explored chain-of-thought reasoning in two models: GPT-4o and Gemini-1.5; details are in Appendix C.3.1.

²We also experimented with Kosmos-2 (Peng et al., 2024) and OpenFlamingo (Awadalla et al., 2023), but found that they did not understand the multiple choice task (Appendix C.1).

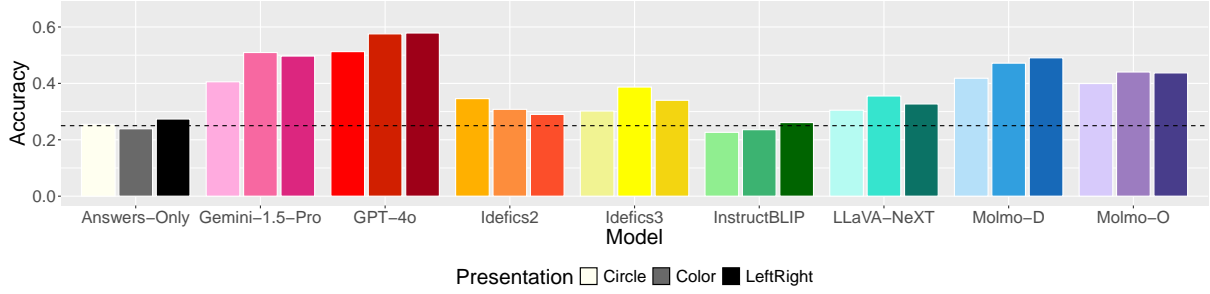


Figure 3: Model results by visual style. Random baseline is plotted as a dashed line.

Model	0-shot	1	3	5
Gemini-1.5	47.1	52.0	53.7	53.4
Gemini-1.5 CoT	46.6	-	39.3	-
GPT-4o	55.6	53.4	55.4	52.9
GPT-4o CoT	50.0	-	49.1	-
Idefics2	31.5	30.1	31.2	30.0
Idefics3	34.3	33.5	32.7	33.3
InstructBLIP	24.1	-	-	-
LLaVA-NeXT	32.9	27.0	28.0	26.8
Molmo-O	42.6	-	-	-
Molmo-D	46.0	-	-	-

Table 1: Zero-shot and few-shot accuracy by model.

4.2 Free Response Pattern Generation

We also conduct a pattern description generation experiment with the two best-performing models. We prompt models with three image/pattern pairs and ask them to produce a rule for a fourth image.

5 Results

We use GlyphPattern to measure abstract pattern understanding in eight VLMs. We test statistical reliability at a significance threshold of $\alpha=0.05$ using mixed-effects logistic regression models for each VLM (see Appendix D.7).

5.1 Zero-shot Results

We benchmark eight VLMs on GlyphPattern using the zero-shot multiple choice format. The benchmark is challenging for all models: even the best model, GPT-4o, selects the correct pattern little more than half of the time, showing that there is significant room for improvement in the abstract pattern recognition abilities of VLMs.

Table 1 shows the zero-shot multiple choice results. GPT-4o is the best-performing model, followed by Gemini-1.5. Molmo-D is competitive with Gemini-1.5. The open-source Idefics2,

Model	Circle	Color	L/R	Total
GPT-4o	12.3	18.1	17.2	15.9
Gemini-1.5	6.9	8.3	9.9	8.4

Table 2: Validity of generated patterns by visual styles.

Idefics3, and LLaVA-NeXT models perform less well, but still better than chance. InstructBLIP does not outperform the answers-only baseline.

Figure 3 displays the zero-shot results by visual style. For most models, the circular partition is the most challenging, with a particularly substantial performance gap for Gemini-1.5 ($\sim 10\%$; $p=0.006$). The difference is smaller but statistically reliable for GPT-4o ($\sim 6\%$; $p=0.03$). For Idefics2, the left-right style is slightly more challenging.

5.2 Does Few-shot Prompting Help?

Table 1 shows the few-shot multiple choice results for models that support it.³ The impact of few-shot prompting is inconsistent: GPT-4o shows no improvement and performance drops for Idefics2, Idefics3, and LLaVA-NeXT. However, few-shot prompting benefits Gemini-1.5, bringing its performance close to GPT-4o. The impact was greater for Gemini-1.5’s performance on the circle and left-right visual styles. Surprisingly, more examples do not always help: Gemini-1.5 does best with three.

The inconsistent gains that we observe for few-shot prompting may be surprising, but similar findings have been reported for other VLM benchmarks, such as Wu et al. (2024a). Wu et al. (2024a) find that Gemini benefits from few-shot prompting, while Idefics and GPT-4V do not. Given the converging evidence from different benchmarks, we think that this indicates an actual limitation of existing VLMs.

³Molmo and InstructBLIP do not support few-shot.

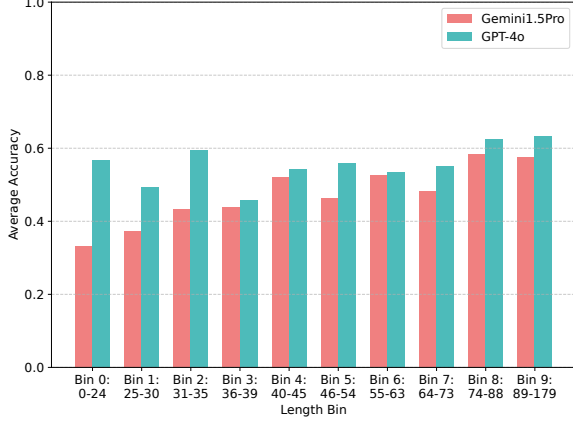


Figure 4: Accuracy by pattern length.



Figure 5: Accuracy by pattern parse tree depth.

5.3 Does Chain-of-Thought Reasoning Help?

We also tested whether chain-of-thought reasoning improved the performance of the best models, Gemini-1.5 and GPT-4o. We find that it actually hurts performance (Table 1), likely because, as we discuss in Section 6.5, models often produce incorrect reasoning.

5.4 Can Models Describe Patterns?

The multiple choice version of GlyphPattern explores whether VLMs can *understand and assess* patterns described in natural language. We can also use GlyphPattern to assess whether models can *describe* patterns.

We explore the pattern description ability of the two best-performing VLMs, GPT-4o and Gemini-1.5, in a generation task. We prompt models with three image/pattern pairs and ask them to produce a rule for a fourth image. The authors independently assessed the validity of the patterns, with inter-annotator agreement of $\kappa=0.89$ for GPT-4o and 0.88 for Gemini-1.5.

Table 2 presents the results, averaged over annotators. We find that GPT-4o is able to generate valid patterns for just 15.9% of the images, and Gemini-1.5 for 8.4%. We also find that models struggle with the same visual styles in pattern generation as in the multiple-choice task.

Manual inspection reveals that models typically generate patterns similar to human ones: both models generated rules related to enclosed spaces, symmetry, and line type, which are common in GlyphPattern. Despite generating human-like rules, models struggle to apply them. This points to issues in pattern recognition or in text/image mapping.

6 Exploring Model Errors

We have shown that GlyphPattern is a challenging benchmark for VLMs. The rich human-written rules and multiple visual styles of GlyphPattern allow us to explore how various aspects of the task impact performance. In this section, we highlight key challenges that VLMs face in abstract pattern recognition, focusing on the two best-performing models, Gemini-1.5 and GPT-4o.

6.1 Are Complex Descriptions Harder?

One potential source of error is in understanding textual descriptions of patterns. A model may be able to distinguish visual properties of included versus excluded characters, but fail to understand how they are described in text. To explore this possibility, we investigate the *complexity* of the pattern descriptions. Are simpler descriptions easier?

Figure 4 plots the performance of the two best-performing models against the number of characters in each rule as a proxy for complexity. Contrary to expectations, we observe that accuracy increases slightly with description length, suggesting that longer patterns are not more challenging. Mixed-effects models reveal a weak but statistically reliable correlation for both Gemini-1.5 ($\hat{\beta}=0.003$; $p=0.001$) and GPT-4o ($\hat{\beta}=0.003$; $p=0.002$).

Another possibility is that models struggle with compositional language in the patterns. We quantify the syntactic complexity of each rule as the maximum depth of their dependency parse tree. Figure 5 shows the performance of Gemini-1.5 and GPT-4o by parse tree depth. We see no strong correlation in performance for either model, though GPT-4o performs best on very simple rules.

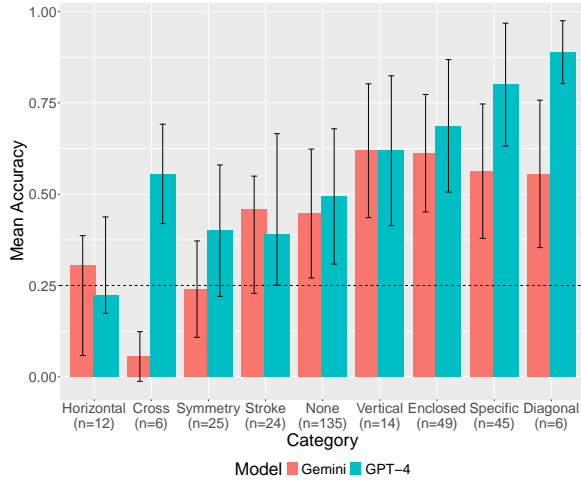


Figure 6: Accuracy by cross-linguistic pattern

6.2 Do Models Understand Cross-linguistically Common Patterns?

The *Glyph* project aims to explore cross-linguistic universals in visual properties of human writing systems. Towards this end, they have identified categories of rules that players commonly propose across languages. These categories represent visual properties of writing systems that are salient for humans. Are they also visually salient to VLMs?

Figure 6 plots performance by rule category for the two best-performing models. We observe significant variation in accuracy by rule category, showing that many rule categories commonly identified by humans pose challenges for models.

The easiest kind of patterns for GPT-4o involve specific character mentions: for instance, *Resemble the letter E*. Compared to rules that did not belong to a category, accuracy was higher ($\hat{\beta}=0.37$; $p<0.001$). For Gemini-1.5, the easiest pattern was vertical lines ($\hat{\beta}=0.28$; $p=0.009$).

One surprising finding is the large difference for both models between vertical and horizontal lines. Accuracy was low in the horizontal category for both models, though only significantly different for GPT-4o (Gemini-1.5: $\hat{\beta}=-0.08$; $p=0.45$; GPT-4o: $\hat{\beta}=-0.23$; $p=0.03$). It is not clear why this line orientation is so challenging. Our results show that symmetry is also a difficult property.

It is less surprising that models struggle with the Stroke category, which describes how many pen strokes it would take to draw the character. These rules are grounded in the experience of drawing letters by hand, which VLMs lack.

6.3 What Patterns Do Models Produce Well?

Our free response experiment allows us to examine what kinds of patterns Gemini-1.5 and GPT-4o are most successful at generating. As discussed in Section 5.4, models often generate rules using similar language as humans even though their rules are usually incorrect for the target image.

We categorize the valid patterns generated by each model according to the cross-linguistically common categories from Section 6.2 (full results in Appendix D.5). We find that the most common correctly generated patterns are ones that refer to enclosed spaces or specific characters. Gemini-1.5 is more successful at identifying symmetry than GPT-4o.

Qualitatively, we note that models seem to struggle especially to generate relative spatial terms. There are many generated patterns that would be correct if “horizontal” and “vertical” or “left” and “right” were swapped. This suggests that relative spatial language remains a challenge. Alternatively, these struggles could spring from a loss of spatial information in image-processing.

6.4 Which Writing Systems Are Challenging?

We also explore whether certain writing systems are more challenging for VLMs. Figure 7 shows the performance of Gemini-1.5 and GPT-4o by writing system. We see that some scripts are harder for them than others: GPT-4o struggles with Grantha ($\hat{\beta}=-0.4$; $p=0.02$) and Meroitic Cursive ($\hat{\beta}=-0.5$; $p=0.03$). This suggests that the visual properties of these writing systems are less salient for VLMs.

6.5 How Does Reasoning Go Awry?

When VLMs make mistakes, they can stem from issues in natural language understanding, image processing, or the vision-language mapping. To explore sources of error in GlyphPattern decisions, we prompt Gemini-1.5 and GPT-4o to explain their reasoning for the multiple-choice task and manually inspect the results. We find issues stemming from multiple aspects of the task:

Errors in image-processing In one case, GPT-4o correctly identifies the four excluded characters when presented in the color style. However, in the circle style, GPT-4o lists all 22 characters in the script as falling inside the circle, indicating an error in recognizing the circle itself (see Appendix D.2). This is similar to the image-processing errors in Rahmanzadehgervi et al. (2024).

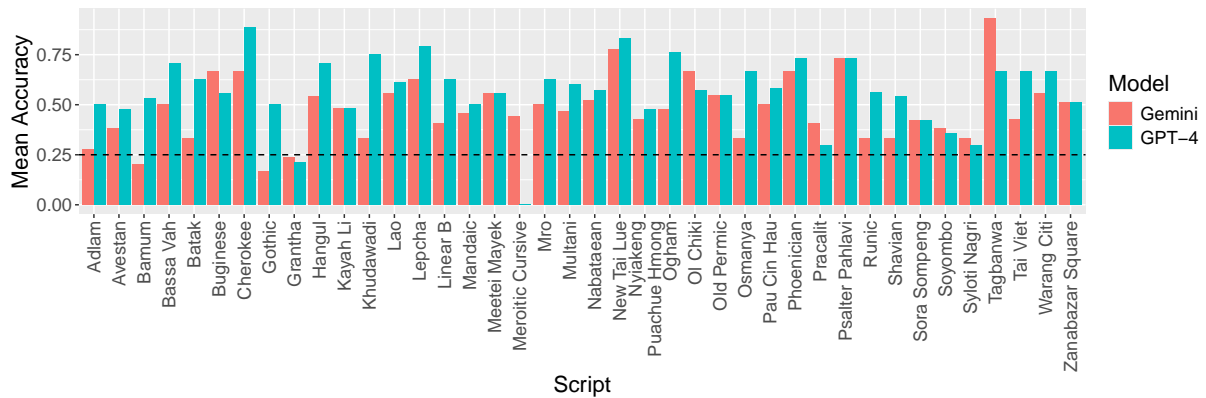


Figure 7: Accuracy by writing system.

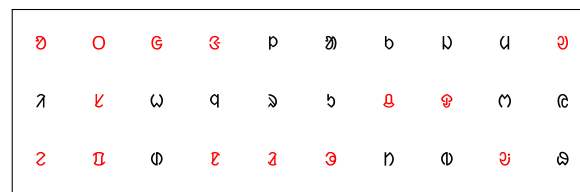


Target Pattern: Characters that have exactly one closed end.

Gemini-1.5: The characters ‘S’, ‘2’, ‘C’, and ‘3’ have exactly one closed end, but they are not colored red.

GPT-4o: The red characters do not have exactly one closed end. They have loops, which means they have more than one closed end.

Figure 8: Ex. natural language understanding failure



Target Pattern: Characters that have a curved bottom.

Gemini-1.5: While some red characters have curved bottoms, others do not. For example, the red character “O” does not have a curved bottom.

GPT-4o: Not all red characters have a curved bottom. For example, the character in second row, second column from the left (which looks like an upside-down “L”) does not have a curved bottom.

Figure 9: Ex. incorrect character property reasoning

Errors in natural language understanding In other cases, models fail to understand the language that human pattern writers use. For the example in Figure 8, GPT-4o is able to map the included characters to their Unicode symbols and identify that these characters contain loops. However, GPT-4o rejects the correct option because it does not understand “closed end” as the human writer intended. Gemini-1.5 also misunderstands this description and incorrectly identifies it as applying to multiple characters in the excluded set.

Text/image mapping issues Models sometimes fail to map the properties described in natural language to the character images. For the curved bottom pattern in Figure 9, Gemini-1.5 correctly picks out a red character that is similar to “O”, but does not seem to understand the visual properties of the character or its Latin equivalent. GPT-4o makes a similar mistake for the same pattern, incorrectly deciding that the “L”-like character lacks a curved

bottom but does have “enclosed spaces or loops” (one of the distractor options). Here, the models either fail to understand the property description, or to map it to the visual properties of the characters.

Reasoning errors Other errors arise from failures to reason about the excluded characters. For instance, GPT-4o chooses the pattern *Characters that look like Latin capital letters* for the item in Figure 2. Its explanation (*The characters on the right side, “U” and “W”, look like Latin capital letters*) correctly judges that the included characters fit the pattern, but fails consider that excluded characters do too.

Thus, GlyphPattern shows that VLM errors arise across the image-processing, text-image mapping, and reasoning aspects of the abstract pattern recognition task.

7 Conclusion

We present GlyphPattern, an abstract pattern recognition benchmark consisting of 954 items in three visual styles. GlyphPattern challenges models to understand visually-presented patterns paired with human descriptions, revealing significant gaps in VLM capabilities. We find that even the best performing models, Gemini-1.5 and GPT-4o, solve around half of GlyphPattern problems.

GlyphPattern is based on human-written descriptions of visual properties of writing systems, facilitating the exploration of multiple task aspects, from pattern length, complexity, and visual presentation style, to differences across cross-linguistically common pattern types and human writing systems. Our detailed analysis reveals errors at multiple levels of processing, including image-processing, natural language understanding, and reasoning. We hope that the GlyphPattern benchmark will guide future work on improving VLMs by providing a challenging, targeted and diverse evaluation set.

Limitations

Our work reveals weaknesses in abstract pattern recognition for VLMs. GlyphPattern moves closer to Bongard’s original task by requiring models to understand natural language pattern descriptions; however, evaluating the ability to describe patterns is challenging due to the lack of automated scoring. Automating this process is unlikely to succeed in the near-term: since our benchmark items are challenging for state-of-the-art VLMs, using LLMs to score the rules would not be reliable.

Our few-shot evaluation of Idefics2 was limited by GPU memory constraints. We also excluded two VLMs due to inability to understand the task format. It is possible that they could perform well in other abstract pattern recognition tasks, but ideally, models would be robust to task format.

Ethics Statement

Our work does not involve any data collection from human subjects. We rely on publicly-available human data from the Glyph project, which was originally collected under the oversight of the Ethikkommission affiliated with Universitätsklinikum, ensuring adherence to ethical guidelines and standards.

Our findings have implications for the visual salience of character sets across human writing systems, and we hope that they complement ongoing cognitive scientific efforts to understand the visual

properties of human writing systems. Although our results show that some writing systems are more challenging for VLMs, this should not be misinterpreted to suggest that some writing systems are better or more efficient than others.

Acknowledgements

We thank the ARR reviewers for their thoughtful feedback. This work is partially supported by the U.S. National Science Foundation (SES-2326174).

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- AI@Meta. 2024. [Llama 3.2 model card vision](#).
- Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenna Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. 2023. [Openflamingo: An open-source framework for training large autoregressive vision-language models](#). *Preprint*, arXiv:2308.01390.
- David G. T. Barrett, Felix Hill, Adam Santoro, Ari S. Morcos, and Timothy Lillicrap. 2018. Measuring abstract reasoning in neural networks. In *International Conference on Learning Representations (ICLR)*.
- Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. [Fitting linear mixed-effects models using lme4](#). *Journal of Statistical Software*, 67(1):1–48.
- Mikhail Moiseevich Bongard. 1967. *Pattern Recognition*. Hayden Book Co., Spartan Books.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2024. Instructblip: towards general-purpose vision-language models with instruction tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS ’23*, Red Hook, NY, USA. Curran Associates Inc.
- Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris

- Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Witliff, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*.
- Li Deng. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.
- Anuj Diwan, Layne Berry, Eunsol Choi, David Harwath, and Kyle Mahowald. 2022. [Why is winoground hard? investigating failures in visuolinguistic compositionality](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2250, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Drew A. Hudson and Christopher D. Manning. 2019. [Gqa: A new dataset for real-world visual reasoning and compositional question answering](#). In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702.
- Anyu Ji, Noriyuki Kojima, Noah Rush, Alane Suhr, Wai Keen Vong, Robert Hawkins, and Yoav Artzi. 2022. [Abstract visual reasoning with tangram shapes](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 582–601, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Huaizu Jiang, Xiaojuan Ma, Weili Nie, Zhiding Yu, Yuke Zhu, and Anima Anandkumar. 2022. [Bongard-hoi: Benchmarking few-shot visual reasoning for human-object interactions](#). In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19034–19043.
- Amita Kamath, Jack Hessel, and Kai-Wei Chang. 2023. [What’s “up” with vision-language models? investigating their struggle with spatial reasoning](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9161–9175, Singapore. Association for Computational Linguistics.
- Yoolim Kim, Helena Miton, Marc Allasonnière-Tang, and Olivier Morin. 2024. [The glyph project: The distinctiveness of written characters — online crowdsourcing for a typology of letter shapes](#).
- Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.
- Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*.
- Fangyu Liu, Guy Emerson, and Nigel Collier. 2023. [Visual spatial reasoning](#). *Transactions of the Association for Computational Linguistics*, 11:635–651.
- Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. [Llava-next: Improved reasoning, ocr, and world knowledge](#).
- Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. 2023. Crepe: Can vision-language foundation models reason compositionally? *arXiv preprint arXiv:2212.07796*.
- Nimish Magre and Nicholas Brown. 2022. Typography-mnist (tmnist): an mnist-style image dataset to categorize glyphs and font-styles.
- Weili Nie, Zhiding Yu, Lei Mao, Ankit B Patel, Yuke Zhu, and Animashree Anandkumar. 2020. Bongard-logo: A new benchmark for human-level concept learning and reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shao-han Huang, Shuming Ma, Qixiang Ye, and Furu Wei. 2024. [Grounding multimodal large language models to the world](#). In *The Twelfth International Conference on Learning Representations*.
- Pooyan Rahmazadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. 2024. [Vision language models are blind](#). *Preprint*, arXiv:2407.06581.
- John C. Raven. 1936. Mental tests used in genetic studies: The performance of related individuals on tests mainly educative and mainly reproductive. Master’s thesis, University of London.
- Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A. Plummer, Ranjay Krishna, and Kate Saenko. 2023. [Cola: A benchmark for compositional text-to-image retrieval](#). In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Steven Spratley, Krista A. Ehinger, and Tim Miller. 2023. [Unicode analogies: An anti-objectivist visual reasoning challenge](#). In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19082–19091.
- Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. [A corpus of natural language for visual reasoning](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, Vancouver, Canada. Association for Computational Linguistics.

- Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. [A corpus for reasoning about natural language grounded in photographs](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.
- Damien Teney, Peng Wang, Jiewei Cao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. 2020. V-prom: A benchmark for visual reasoning using visual progressive matrices. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12071–12078.
- T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela, and C. Ross. 2022. [Winoground: Probing vision and language models for visio-linguistic compositionality](#). In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5228–5238, Los Alamitos, CA, USA. IEEE Computer Society.
- Anne Wu, Kianté Brantley, and Yoav Artzi. 2024a. [A surprising failure? multimodal llms and the nlvr challenge](#). *Preprint*, arXiv:2402.17793.
- Rujie Wu, Xiaojian Ma, Zhenliang Zhang, Wei Wang, Qing Li, Song-Chun Zhu, and Yizhou Wang. 2024b. [Bongard-openworld: Few-shot reasoning for free-form visual concepts in the real world](#). *Preprint*, arXiv:2310.10207.
- Xiang Yue, Yuansheng Ni, Tianyu Zheng, Kai Zhang, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. 2024. [Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi](#). In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567.
- Aimen Zerroug, Mohit Vaishnav, Julien Colin, Sebastian Musslick, and Thomas Serre. 2022. A benchmark for compositional visual reasoning. *Advances in neural information processing systems*, 35:29776–29788.
- Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. 2019. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

A Checklist

A2 Potential Risks: Did you discuss any potential risks of your work? [Yes/No/NA] **No**

A2 Elaboration: For yes, provide a section number. For no, justify why not. **The risks of this work lie in the original dataset creation and collection, as described by Kim et al. (2024).**

B Use Or Create Scientific Artifacts: Did you use or create scientific artifacts? [Yes/No] **Yes**

B1 Cite Creators Of Artifacts: Did you cite the creators of artifacts you used? [Yes/No/NA] **Yes**

B1 Elaboration: For yes, provide a section number. For no, justify why not. **Section 3.**

B2 Discuss The License For Artifacts: Did you discuss the license or terms for use and/or distribution of any artifacts? [Yes/No/NA] **Yes**

B2 Elaboration: For yes, provide a section number. For no, justify why not. **Section B.1**

B3 Artifact Use Consistent With Intended Use: Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions? [Yes/No/NA] **Yes**

B3 Elaboration: For yes, provide a section number. For no, justify why not. **Ethics Statement**

B4 Data Contains Personally Identifying Info Or Offensive Content: Did you discuss the steps taken to check whether the data that was collected/used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect/anonymize it? [Yes/No/NA] **NA**

B4 Elaboration: For yes, provide a section number. For no, justify why not.

B5 Documentation Of Artifacts: Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.? [Yes/No/NA] **Yes**

B5 Elaboration: For yes, provide a section number. For no, justify why not. **Section D.7.**

B6 Statistics For Data: Did you report relevant statistics like the number of examples, details of train/test/dev splits, etc. for the data that you used/created? [Yes/No/NA] **Yes**

B6 Elaboration: For yes, provide a section number. For no, justify why not. **Section 3**

C Computational Experiments: Did you run computational experiments? [Yes/No/NA] **Yes**

C1 Model Size And Budget: Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used? [Yes/No/NA] **Yes**

C1 Elaboration: For yes, provide a section number. For no, justify why not. **Section C.2**

C2 Experimental Setup And Hyperparameters: Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values? [Yes/No/NA] **Yes**

C2 Elaboration: For yes, provide a section number. For no, justify why not. **Section 4.**

C3 Descriptive Statistics: Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run? [Yes/No/NA] **Yes**

C3 Elaboration: For yes, provide a section number. For no, justify why not. **Sections 5 and D.7.**

C4 Parameters For Packages: If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation, such as NLTK, SpaCy, ROUGE, etc.), did you report the implementation, model, and parameter settings used? [Yes/No/NA] **Yes**

C4 Elaboration: For yes, provide a section number. For no, justify why not. **Section D.1.**

D Human Subjects Including Annotators: Did you use human annotators (e.g., crowdworkers) or research with human subjects? [Yes/No/NA] **No**

D1 Instructions Given To Participants: Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.? [Yes/No/NA] **NA**

D1 Elaboration: For yes, provide a section number. For no, justify why not.

D2 Recruitment And Payment: Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)? [Yes/No/NA] **NA**

D2 Elaboration: For yes, provide a section number. For no, justify why not.

D3 Data Consent: Did you discuss whether and how consent was obtained from people whose data you're using/curating (e.g., did your instructions explain how the data would be used)? [Yes/No/NA] **NA**

D3 Elaboration: For yes, provide a section number. For no, justify why not.

D4 Ethics Review Board Approval: Was the data collection protocol approved (or determined exempt) by an ethics review board? [Yes/No/NA] **NA**

D4 Elaboration: For yes, provide a section number. For no, justify why not.

D5 Characteristics Of Annotators: Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data? [Yes/No/NA] **NA**

D5 Elaboration: For yes, provide a section number. For no, justify why not.

E Ai Assistants In Research Or Writing: Did you use AI assistants (e.g., ChatGPT, Copilot) in your research, coding, or writing? [Yes/No] **Yes**

E1 Information About Use Of AI Assistants: Did you include information about your use of AI assistants? [Yes/No/NA] **Yes**

E1 Elaboration: For yes, provide a section number. For no, justify why not. **Section E.**

B Further Details of Dataset

B.1 Licensing

The full GlyphPattern dataset and support code are available at <https://github.com/Wellesley-EASEL-lab/GlyphPattern>. The benchmark is licensed for reuse under an MIT license.

The dataset builds on data collected as part of the Glyph cognitive science project, which is publicly available at <https://osf.io/j9zhn>.

B.2 Data Selection and Cleaning

We use the 318 cleaned rules from Kim et al. (2024)’s Replicability Assessment datafile. We further standardize the wording of the rules by hand, such that all are grammatical following the prefix *Characters that*. This is necessary in order to consistently produce grammatical prompts.

We eliminated two scripts included in Glyph, Afaka and Miao, due to difficulties in obtaining font libraries containing their full character lists.

C Further Details of Experiments

C.1 Model Selection

We use the following model versions:

- GPT-4o: gpt-4o-2024-05-13
- Gemini-1.5: gemini-1.5-pro-001
- LLaVA-NeXT: llava-v1.6-mistral-7b-hf
- Idefics2: idefics2-8b
- Idefics3: Idefics3-8B-Llama3
- InstructBLIP: instructblip-vicuna-7b

For Idefics2, we used image scaling parameters that set the longest edge of 700 and a shortest edge of 378. This did not decrease performance compared to the default Idefics2 settings (Table 3).

For Idefics3, we used image scaling parameter that set the longest edge of 1092. This did not decrease performance compared to the default Idefics3 settings (Table 3).

Model	Image Scaling	Accuracy (%)
Idefics2	base	31.45
Idefics2	scaled	31.45
Idefics3	base	33.86
Idefics3	scaled	34.28

Table 3: Zeroshot accuracy for Idefics2 and Idefics3 with and without image scaling.

Model	N	Circle	Color	L/R	Total
Gemini	0	40.6	51.0	49.7	47.1
Gemini	1	49.1	50.6	56.3	52.0
Gemini	3	50.3	52.2	58.5	53.7
Gemini	5	50.6	52.5	57.0	53.4
GPT-4o	0	51.3	57.6	57.9	55.6
GPT-4o	1	48.5	57.6	54.1	53.4
GPT-4o	3	49.7	57.6	59.1	55.4
GPT-4o	5	45.9	58.5	54.4	52.9
Idefics2	0	34.6	30.8	29.0	31.5
Idefics2	1	33.0	30.8	26.4	30.1
Idefics2	3	34.0	29.9	29.9	31.2
Idefics2	5	33.0	30.5	26.4	30.0
Idefics3	0	30.2	38.7	34.0	34.3
Idefics3	1	31.5	32.7	36.5	33.5
Idefics3	3	32.7	34.6	30.8	32.7
Idefics3	5	33.3	35.0	31.8	33.3
LLaVA	0	30.5	35.5	32.7	32.9
LLaVA	1	28.3	24.8	28.0	27.0
LLaVA	3	29.6	27.0	27.4	28.0
LLaVA	5	26.4	26.4	27.7	26.8

Table 4: Few-shot accuracy by number of examples (N)

We attempted to benchmark Kosmos-2 (Peng et al., 2024) and OpenFlamingo (Awadalla et al., 2023), but found that they struggled with the multiple choice task format. Both models tended to produce additional answer options rather than selecting an option from among the four choices. Adding the task explanation to the end did not address this issue; instead, Kosmos-2 generated a new explanation without an answer option, and OpenFlamingo began generating a new set of answers starting at A.

We also attempted to benchmark Llama3.2 Vision (AI@Meta, 2024), but found that it is safeguard to the extend of not providing valid response for our test. Instead, the model respond "I’m not able to provide that information".

C.2 Computing Resources

Experiments were run with greedy decoding on a single A100 GPU. We estimate that the experiments reported in this paper consumed less than 400 GPU hours; however, we do not know how much GPU time was used to run the proprietary models Gemini-1.5 and GPT-4o.

C.3 Further Prompting Details

We ran Idefics2, Idefics3, LLaVA-NeXT, and InstructBLIP using Hugging Face’s AutoModelForVision2Seq API, following the format of their model card.

System Prompts For multiple choice experiments, we query the model with system prompt: *Reply with only the letter of the correct option.* The system prompt is provided to GPT-4o and Gemini-1.5 through their APIs. We provide no system prompt for Idefics2 or Idefics3. For LLaVA-NeXT, and InstructBLIP zero-shot experiments, we concatenated the system prompt at the end of the question prompt for zero-shot experiments. We do not provide a system prompt for their few-shot experiments.

For few-shot free response experiments, we prompt GPT-4o and Gemini-1.5 with the system instruction: *Come up with a descriptions for all of the characters {on the right} side in the image. Your description should be true for all of the characters {on the right side} and none of the characters {on the left side}, reworded by the image’s visual style.*

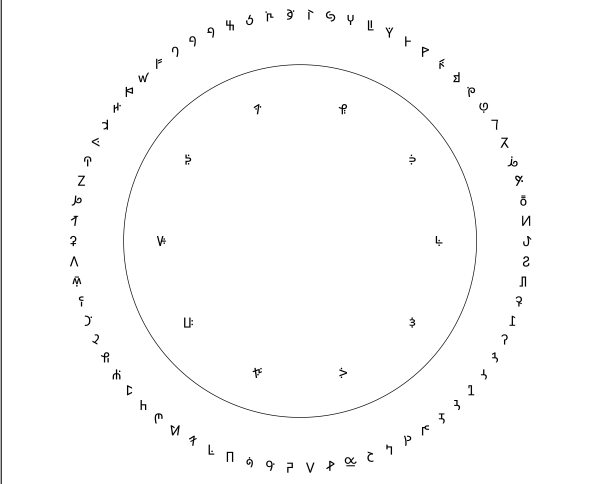
Few-shot Prompting For Idefics2, Idefics3, LLaVA-NeXT, and Gemini-1.5, the image is presented before the question, followed by the answer. For GPT-4o, the question is shown first, then the image and the answer.

Figure 10 shows an example of few-shot multiple choice prompting with one sample question and answer pair.

Figure 11 is an example of few-shot free response.

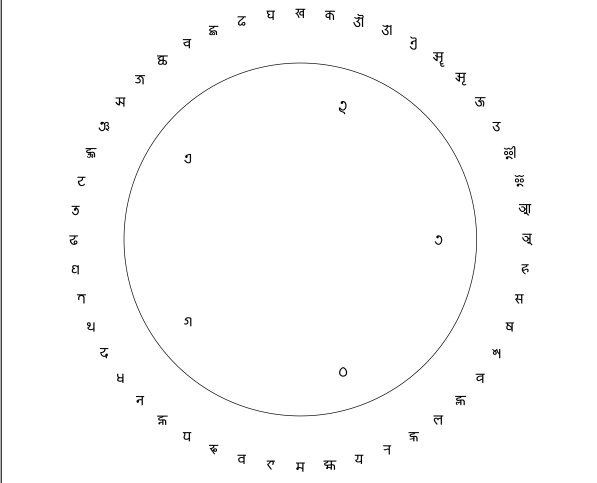
C.3.1 Chain-of-Thought Reasoning Prompting

We ran zero-shot and three-shot experiments with chain-of-thought prompting. Two example reasoning chains used for chain-of-thought few-shot prompting are shown in Figure 12.



Which characters are inside the circle in the image? A. Characters that contain a horizontal line and a descending curve from one end of the line. B. Characters that have a downward curve that resembles a bridge with two feet. C. Characters that have two round dots. D. Characters that contain diagonal lines and acute angles, but no flags.

MODEL:C.



Which characters are colored red in the image? A. Characters that contain a hook shape. B. Characters that start at the top left with a small or very small spiral-like approach. C. Characters that can be written with a single continuous stroke without retracing or lifting the pen. D. Characters that have vertical and horizontal symmetry.

MODEL:

Groundtruth:C.

Gemini-1.5 Response:C.

GPT-4o Response:D.

Figure 10: Few-shot Multiple Choice Example

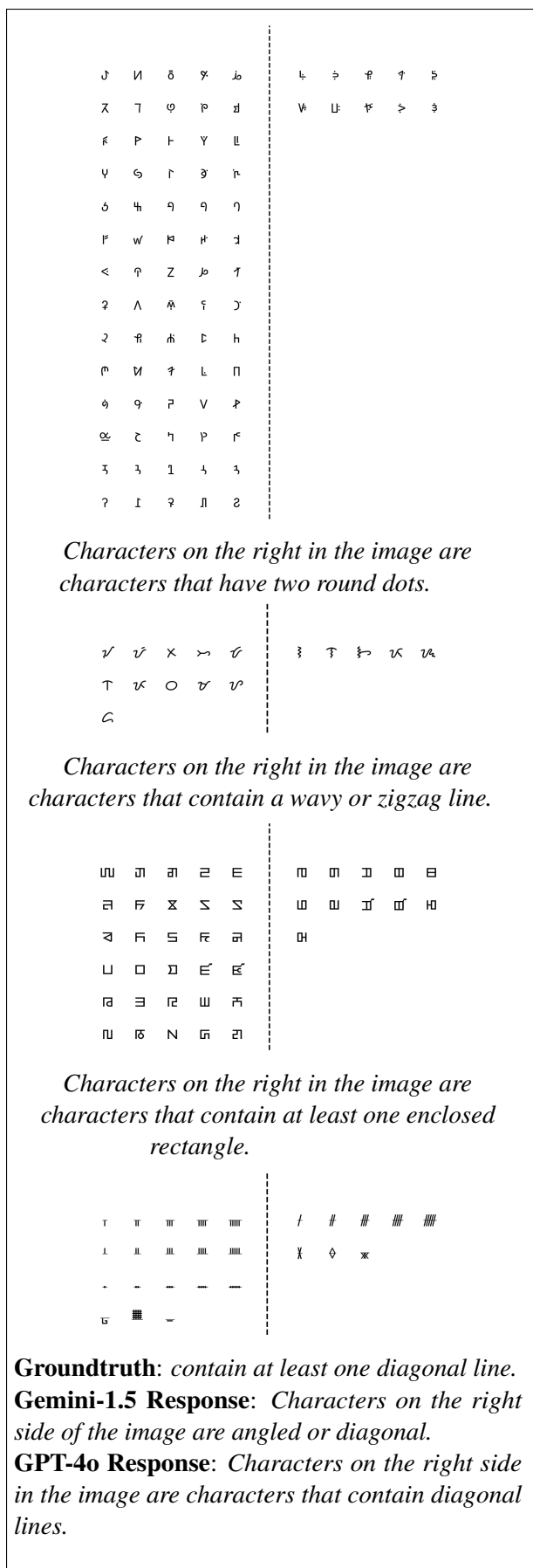


Figure 11: Few-shot Free Response Example

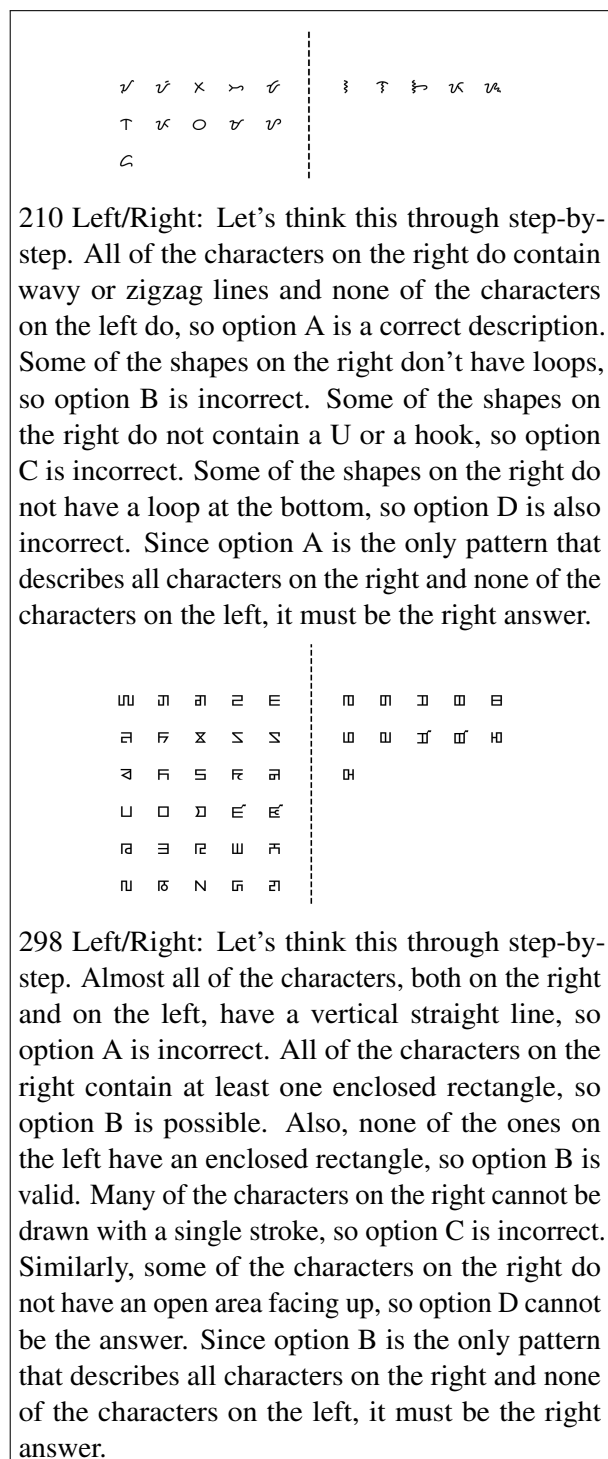


Figure 12: Chain-of-Thought Reasoning Examples for Few-shot Prompting

D Further Analysis

D.1 Pattern Complexity

We compute the syntactic complexity of a pattern by taking the maximum depth of its dependency parse tree. We parse the pattern descriptions using SpaCy (“en_core_web_lg” model).

D.2 Error Analysis

Figure 13 illustrates a case where the circular visual style poses a challenge for the models. GPT-4o performed correct visual recognition of the excluded character sets on both the color and leftright visual style, but failed on the circle visual style.

D.3 In-Depth Few-shot Results

Table 4 shows few-shot results by model and visual style.

D.4 In-Depth Chain-of-Thought Results

Table 5 shows the chain-of-thought reasoning results by model and visual style.

Model	N	Circle	Color	L/R	Total
GPT-4o	0	45.0	54.7	50.3	50.0
GPT-4o	3	40.3	51.9	55.0	49.1
Gemini-1.5	0	39.3	50.6	50.0	46.6
Gemini-1.5	3	35.2	39.0	43.7	39.3

Table 5: Chain-of-Thought Prompting Results by Model

D.5 Free Response Task

A passing pattern is true of all included characters and false of all excluded characters. In borderline cases, we ask whether the rule applies equally well to a character from each set; if so, it fails, since a human Glyph player would be unlikely to pick only the correct characters. We measure inter-annotator agreement using Cohen’s κ . Figures 14 and 15 show the correct patterns generated by GPT-4o and Gemini-1.5 in the free response task, grouped by the cross-linguistically common pattern categories from Section 6.2.

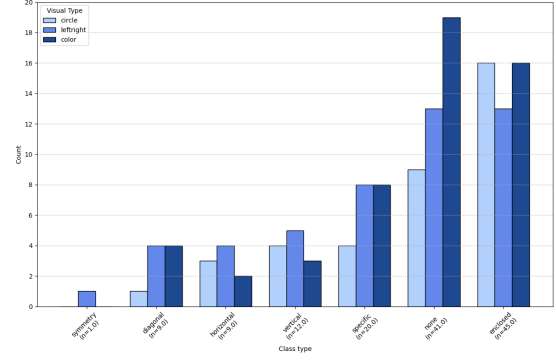


Figure 14: GPT-4o free response results by rule class.

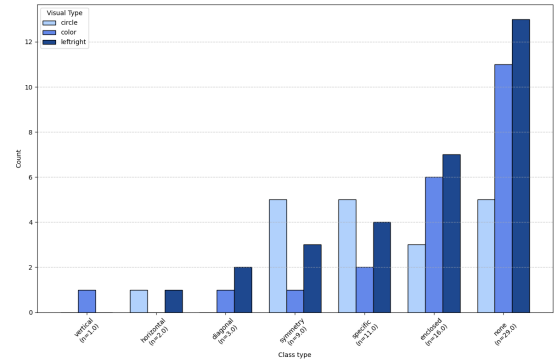


Figure 15: Gemini-1.5 free response results by rule class.

D.6 Explanation Prompting

To elicit model explanations for our manual error analysis, we prompt Gemini-1.5 and GPT-4o for an explanation after their answer: *Reply with the letter of the correct option, and then explain your step-by-step reasoning through all options.* Figure 16, 17, 18, 19, and 20 are five examples showing how the models go awry in their explanations. The answer choices can be seen in the models’ response.

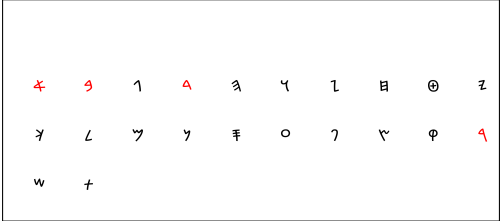
D.7 Logistic Regression Results

We test the statistical reliability of various factors that impact model performance using mixed-effects logistic regression models. We treat correctness as a binary outcome and include item random effects. We include fixed effects for visual style, pattern length, pattern category (using the cross-linguistically common patterns discussed in Section 6.2), and writing system. We fit models to each VLM’s results using the lme4 package in R (Bates et al., 2015). Tables 6-21 show the results. We split the main effects and language effects into separate tables for length reasons, but all effects were fitted as part of the same model.

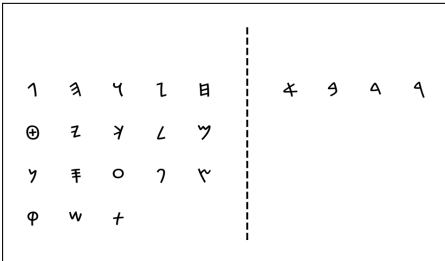
E Use of Generative AI

Some code for this paper was written with AI assistants enabled.

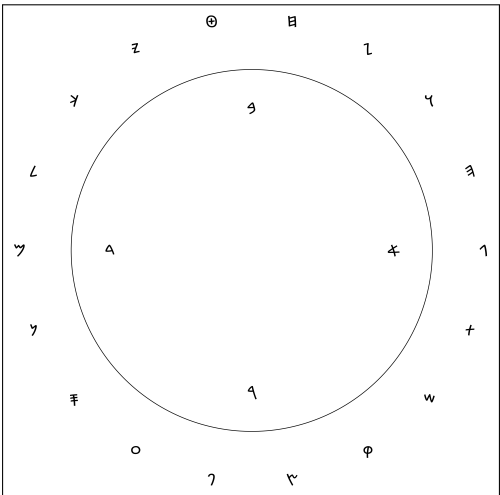
Target Pattern: Characters that have a rough triangle pointing lower-left



GPT-4o Color Explanation: The red characters in the image all have a shape that resembles a triangle pointing towards the lower-left direction.



GPT-4o Leftright Explanation: All the characters on the right side have a rough triangle pointing lower-left. The characters resemble "4", "9", and "A", which all have a triangular shape pointing towards the lower-left.



GPT-4o Circle Explanation: The characters inside the circle are 4, 9, 1, 8, 0, 2, 7, L, 7, 7, 8, 0, 7, 7, 0. They do not all have a rough triangle pointing lower-left.

Figure 13: Example of image-processing error

୧	୦	୨	୩	୪	୫	୬	୭	୮	୯
୧୦	୧୧	୧୨	୧୩	୧୪	୧୫	୧୬	୧୭	୧୮	୧୯
୨୦	୨୧	୨୨	୨୩	୨୪					

Groundtruth:"B.Characters that contain at least one right angle."

Gemini-1.5 Response:"The answer is ****D****."

****A.** Characters that contain a circular element.****** Incorrect, while some red characters contain circles, some do not. ****B.** Characters that contain at least one right angle.****** Incorrect, several red characters lack right angles. ****C.** Characters that exceed the average size of the character set in height or width.****** Incorrect, the red characters vary in size and are not all larger than the average. ****D.** Characters that have a right turn at the bottom of a vertical line, like an L or reversed J.****** Correct, all the red characters feature a right turn at the bottom of a vertical line. Therefore, the characters colored red are those that have a right turn at the bottom of a vertical line, like an L or reversed J. This pattern is consistently observed in all the red characters. "

GPT-4o Response:"The correct option is B. Characters that contain at least one right angle.

Here’s the step-by-step reasoning:

A. Characters that contain a circular element. - Some of the red characters do not contain circular elements. For example, the red character in the first row, fourth column from the left, does not have a circular element. Therefore, this option is incorrect.

B. Characters that contain at least one right angle. - All the red characters contain at least one right angle. For example, the red character in the first row, fourth column from the left, has a right angle. This option is correct.

C. Characters that exceed the average size of the character set in height or width. - Not all red characters exceed the average size of the character set. For example, the red character in the second row, first column from the left, does not exceed the average size. Therefore, this option is incorrect.

D. Characters that have a right turn at the bottom of a vertical line, like an L or reversed J. - Not all red characters have a right turn at the bottom of a vertical line. For example, the red character in the second row, fourth column from the left, does not have a right turn at the bottom of a vertical line. Therefore, this option is incorrect.

Thus, the correct answer is B."

Figure 16: Multiple Choice Explanation Example 1

𐀀	𐀁	𐀂	𐀃	𐀄	𐀅	𐀆	𐀇	𐀈	𐀉
𐀊	𐀋	𐀌	𐀍	𐀎	𐀏	𐀐	𐀑	𐀒	𐀓
𐀔	𐀕	𐀖	𐀗	𐀘	𐀙	𐀚	𐀛	𐀜	𐀝
𐀞	𐀟	𐀠	𐀡	𐀢	𐀣	𐀤	𐀥	𐀦	𐀧
𐀨	𐀩	𐀪	𐀫	𐀬	𐀭	𐀮	𐀯	𐀰	𐀱
𐀲	𐀳	𐀴	𐀵	𐀶	𐀷	𐀸	𐀹	𐀺	𐀻

Groundtruth: "D. Characters that have a closed loop or circular element, which may be crossed over itself in a swirl motion or formed by a line bisecting itself."

Gemini-1.5 Response: "The answer is **D. Characters that have a closed loop or circular element, which may be crossed over itself in a swirl motion or formed by a line bisecting itself.**"

Here's why:

* **A. Characters that have parallel vertical lines or multiple vertical lines.** While some characters on the right have vertical lines, this isn't a defining characteristic shared by all. * **B. Characters that have a large, open, vertical circle curve on the left side.** This describes some characters on the *left* side of the image, not the right. * **C. Characters that have a full-height straight line on the right side, ending in a vertical line perpendicular to a horizontal line at the base of the character.** This pattern doesn't accurately describe any of the characters on the right.

****D is the most accurate:**** Every character on the right side features a closed loop or circle:

* Some loops are simple circles (like the third character). * Some are more complex, with lines crossing through (like the first and last characters). * The second character's loop is formed by a line bisecting itself. "

GPT-4o Response: "The correct option is D. Let's analyze each option:

A. Characters that have parallel vertical lines or multiple vertical lines. - The characters on the right side do not predominantly feature parallel vertical lines or multiple vertical lines. This option does not match.

B. Characters that have a large, open, vertical circle curve on the left side. - The characters on the right side do not have a large, open, vertical circle curve on the left side. This option does not match.

C. Characters that have a full-height straight line on the right side, ending in a vertical line perpendicular to a horizontal line at the base of the character. - The characters on the right side do not consistently have a full-height straight line on the right side ending in a vertical line perpendicular to a horizontal line at the base. This option does not match.

D. Characters that have a closed loop or circular element, which may be crossed over itself in a swirl motion or formed by a line bisecting itself. - The characters on the right side do have closed loops or circular elements, some of which are crossed over themselves or formed by a line bisecting itself. This option matches."

Figure 17: Multiple Choice Explanation Example 2

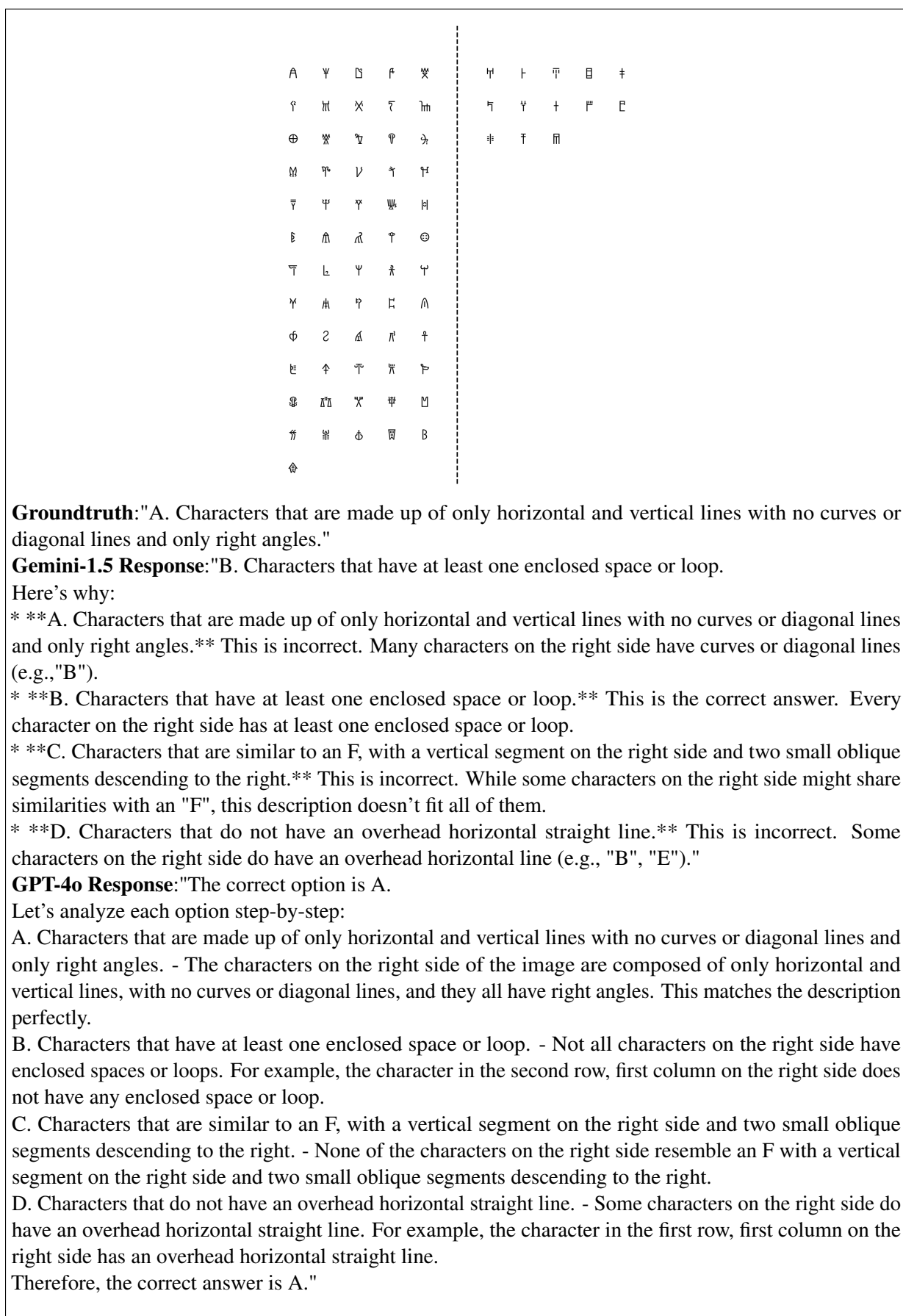


Figure 18: Multiple Choice Explanation Example 3

Fixed effects	$\hat{\beta}$	z	p
(Intercept)	0.12 (\pm 0.15)	0.83	0.41
color	0.10 (\pm 0.03)	3.11	0.002
leftright	0.09 (\pm 0.03)	2.74	0.006
length	0.003 (\pm 0.001)	3.24	0.001
cross	-0.31 (\pm 0.15)	-2.08	0.04
diagonal	0.27 (\pm 0.15)	1.50	0.14
enclosed	0.18 (\pm 0.06)	2.97	0.003
horizontal	-0.08 (\pm 0.11)	-0.76	0.45
mirror	-0.05 (\pm 0.27)	-0.19	0.85
specific	0.13 (\pm 0.06)	2.07	0.04
stroke	0.02 (\pm 0.08)	0.29	0.77
symmetry	-0.17 (\pm 0.08)	-2.16	0.03
vertical	0.28 (\pm 0.11)	2.64	0.009

Table 6: Full logistic regression table for Gemini-1.5, main effects

Fixed effects	$\hat{\beta}$	z	p
(Intercept)	0.27 (\pm 0.15)	1.84	0.066
color	0.06 (\pm 0.03)	2.05	0.041
leftright	0.07 (\pm 0.03)	2.15	0.032
length	0.003 (\pm 0.001)	3.12	0.002
cross	0.13 (\pm 0.15)	0.91	0.37
diagonal	0.46 (\pm 0.15)	3.07	0.002
enclosed	0.24 (\pm 0.06)	4.17	<0.001
horizontal	-0.23 (\pm 0.11)	-2.20	0.028
mirror	-0.24 (\pm 0.27)	-0.89	0.38
specific	0.37 (\pm 0.06)	5.78	<0.001
stroke	-0.13 (\pm 0.08)	-1.56	0.12
symmetry	-0.09 (\pm 0.08)	-1.16	0.25
vertical	0.19 (\pm 0.11)	1.75	0.08

Table 7: Full logistic regression table for GPT-4o, main effects

Fixed effects	$\hat{\beta}$	z	p
Avestan	0.01 (\pm 0.19)	0.08	0.94
Bamum	-0.01 (\pm 0.20)	-0.05	0.96
Bassa Vah	0.17 (\pm 0.18)	0.91	0.37
Batak	0.04 (\pm 0.19)	0.19	0.85
Buginese	0.36 (\pm 0.24)	1.50	0.14
Cherokee	0.24 (\pm 0.18)	1.32	0.19
Gothic	-0.03 (\pm 0.22)	-0.16	0.88
Grantha	-0.19 (\pm 0.17)	-1.16	0.25
Hangul	0.15 (\pm 0.18)	0.81	0.42
Kayah Li	0.11 (\pm 0.18)	0.59	0.55
Khudawadi	-0.05 (\pm 0.22)	-0.23	0.82
Lao	0.14 (\pm 0.17)	0.82	0.42
Lepcha	0.32 (\pm 0.18)	1.77	0.08
Linear B	0.01 (\pm 0.18)	0.05	0.96
Mandaic	0.07 (\pm 0.18)	0.39	0.69
Meetei Mayek	0.17 (\pm 0.17)	1.04	0.30
Meroitic Cursive	0.05 (\pm 0.24)	0.19	0.85
Mro	0.16 (\pm 0.20)	0.80	0.42
Multani	0.10 (\pm 0.17)	0.59	0.56
Nabataean	0.10 (\pm 0.19)	0.55	0.58
New Tai Lue	0.38 (\pm 0.19)	1.94	0.05
Nyiakeng Puachue Hmong	0.05 (\pm 0.19)	0.27	0.79
Ogham	-0.05 (\pm 0.20)	-0.25	0.80
Ol Chiki	0.22 (\pm 0.19)	1.17	0.24
Old Permic	0.15 (\pm 0.17)	0.85	0.40
Osmanya	0.07 (\pm 0.27)	0.25	0.80
Pau Cin Hau	0.23 (\pm 0.17)	1.38	0.17
Phoenician	0.24 (\pm 0.20)	1.16	0.25
Pracalit	0.02 (\pm 0.18)	0.10	0.92
Psalter Pahlavi	0.38 (\pm 0.20)	1.88	0.06
Runic	-0.02 (\pm 0.17)	-0.11	0.92
Shavian	0.007 (\pm 0.18)	0.04	0.97
Sora Sompeng	0.07 (\pm 0.17)	0.40	0.69
Soyombo	-0.04 (\pm 0.17)	-0.26	0.80
Syloti Nagri	-0.03 (\pm 0.18)	-0.14	0.89
Tagbanwa	0.58 (\pm 0.20)	2.86	0.005
Tai Viet	0.11 (\pm 0.19)	0.58	0.56
Warang Citi	0.26 (\pm 0.20)	1.34	0.18
Zanabazar Square	0.19 (\pm 0.17)	1.14	0.26

Table 8: Full logistic regression table for Gemini-1.5, language effects

Fixed effects	$\hat{\beta}$	z	p
Avestan	-0.08 (\pm 0.19)	-0.43	0.67
Bamum	0.11 (\pm 0.20)	0.54	0.59
Bassa Vah	0.23 (\pm 0.18)	1.26	0.21
Batak	0.25 (\pm 0.19)	1.34	0.18
Buginese	0.12 (\pm 0.24)	0.51	0.61
Cherokee	0.21 (\pm 0.18)	1.17	0.24
Gothic	0.23 (\pm 0.22)	1.04	0.30
Grantha	-0.40 (\pm 0.17)	-2.40	0.017
Hangul	0.20 (\pm 0.18)	1.12	0.27
Kayah Li	-0.05 (\pm 0.18)	-0.28	0.78
Khudawadi	0.17 (\pm 0.22)	0.77	0.44
Lao	-0.06 (\pm 0.17)	-0.37	0.71
Lepcha	0.35 (\pm 0.18)	1.91	0.057
Linear B	0.09 (\pm 0.18)	0.52	0.60
Mandaic	-0.04 (\pm 0.18)	-0.20	0.84
Meetei Mayek	0.02 (\pm 0.17)	0.11	0.91
Meroitic Cursive	-0.52 (\pm 0.24)	-2.18	0.03
Mro	0.22 (\pm 0.19)	1.16	0.25
Multani	0.11 (\pm 0.17)	0.65	0.52
Nabataean	-0.08 (\pm 0.19)	-0.41	0.68
New Tai Lue	0.30 (\pm 0.19)	1.55	0.12
Nyiakeng Puachue Hmong	-0.05 (\pm 0.19)	-0.24	0.81
Ogham	0.10 (\pm 0.20)	0.49	0.63
Ol Chiki	-0.05 (\pm 0.19)	-0.27	0.79
Old Permic	-0.03 (\pm 0.17)	-0.18	0.86
Osmanya	0.28 (\pm 0.27)	1.02	0.31
Pau Cin Hau	0.06 (\pm 0.17)	0.38	0.71
Phoenician	0.07 (\pm 0.20)	0.36	0.72
Pracalit	-0.20 (\pm 0.18)	-1.12	0.27
Psalter Pahlavi	0.20 (\pm 0.20)	0.98	0.33
Runic	0.08 (\pm 0.17)	0.48	0.63
Shavian	0.07 (\pm 0.18)	0.40	0.69
Sora Sompeng	-0.09 (\pm 0.17)	-0.52	0.61
Soyombo	-0.24 (\pm 0.17)	-1.43	0.15
Syloti Nagri	-0.19 (\pm 0.18)	-1.05	0.30
Tagbanwa	0.21 (\pm 0.20)	1.03	0.31
Tai Viet	0.07 (\pm 0.19)	0.39	0.70
Warang Citi	0.19 (\pm 0.19)	0.96	0.34
Zanabazar Square	0.03 (\pm 0.17)	0.19	0.85

Table 9: Full logistic regression table for GPT-4o, language effects

Fixed effects	$\hat{\beta}$	z	p
(Intercept)	0.25 (\pm 0.13)	1.96	0.05
color	-0.038 (\pm 0.031)	-1.20	0.23
leftright	-0.057 (\pm 0.031)	-1.80	0.072
length	0.0019 (\pm 0.00068)	2.85	0.005
cross	-0.11 (\pm 0.13)	-0.84	0.40
diagonal	0.12 (\pm 0.13)	0.92	0.36
enclosed	0.051 (\pm 0.050)	1.03	0.30
horizontal	-0.097 (\pm 0.089)	-1.09	0.28
mirror	0.23 (\pm 0.23)	1.00	0.32
specific	0.48 (\pm 0.053)	9.02	< 0.001
stroke	-0.097 (\pm 0.068)	-1.43	0.15
symmetry	0.13 (\pm 0.068)	1.97	0.05
vertical	0.28 (\pm 0.090)	3.15	0.002

Table 10: Full logistic regression table for Idefics2, main effects

Fixed effects	$\hat{\beta}$	z	p
(Intercept)	0.21 (\pm 0.13)	1.67	0.10
color	0.08 (\pm 0.03)	2.55	0.01
leftright	0.04 (\pm 0.03)	1.13	0.26
length	0.002 (\pm <0.001)	3.09	0.002
cross	-0.11 (\pm 0.13)	-0.86	0.39
diagonal	0.06 (\pm 0.13)	0.46	0.64
enclosed	0.04 (\pm 0.05)	0.75	0.45
horizontal	-0.10 (\pm 0.09)	-1.07	0.28
mirror	-0.03 (\pm 0.23)	-0.14	0.89
specific	0.34 (\pm 0.05)	6.24	<0.001
stroke	-0.06 (\pm 0.07)	-0.87	0.39
symmetry	-0.15 (\pm 0.07)	-2.10	0.04
vertical	0.12 (\pm 0.09)	1.27	0.20

Table 11: Full logistic regression table for Idefics3, main effects

Fixed effects	$\hat{\beta}$	z	p
Avestan	-0.13 (\pm 0.16)	-0.85	0.40
Bamum	-0.12 (\pm 0.17)	-0.73	0.47
Bassa Vah	-0.094 (\pm 0.15)	-0.61	0.54
Batak	-0.31 (\pm 0.16)	-1.95	0.052
Buginese	0.027 (\pm 0.20)	0.14	0.89
Cherokee	-0.13 (\pm 0.15)	-0.87	0.39
Gothic	-0.0029 (\pm 0.18)	-0.016	0.99
Grantha	-0.23 (\pm 0.14)	-1.64	0.10
Hangul	-0.0026 (\pm 0.16)	-0.017	0.99
Kayah Li	-0.15 (\pm 0.15)	-0.98	0.33
Khudawadi	-0.020 (\pm 0.18)	-0.11	0.91
Lao	0.11 (\pm 0.14)	0.75	0.45
Lepcha	-0.17 (\pm 0.15)	-1.08	0.28
Linear B	-0.18 (\pm 0.15)	-1.19	0.24
Mandaic	-0.088 (\pm 0.15)	-0.57	0.57
Meetei Mayek	0.071 (\pm 0.14)	0.50	0.62
Meroitic Cursive	0.052 (\pm 0.20)	0.26	0.80
Mro	-0.10 (\pm 0.16)	-0.62	0.54
Multani	-0.26 (\pm 0.15)	-1.78	0.08
Nabataean	-0.069 (\pm 0.16)	-0.44	0.66
New Tai Lue	0.14 (\pm 0.16)	0.86	0.39
Nyiakeng Puachue Hmong	-0.11 (\pm 0.16)	-0.69	0.49
Ogham	-0.16 (\pm 0.17)	-0.95	0.35
Ol Chiki	-0.060 (\pm 0.16)	-0.38	0.71
Old Permic	-0.13 (\pm 0.14)	-0.87	0.38
Osmanya	-0.10 (\pm 0.23)	-0.45	0.65
Pau Cin Hau	-0.17 (\pm 0.14)	-1.23	0.22
Phoenician	-0.33 (\pm 0.17)	-1.94	0.054
Pracalit	-0.11 (\pm 0.15)	-0.73	0.47
Psalter Pahlavi	-0.13 (\pm 0.17)	-0.76	0.45
Runic	-0.17 (\pm 0.14)	-1.23	0.22
Shavian	0.076 (\pm 0.15)	0.50	0.62
Sora Sompeng	-0.10 (\pm 0.14)	-0.69	0.49
Soyombo	-0.18 (\pm 0.14)	-1.26	0.21
Syloti Nagri	-0.014 (\pm 0.15)	-0.093	0.93
Tagbanwa	-0.097 (\pm 0.17)	-0.57	0.57
Tai Viet	-0.037 (\pm 0.16)	-0.24	0.81
Warang Citi	0.046 (\pm 0.16)	0.28	0.78
Zanabazar Square	0.039 (\pm 0.14)	0.28	0.78

Table 12: Full logistic regression table for Idefics2, language effects

Fixed effects	$\hat{\beta}$	z	p
Avestan	-0.11 (\pm 0.16)	-0.66	0.51
Bamum	-0.24 (\pm 0.17)	-1.35	0.18
Bassa Vah	-0.05 (\pm 0.16)	-0.31	0.76
Batak	-0.12 (\pm 0.16)	-0.75	0.46
Buginese	0.31 (\pm 0.20)	1.54	0.13
Cherokee	0.22 (\pm 0.15)	1.45	0.15
Gothic	-0.07 (\pm 0.19)	-0.37	0.71
Grantha	-0.19 (\pm 0.14)	-1.37	0.17
Hangul	-0.04 (\pm 0.16)	-0.23	0.82
Kayah Li	-0.02 (\pm 0.15)	-0.15	0.88
Khudawadi	-0.03 (\pm 0.19)	-0.16	0.88
Lao	0.05 (\pm 0.15)	0.36	0.72
Lepcha	-0.10 (\pm 0.16)	-0.66	0.51
Linear B	-0.22 (\pm 0.15)	-1.48	0.14
Mandaic	-0.12 (\pm 0.16)	-0.78	0.44
Meetei Mayek	-0.03 (\pm 0.14)	-0.20	0.84
Meroitic Cursive	-0.11 (\pm 0.21)	-0.52	0.60
Mro	-0.23 (\pm 0.17)	-1.35	0.18
Multani	-0.07 (\pm 0.15)	-0.44	0.66
Nabataean	-0.11 (\pm 0.16)	-0.67	0.51
New Tai Lue	0.15 (\pm 0.17)	0.87	0.38
Nyiakeng Puachue Hmong	0.1016 (\pm 0.16)	0.63	0.53
Ogham	0.05 (\pm 0.17)	0.27	0.79
Ol Chiki	-0.12 (\pm 0.16)	-0.77	0.45
Old Permic	-0.14 (\pm 0.15)	-0.93	0.35
Osmanya	-0.32 (\pm 0.23)	-1.35	0.18
Pau Cin Hau	0.02 (\pm 0.14)	0.14	0.89
Phoenician	-0.26 (\pm 0.17)	-1.47	0.14
Pracalit	-0.01 (\pm 0.15)	-0.08	0.93
Psalter Pahlavi	-0.14 (\pm 0.17)	-0.83	0.41
Runic	0.05 (\pm 0.14)	0.38	0.70
Shavian	-0.10 (\pm 0.16)	-0.67	0.50
Sora Sompeng	-0.10 (\pm 0.15)	-0.67	0.51
Soyombo	-0.28 (\pm 0.14)	-1.97	0.05
Syloti Nagri	-0.17 (\pm 0.15)	-1.11	0.27
Tagbanwa	-0.15 (\pm 0.17)	-0.86	0.39
Tai Viet	0.11 (\pm 0.16)	0.66	0.51
Warang Citi	0.25 (\pm 0.17)	1.51	0.13
Zanabazar Square	-0.04 (\pm 0.14)	-0.31	0.76

Table 13: Full logistic regression table for Idefics3, language effects

Fixed effects	$\hat{\beta}$	z	p
(Intercept)	0.33 (\pm 0.14)	2.5	0.02
color	0.050 (\pm 0.032)	1.6	0.11
leftright	0.022 (\pm 0.032)	0.70	0.49
length	0.0014 (\pm <0.001)	1.9	0.05
cross	-0.18 (\pm 0.14)	-1.3	0.18
diagonal	0.055 (\pm 0.14)	0.40	0.69
enclosed	0.26 (\pm 0.053)	4.8	<0.001
horizontal	-0.031 (\pm 0.095)	-0.32	0.75
mirror	0.52 (\pm 0.25)	2.1	0.04
specific	0.37 (\pm 0.057)	6.5	<0.001
stroke	0.026 (\pm 0.073)	0.36	0.72
symmetry	0.0022 (\pm 0.073)	0.030	0.98
vertical	0.043 (\pm 0.097)	0.44	0.66

Table 14: Full logistic regression table for LLaVA-NeXT, main effects.

Fixed effects	$\hat{\beta}$	z	p
(Intercept)	0.17 (\pm 0.12)	1.48	0.14
color	0.01 (\pm 0.03)	0.30	0.77
leftright	0.03 (\pm 0.03)	1.08	0.28
length	0.002 (\pm <0.001)	3.88	0.0001
cross	-0.24 (\pm 0.12)	-2.04	0.04
diagonal	-0.06 (\pm 0.12)	-0.54	0.59
enclosed	0.02 (\pm 0.05)	0.39	0.70
horizontal	-0.09 (\pm 0.08)	-1.15	0.25
mirror	-0.06 (\pm 0.22)	-0.26	0.80
specific	0.17 (\pm 0.05)	3.34	0.001
stroke	-0.15 (\pm 0.06)	-2.37	0.02
symmetry	-0.04 (\pm 0.06)	-0.64	0.52
vertical	0.06 (\pm 0.08)	0.71	0.48

Table 15: Full logistic regression table for InstructBLIP, main effects.

Fixed effects	$\hat{\beta}$	z	p
Avestan	-0.29 (\pm 0.17)	-1.7	0.09
Bamum	-0.098 (\pm 0.18)	-0.54	0.59
Bassa Vah	-0.32 (\pm 0.17)	-1.9	0.05
Batak	-0.19 (\pm 0.17)	-1.1	0.25
Buginese	0.023 (\pm 0.21)	0.11	0.91
Cherokee	-0.16 (\pm 0.16)	-0.96	0.34
Gothic	-0.39 (\pm 0.20)	-2.0	0.05
Grantha	-0.28 (\pm 0.15)	-1.9	0.06
Hangul	-0.27 (\pm 0.17)	-1.6	0.11
Kayah Li	-0.24 (\pm 0.16)	-1.5	0.14
Khudawadi	-0.24 (\pm 0.20)	-1.2	0.21
Lao	-0.066 (\pm 0.15)	-0.43	0.67
Lepcha	-0.067 (\pm 0.16)	-0.41	0.68
Linear B	-0.29 (\pm 0.16)	-1.8	0.07
Mandaic	-0.36 (\pm 0.17)	-2.2	0.03
Meetei Mayek	-0.068 (\pm 0.15)	-0.45	0.65
Meroitic Cursive	-0.25 (\pm 0.22)	-1.1	0.25
Mro	-0.32 (\pm 0.18)	-1.8	0.07
Multani	-0.27 (\pm 0.16)	-1.7	0.09
Nabataean	-0.15 (\pm 0.17)	-0.86	0.39
New Tai Lue	-0.053 (\pm 0.18)	-0.30	0.76
Nyiakeng Puachue Hmong	-0.049 (\pm 0.17)	-0.29	0.77
Ogham	-0.075 (\pm 0.18)	-0.41	0.68
Ol Chiki	-0.13 (\pm 0.17)	-0.76	0.45
Old Permic	-0.068 (\pm 0.15)	-0.44	0.66
Osmanya	-0.23 (\pm 0.25)	-0.92	0.36
Pau Cin Hau	-0.19 (\pm 0.15)	-1.2	0.22
Phoenician	-0.44 (\pm 0.18)	-2.4	0.02
Pracalit	-0.17 (\pm 0.16)	-1.1	0.29
Psalter Pahlavi	-0.069 (\pm 0.18)	-0.37	0.71
Runic	-0.25 (\pm 0.15)	-1.7	0.10
Shavian	-0.13 (\pm 0.16)	-0.79	0.43
Sora Sompeng	-0.35 (\pm 0.16)	-2.3	0.03
Soyombo	-0.49 (\pm 0.15)	-3.2	0.001
Syloti Nagri	-0.31 (\pm 0.16)	-1.9	0.05
Tagbanwa	-0.14 (\pm 0.18)	-0.79	0.43
Tai Viet	-0.13 (\pm 0.17)	-0.76	0.45
Warang Citi	-0.070 (\pm 0.18)	-0.40	0.69
Zanabazar Square	-0.0030 (\pm 0.15)	-0.020	0.98

Table 16: Full logistic regression table for LLaVA-NeXT, language effects

Fixed effects	$\hat{\beta}$	z	p
Avestan	0.04 (\pm 0.15)	0.27	0.78
Bamum	0.05 (\pm 0.16)	0.29	0.77
Bassa Vah	-0.14 (\pm 0.14)	-1.01	0.32
Batak	-0.07 (\pm 0.15)	-0.49	0.63
Buginese	0.36 (\pm 0.19)	1.95	0.05
Cherokee	-0.24 (\pm 0.14)	-1.68	0.09
Gothic	-0.10 (\pm 0.17)	-0.58	0.56
Grantha	-0.16 (\pm 0.13)	-1.23	0.22
Hangul	-0.15 (\pm 0.15)	-1.01	0.31
Kayah Li	-0.15 (\pm 0.14)	-1.11	0.27
Khudawadi	-0.01 (\pm 0.17)	-0.08	0.94
Lao	-0.22 (\pm 0.13)	-1.61	0.11
Lepcha	-0.02 (\pm 0.14)	-0.13	0.90
Linear B	0.0003 (\pm 0.14)	0.003	0.99
Mandaic	-0.07 (\pm 0.14)	-0.45	0.65
Meetei Mayek	0.01 (\pm 0.13)	0.04	0.97
Meroitic Cursive	0.10 (\pm 0.19)	0.54	0.59
Mro	-0.16 (\pm 0.15)	-1.06	0.29
Multani	0.01 (\pm 0.14)	0.06	0.95
Nabataean	-0.02 (\pm 0.15)	-0.12	0.90
New Tai Lue	-0.03 (\pm 0.15)	-0.21	0.84
Nyiakeng Puachue Hmong	-0.21 (\pm 0.15)	-1.40	0.16
Ogham	-0.12 (\pm 0.16)	-0.76	0.45
Ol Chiki	-0.06 (\pm 0.15)	-0.43	0.67
Old Permic	-0.12 (\pm 0.13)	-0.89	0.38
Osmanya	-0.26 (\pm 0.22)	-1.20	0.23
Pau Cin Hau	-0.06 (\pm 0.13)	-0.48	0.63
Phoenician	-0.14 (\pm 0.16)	-0.86	0.39
Pracalit	0.01 (\pm 0.14)	0.06	0.95
Psalter Pahlavi	0.01 (\pm 0.16)	0.09	0.93
Runic	-0.03 (\pm 0.13)	-0.24	0.81
Shavian	-0.02 (\pm 0.14)	-0.17	0.86
Sora Sompeng	-0.14 (\pm 0.14)	-1.06	0.29
Soyombo	-0.12 (\pm 0.13)	-0.90	0.37
Syloti Nagri	-0.10 (\pm 0.14)	-0.68	0.50
Tagbanwa	-0.14 (\pm 0.16)	-0.90	0.37
Tai Viet	-0.10 (\pm 0.15)	-0.68	0.50
Warang Citi	-0.01 (\pm 0.15)	-0.01	0.99
Zanabazar Square	-0.08 (\pm 0.13)	-0.64	0.52

Table 17: Full logistic regression table for InstructBLIP, language effects.

Fixed effects	$\hat{\beta}$	z	p
(Intercept)	0.21 (\pm 0.15)	1.47	0.14
color	0.04 (\pm 0.03)	1.33	0.18
leftright	0.04 (\pm 0.03)	1.23	0.22
length	0.004 (\pm 0.001)	4.77	<0.001
cross	-0.14 (\pm 0.15)	-0.96	0.34
diagonal	-0.02 (\pm 0.15)	-0.14	0.89
enclosed	0.15 (\pm 0.06)	2.62	0.009
horizontal	-0.19 (\pm 0.10)	-1.89	0.060
mirror	0.45 (\pm 0.27)	1.70	0.089
specific	0.49 (\pm 0.06)	7.96	<0.001
stroke	-0.13 (\pm 0.08)	-1.59	0.11
symmetry	0.01 (\pm 0.08)	0.18	0.85
vertical	0.14 (\pm 0.10)	1.37	0.17

Table 18: Full logistic regression table for Molmo-O, main effects.

Fixed effects	$\hat{\beta}$	z	p
(Intercept)	0.070 (\pm 0.14)	0.49	0.63
color	0.05 (\pm 0.03)	1.85	0.065
leftright	0.07 (\pm 0.03)	2.50	0.013
length	0.005 (\pm <0.001)	5.84	<0.001
cross	-0.20 (\pm 0.14)	-1.36	0.18
diagonal	0.35 (\pm 0.15)	2.37	0.02
enclosed	0.003 (\pm 0.06)	0.06	0.95
horizontal	-0.19 (\pm 0.10)	-1.90	0.059
mirror	0.62 (\pm 0.27)	2.35	0.02
specific	0.45 (\pm 0.06)	7.29	<0.001
stroke	-0.18 (\pm 0.08)	-2.24	0.03
symmetry	-0.29 (\pm 0.08)	-3.64	<0.001
vertical	0.23 (\pm 0.10)	2.18	0.03

Table 19: Full logistic regression table for Molmo-D, main effects.

Fixed effects	$\hat{\beta}$	z	p
Avestan	-0.25 (\pm 0.18)	-1.36	0.18
Bamum	-0.13 (\pm 0.20)	-0.67	0.50
Bassa Vah	-0.05 (\pm 0.18)	-0.26	0.79
Batak	0.01 (\pm 0.18)	0.05	0.96
Buginese	0.14 (\pm 0.23)	0.61	0.54
Cherokee	-0.05 (\pm 0.17)	-0.31	0.76
Gothic	-0.18 (\pm 0.21)	-0.84	0.40
Grantha	-0.32 (\pm 0.16)	-1.97	0.05
Hangul	0.06 (\pm 0.18)	0.32	0.75
Kayah Li	-0.21 (\pm 0.17)	-1.19	0.24
Khudawadi	-0.32 (\pm 0.21)	-1.51	0.13
Lao	-0.10 (\pm 0.17)	-0.61	0.54
Lepcha	0.08 (\pm 0.18)	0.44	0.66
Linear B	-0.10 (\pm 0.17)	-0.57	0.57
Mandaic	-0.14 (\pm 0.18)	-0.77	0.44
Meetei Mayek	-0.09 (\pm 0.16)	-0.57	0.57
Meroitic Cursive	-0.09 (\pm 0.23)	-0.37	0.71
Mro	-0.32 (\pm 0.19)	-1.71	0.09
Multani	0.01 (\pm 0.17)	0.08	0.94
Nabataean	-0.05 (\pm 0.18)	-0.30	0.76
New Tai Lue	0.20 (\pm 0.19)	1.04	0.30
Nyiakeng Puachue Hmong	-0.05 (\pm 0.18)	-0.28	0.78
Ogham	0.05 (\pm 0.20)	0.25	0.80
Ol Chiki	0.11 (\pm 0.18)	0.58	0.56
Old Permic	-0.28 (\pm 0.17)	-1.66	0.10
Osmanya	0.15 (\pm 0.27)	0.58	0.56
Pau Cin Hau	-0.25 (\pm 0.16)	-1.55	0.12
Phoenician	-0.15 (\pm 0.20)	-0.77	0.44
Pracalit	-0.18 (\pm 0.17)	-1.05	0.29
Psalter Pahlavi	-0.04 (\pm 0.20)	-0.22	0.83
Runic	-0.12 (\pm 0.16)	-0.71	0.48
Shavian	-0.12 (\pm 0.18)	-0.70	0.48
Sora Sompeng	-0.16 (\pm 0.17)	-0.95	0.34
Soyombo	-0.02 (\pm 0.16)	-0.10	0.92
Syloti Nagri	0.04 (\pm 0.17)	0.22	0.82
Tagbanwa	-0.02 (\pm 0.20)	-0.10	0.92
Tai Viet	-0.05 (\pm 0.18)	-0.28	0.78
Warang Citi	-0.10 (\pm 0.19)	-0.52	0.61
Zanabazar Square	-0.003 (\pm 0.16)	-0.02	0.99

Table 20: Full logistic regression table for Molmo-O, language effects.

Fixed effects	$\hat{\beta}$	z	p
Avestan	0.02 (\pm 0.18)	0.08	0.93
Bamum	0.05 (\pm 0.20)	0.24	0.81
Bassa Vah	0.06 (\pm 0.18)	0.34	0.73
Batak	0.28 (\pm 0.18)	1.56	0.12
Buginese	0.68 (\pm 0.23)	2.94	0.004
Cherokee	0.14 (\pm 0.17)	0.80	0.42
Gothic	0.18 (\pm 0.21)	0.84	0.40
Grantha	-0.10 (\pm 0.16)	-0.59	0.55
Hangul	0.22 (\pm 0.18)	1.25	0.21
Kayah Li	0.08 (\pm 0.17)	0.47	0.64
Khudawadi	0.07 (\pm 0.21)	0.33	0.74
Lao	0.05 (\pm 0.17)	0.28	0.78
Lepcha	0.04 (\pm 0.18)	0.21	0.83
Linear B	0.17 (\pm 0.17)	0.98	0.33
Mandaic	0.03 (\pm 0.18)	0.17	0.87
Meetei Mayek	0.003 (\pm 0.16)	0.02	0.99
Meroitic Cursive	0.13 (\pm 0.23)	0.55	0.58
Mro	-0.10 (\pm 0.19)	-0.51	0.61
Multani	0.08 (\pm 0.17)	0.50	0.62
Nabataean	-0.04 (\pm 0.18)	-0.22	0.83
New Tai Lue	0.18 (\pm 0.19)	0.95	0.34
Nyiakeng Puachue Hmong	0.10 (\pm 0.18)	0.56	0.57
Ogham	0.08 (\pm 0.20)	0.43	0.67
Ol Chiki	0.17 (\pm 0.18)	0.91	0.36
Old Permic	0.16 (\pm 0.17)	0.99	0.32
Osmanya	-0.08 (\pm 0.27)	-0.29	0.78
Pau Cin Hau	0.01 (\pm 0.16)	0.08	0.94
Phoenician	-0.03 (\pm 0.20)	-0.13	0.90
Pracalit	0.005 (\pm 0.17)	0.03	0.98
Psalter Pahlavi	0.29 (\pm 0.20)	1.47	0.14
Runic	0.03 (\pm 0.16)	0.20	0.84
Shavian	-0.03 (\pm 0.18)	-0.15	0.88
Sora Sompeng	-0.07 (\pm 0.17)	-0.39	0.69
Soyombo	-0.06 (\pm 0.16)	-0.35	0.73
Syloti Nagri	-0.006 (\pm 0.17)	-0.03	0.97
Tagbanwa	0.44 (\pm 0.20)	2.25	0.026
Tai Viet	0.02 (\pm 0.18)	0.08	0.93
Warang Citi	0.24 (\pm 0.19)	1.28	0.20
Zanabazar Square	0.21 (\pm 0.16)	1.31	0.19

Table 21: Full logistic regression table for Molmo-D, language effects.