COLUMBUS: Evaluating COgnitive Lateral Understanding Through Multiple-Choice reBUSes

Koen Kraaijveld[1], **Yifan Jiang**[2], **Kaixin Ma**[3], **Filip Ilievski**[1]
[1]Department of Computer Science, Faculty of Science, Vrije Universiteit Amsterdam
[2]Information Sciences Institute, University of Southern California
[3]Tencent AI Lab, Bellevue, WA
h.j.kraaijveld@student.vu.nl, yjiang44@usc.edu, kaixinma@global.tencent.com, f.ilievski@vu.nl

Abstract

While visual question-answering (VQA) benchmarks have catalyzed the development of reasoning techniques, they have focused on vertical thinking. Effective problem-solving also necessitates lateral thinking, which remains understudied in AI and has not been used to test visual perception systems. To bridge this gap, we formulate *visual lateral thinking* as a multiple-choice question-answering task and describe a three-step taxonomy-driven methodology for instantiating task examples. Then, we develop COLUMBUS, a synthetic benchmark that applies the task pipeline to create QA sets with text and icon rebus puzzles based on publicly available collections of compounds and common phrases. COLUMBUS comprises over 1,000 puzzles, each with four answer candidates. While the SotA vision-language models (VLMs) achieve decent performance, our evaluation demonstrates a substantial gap between humans and models. VLMs benefit from human-curated descriptions but struggle to self-generate such representations at the right level of abstraction.

Code — https://github.com/koen-47/COLUMBUS

1 Introduction

Human problem-solving seamlessly combines vertical and lateral thinking (De Bono 2016). *Vertical* thinking is an analytical search process that rewards logic, rules, and rationality. It optimizes correctness by narrowing down on quality solutions and rejecting suboptimal ones (Hernandez and Varkey 2008). For example, resolving the question mark in Figure 1 (left) requires systematically identifying that all examples adhere to the formula: (*left* - *top*) $\times$ *right* + *bottom* = 77. Meanwhile, *lateral* thinking (De Bono 1971) is explorative, divergent, and creative (Hernandez and Varkey 2008). It expands the solution space by diverging into novel directions. As illustrated in the right part of Figure 1, visual, spatial, verbal, and numerical cues must be interpreted unconventionally (defying common sense; Jiang, Ilievski, and Ma 2024), a process that lends itself to lateral thinking. In this example of a *rebus* puzzle, the numbers "1111" phonetically represent the word "ONCE", while the visual-spatial relationship between the blue letters "MO" and "ON" spell "BLUE MOON". As "ONCE" is placed inside "BLUE MOON" this leads to the solution *B) Once in a blue moon*.

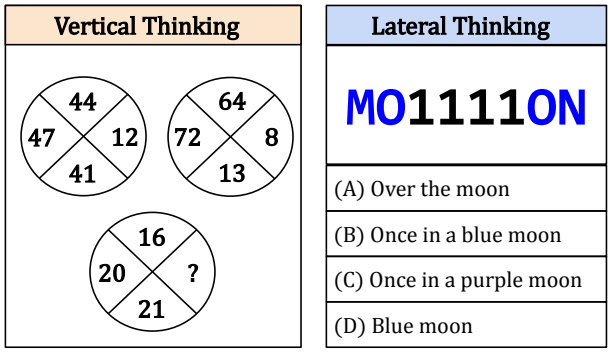

Figure 1: Left: vertical thinking puzzle from Machine Number Sense (Zhang et al. 2020). Right: lateral thinking rebus puzzle from our COLUMBUS benchmark.

Existing benchmarks for visual question answering (VQA) (Agrawal et al. 2016) have been instrumental in exploring and enhancing the vertical thinking skills of vision-language models (VLMs). Popular subtasks are visual reasoning (Johnson et al. 2017; Li and Søgaard 2022; Li et al. 2023b), abstract visual reasoning (AVR) (Chollet 2019; Malinowski and Fritz 2015; Małkiński and Mańdziuk 2023; Zhang et al. 2019), and visual commonsense reasoning (VCR) (Bitton-Guetta et al. 2023), all requiring both processing of visual as well as linguistic information. Meanwhile, lateral thinking benchmarks have recently been proposed as word and sentence puzzles but are limited only to the textual modality (Jiang et al. 2023b; Huang et al. 2024). Hence, there is a lack of lateral thinking benchmarks for multimodal settings combining text and vision.

To this end, we study *how well VLMs exhibit multimodal lateral thinking*. Our contributions are as follows:

1. **A taxonomy-driven three-step methodology** for creating lateral thinking tasks in a multiple-choice VQA format. Our taxonomy definition yields 18 rules that manipulate the visual attributes and relationships of the puzzle's elements (text or icons). The puzzle rendering step leverages this taxonomy to create a graph representation for a puzzle answer and generate an image for the graph. The distractor sampling step is based on a weighted average of orthographic and semantic similarity between a puzzle's correct answer and its visible elements.

2. **A synthetic benchmark called COLUMBUS** that applies the lateral thinking methodology to create QA sets with rebus puzzles based on public collections of phrases (e.g., idioms) and compound words.[1] COLUMBUS comprises over 1,000 puzzles consisting of textual and icon elements, each with four answer candidates.
3. **An experimental analysis** with COLUMBUS with human participants and representative state-of-the-art (SotA) vision-language models evaluated in a zero-shot setting, revealing that models perform decently but lag behind humans. Moreover, models benefit from human-curated descriptions, but even the SotA ones struggle to generate representations at the right level of abstraction.

2 Related Work

Rebus Puzzles. In psychology, rebus puzzles have been known to demand lateral thinking (Salvi et al. 2016; Threadgold, Marsh, and Ball 2018; MacGregor and Cunningham 2008). Prior work (Salvi et al. 2016; Threadgold, Marsh, and Ball 2018) reports human accuracies of 74.5% and 53.31%, respectively, and compares the impact of vertical and lateral thinking, concluding that using lateral thinking led to a significant improvement in the number of puzzles solved. To our knowledge, the only existing benchmark of rebus puzzles that assesses VLMs is REBUS (Gritsevskiy et al. 2024). This benchmark contains 333 human-annotated puzzles separated into 13 categories with three difficulty levels. Half of the models tested in this work achieve less than 5% accuracy. The authors ascribe this difficulty to the benchmark's reliance on world knowledge (e.g., cities, towns, public transport stations) and vertical thinking skills like string manipulation. Instead, we devise a methodology for automatic and scalable generation of rebus puzzles based on publicly available resources. The sources to create COLUMBUS (phrases and compounds) are deliberately selected to focus on lateral thinking only and minimize the need for world knowledge and vertical thinking.

Vertical Thinking in VQA. AVR puzzles, illustrated in Figure 1 (left), are commonly used to assess multimodal reasoning. Discriminative tasks such as Raven's Progressive Matrices (Raven 1941; Barrett et al. 2018; Zhang et al. 2019) and Visual Analogy Problems (Webb et al. 2020) involve completing sequences of panels with abstract shapes selected from a predefined set of options. Bongard problems (Bongard 1968; Nie et al. 2020) require discovering the rules that separate and govern shapes across two sets of panels, though these rules must be described in natural language. MARVEL (Jiang et al. 2024) encompasses these benchmarks with a more comprehensive set of patterns, input shapes, and configurations, along with rigorous checks to assess that model answers are grounded in perception and reasoning. Alternatively, generative approaches, like the Abstraction and Reasoning Corpus (Chollet 2019), test the ability to recreate missing panels without choosing from predefined options. A comprehensive review of AVR puzzles is provided by Małkiński and Mańdziuk (2023). Rather than using puzzles, CLEVR (Johnson et al. 2017), QLEVR (Li and Søgaard 2022), and Super-CLEVR (Li et al. 2023b) are synthetic benchmarks that test logical reasoning by analyzing 3D rendered scenes of objects. WHOOPS! (Bitton-Guetta et al. 2023) is a visual commonsense reasoning benchmark of images generated through diffusion models that consist of illogical scenarios (e.g., Albert Einstein holding a smartphone). Crucially, these benchmarks rely on vertical thinking and do not test out-of-the-box thinking. Thus, our lateral task methodology and the COLUMBUS benchmark enable a complementary assessment of the models' abilities.

Text-based Lateral Evaluation. Recent work has recognized an analogous omission of lateral thinking for the text domain. Jiang et al. (2023b) introduce BRAINTEASER, a multiple-choice lateral thinking benchmark with 1,100 puzzles adapted from online sources. BRAINTEASER requires models to bypass commonsense associations to arrive at the correct answer. Similarly, Huang et al. (2024) present LatEval, a benchmark consisting of 300 lateral puzzles. Each puzzle in LatEval is an interactive game between two large language models (LLMs) in which the LLM under evaluation must solve the puzzle presented by the host LLM. While we share the goal of testing models' lateral thinking ability, we broaden the evaluation scope to a multimodal setting covering text and vision.

3 Methodology for Visual Lateral Tasks

To ensure a straightforward automatic evaluation and minimize answer ambiguity, we frame each puzzle as a multiple-choice VQA pair. A puzzle $p = (I, (q, O, c))$ consists of a rebus image $I \subseteq \mathcal{I}$ and question $q \in \mathcal{S}$ with correct answer $c \in \mathcal{S}$ chosen from options $O = \{o_1, o_2, o_3, c\}$; $O \subseteq \mathcal{S}$. $\mathcal{I}$ and $\mathcal{S}$ denote the space of images and strings, respectively. Each I can be decomposed into a set of elements $E \subseteq \mathcal{I} \cup \mathcal{S}$, where $\forall e \in E \ (e \in \mathcal{I} \oplus e \in \mathcal{S})$. The latent rules that govern the appearance and visual-spatial relationships of each $e \in E$ are determined by $R : \mathcal{S} \rightarrow \mathcal{I}$. The goal in solving p is to select a response $r \in O$ such that $R(r) = R(c)$.

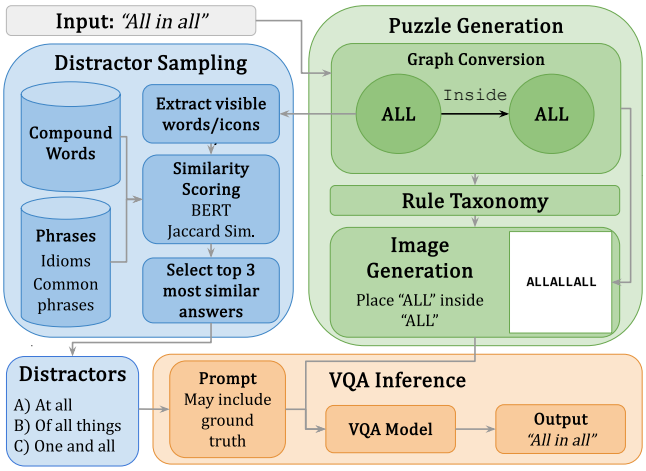

Figure 2: Methodology for visual lateral thinking tasks.

[1]The name refers to the demonstration of lateral thinking in the story of *Columbus' Egg* (Benzoni 2017).

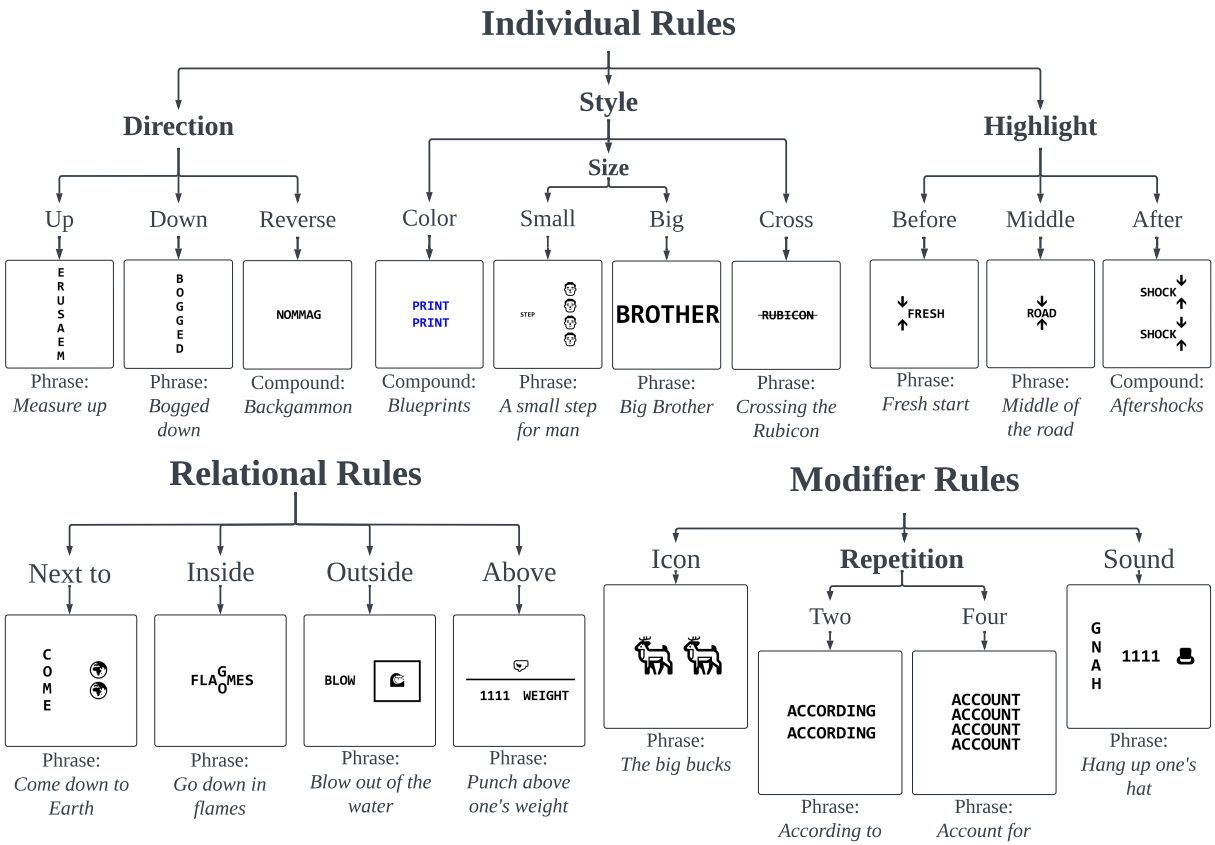

Figure 3: Three taxonomies that classify and organize the *individual* (top), *relational* (bottom left), and *modifier* (bottom right) rules used to manipulate the appearance and position of elements in a rebus puzzle. For each rule, we present an example puzzle and its answer, both taken directly from COLUMBUS.

Figure 2 depicts our method. As rebus puzzles are typically built around idiomatic expressions, compound words, or common phrases (e.g., "according to", "a bit too much") (Salvi et al. 2016; Threadgold, Marsh, and Ball 2018), we assume phrases and compounds as inputs for our method. We start by designing a taxonomy of latent rules. Using this taxonomy, each compound or phrase is converted into an attributed, directed graph, which is subsequently converted into the puzzle image. Finally, distractors are sampled by identifying other compounds or phrases that overlap with, or are semantically similar to, the method input.

3.1 Taxonomy of Latent Rules

We derive a novel taxonomy of latent rules to support the development of lateral thinking puzzles. The taxonomy consolidates online guides and databases of rebus puzzles and a rebus categorization scheme outlined by Salvi et al. (2016). We manually select and organize the categories in these sources such that each rule uniquely manipulates an element through visual, spatial, verbal, and numerical properties. We ensure that each rule can be automatically operationalized and mixed with others in the same puzzle.

The resulting taxonomy (Figure 3) consists of 18 rules, grouped into three categories according to how they manipulate elements in a puzzle: 1. **Individual** rules define the unary characteristics of an element in a rebus. Example rules include reversing character order (direction), the text color (style), and adding arrows before the element (highlight). 2. **Relational** rules define the positioning between a pair of elements. We define four *relational* rules, placing an element beside/inside/above/outside another. 3. **Modifier** rules are designed to be mutually inclusive with other *individual* rules. Examples include repeating an element multiple times or substituting it with a phonetically similar element.

3.2 Puzzle Rendering

Rebus puzzles include elements (i.e., text or icons) whose appearance and position are determined by latent rules triggered by specific keywords in the puzzle's answer. This is illustrated in Figure 1 (right), where the words "ONCE", "IN", "BLUE", and "MOON" determine the puzzle's elements and their arrangement. We expect that SotA generative models cannot be reliably applied to generate rebus puzzles, a hypothesis we validate in Section 6.5. Instead, we render a puzzle by a taxonomy-driven transformation of its input elements, which first produces a graph and subsequently an image (green part in Figure 2).

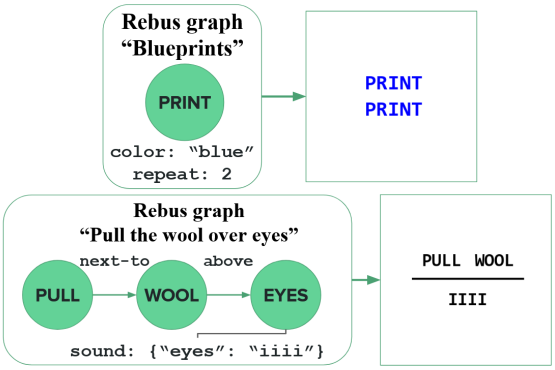

Figure 4: Two examples of directed attributed graphs (left) representing rebus puzzles (right).

Graph Generation Algorithm. We generate a directed, attributed graph whose nodes are elements that will be rendered into a puzzle image. The node attributes specify the rendering of that element, i.e., the *individual* or *modifier* rules that will apply to it. The edges between two nodes are annotated with an attribute that specifies their *relational* rule. We parse a puzzle answer into a graph by following a separate procedure for compounds and phrases. Figure 4 shows the rebus graphs for two puzzle images based on a compound and a phrase, respectively.

For *compounds* (Figure 4 top), we create a graph with a single node using the following steps. First, we split a compound into its constituent words, e.g., "blueprints" consists of "blue" and "prints". For each word, we check if it matches against any of the keywords of an *individual* rule, e.g., "blue" triggers the *color* rule. Then, we create a graph with a single node consisting of the other constituent word ("prints") and set this node's color attribute to blue. Since we detect that "prints" is plural, we set its *repetition* modifier attribute to 2. For the final step, we check if the word in the node corresponds to any available homophones or icons, which does not occur in this case. In cases where both constituent words of a compound trigger a rule, we generate both graph interpretations of the input.

For *phrases* (Figure 4 bottom), we first identify keywords belonging to a *relational* rule, e.g., "over" triggers the *above* rule. At this word, we split the phrase into two substrings: "pull the wool over eyes" yields "pull the wool" and "eyes". On each substring, we run the compound parser over each pair of words from left to right, e.g., the first and only pair of the first substring is ("pull", "wool"). This process results in a set of nodes, which are then connected using the *next to* rule, yielding two path subgraphs. The final step involves connecting these two subgraphs with an edge with the *relational* rule identified in the first step (*above* in our example).

Image Generation. We select one of four templates using only the graph as input. Templates include x, y coordinates and a font-size multiplier. Three templates position up to three points equally along the x-axis in the image center, mapping graph nodes (left to right) to template points. The fourth handles graphs with the *above* rule. We change the appearance of the elements in an image according to the attributes of that element's respective node.

3.3 Distractor Sampling

Distractor sampling (blue part in Figure 2) selects the three most similar compounds or phrases to the input semi-automatically. We opt for sampling rather than data augmentation approaches like rephrasing because compounds and proverbial phrases are challenging to generate automatically. To select a distractor for a puzzle, we obtain its visible elements (text and icons) from the graph representation and compute similarity to all other phrases/compounds. The similarity uses a λ-weighted combination of Jaccard word overlap (Leskovec, Rajaraman, and Ullman 2014) and cosine similarity using Sentence-BERT embeddings (Reimers and Gurevych 2019). We expect that distractors with word overlaps make the task more challenging because the test-taker works with the visible words. Since word overlap may fail to select relevant distractors when the visible words only occur once or too many times across the entire dataset of phrases and compounds, we also leverage semantic cosine similarity to include distractors that contain synonyms of the words in the original input.

4 The COLUMBUS Benchmark

We apply our proposed pipeline to instantiate the first visual lateral thinking benchmark, COLUMBUS.

Puzzle Answer Collection. We start by scraping common English phrases from publicly available sources, namely Wiktionary and www.theidioms.com, yielding 9,745 instances. We use the Large Database of English Compounds (LaDEC) (Gagn, Spalding, and Schmidtke 2019) for compound words. This dataset has been feature-engineered and curated by humans, consisting of 8,957 compounds. We fill rules that appear less than ten times across the benchmark by semi-automatically adding compounds and phrases that trigger them, with the assistance of prompting the OpenAI's ChatGPT-3.5 model (Brown et al. 2020). All combined, we collect 18,836 candidate answers from which to generate puzzles. Additionally, we collected homophones (25 samples) and icons (480 samples). Homophones were added manually by recognizing common ones found in rebus puzzle databases. The icon collection combines icons scraped from an online source and manually added ones.[2]

Quality Control. The graph parsing for all phrases and compounds includes a preprocessing step to remove stopwords that do not belong to the set of rule-triggering keywords.[3] Multiple elements with *individual* rules can still be present in the same puzzle, and more than one *modifier* rule can be applied to an element. However, we apply at most one *individual* rule to a single element. In cases where multiple *individual* rules can be applied to a single element, we generate these individually for each rule as separate puzzles. To further improve readability and limit the risk of overlapping elements, we restrict the image's rendered elements using heuristics based on the number of elements and their rules.

[2]Source: https://unicode.org/Public/emoji/11.0/emoji-test.txt

[3]E.g., "to" is a stopword, but triggers the *repetition: two* rule phonetically.

Category	Statistic	TEXT	ICON	All
General	Number of puzzles	634	371	1,005
	Mean answer length (# words)	4.12	5.35	4.58
Rules	Freq. of *individual* rules	253	76	329
	Freq. of *relational* rules	540	425	965
	Freq. of *modifier* rules	503	255	758
Graphs	Freq. of single node graphs	197	35	232
	Freq. of double node graphs	332	246	578
	Freq. of triple node graphs	105	90	195
Distractors	% of questions with distractors containing visible puzzle words	89.27	97.57	92.34

Table 1: Key statistics of the COLUMBUS benchmark.

These exclude images generated on graphs that (1) have more than three nodes connected by the *next to* rule, (2) have an *above* rule where the top/bottom exceeds two nodes, (3) have an *above* rule where either the top/bottom exceeds two nodes, (4) have an *outside* rule where either the inside/outside portion has more than one node. We take the top 1,000 from the remaining puzzles with the most edges and rules per node to ensure the benchmark is challenging. To provide a fairer comparison between textual and icon puzzles, each puzzle containing an icon is duplicated, and all its icons are converted to their textual counterparts. Finally, we filter out low-quality puzzles with overlapping or overflowing text from this remaining set.

Benchmark Composition. We split the benchmark into two partitions: COLUMBUS-TEXT that only contain text and COLUMBUS-ICON that contain at least one icon. Between these two partitions, COLUMBUS features an *overlap* subset of 338 puzzle pairs. Each pair consists of two versions of the same puzzle: one version uses icons, and the other uses text instead of those icons. Table 1 shows key statistics about COLUMBUS. While non-icon puzzles are more numerous, icon puzzles feature more elements. This can be attributed to the difference in the answer length, as longer answers feature more chances that a word can be replaced with an icon.

5 Experimental Design

Model Families. We include open- and closed-source instruction-tuned and non-instruction-tuned VLMs. We also test closed-source models enriched with forward and backward chaining. We evaluate all models in a zero-shot setting using standard hyperparameter values.

For non-instruction-tuned models, we test 1) *BLIP-2* (Li et al. 2023a) with the OPT-2.7b and the OPT-6.7b LLMs (Zhang et al. 2022); 2) *Fuyu-8b* (Bavishi et al. 2023), a multimodal text and image transformer that achieves competitive performance on VQA tasks. We also evaluate *CLIP* (Radford et al. 2021), a seminal VLM and a foundation for many other models used in our experiments. As CLIP is not a VQA model, we switch its task to image classification, which must match the image of a puzzle to the correct answer from the four available choices.

As instruction-tuned models, we include 1) BLIP-2 coupled with *Flan-T5-11b* (Chung et al. 2022), which achieves SotA performance on zero-shot VQA tasks; 2) *InstructBLIP* (Dai et al. 2023), an instruction tuned version of BLIP-2 model that uses Vicuna-7b (Zheng et al. 2023); 3) *QwenVL* (Bai et al. 2023b), a 7 billion parameter visual multimodal version of the Qwen LLM (Bai et al. 2023a) from which we use the chat variant; 4) *CogVLM* (Wang et al. 2023), a 17 billion parameter VLM that achieves SotA performance on several VQA benchmarks; 5) *Llava* (Liu et al. 2023a), a large VLM that achieves SotA performance on several vision benchmarks despite its lack of billion-scale data. For Llava, we use the 13b (v1.5) and 34b (v1.6) variants; 6) *Mistral-7b* (v2) (Jiang et al. 2023a) to use in text-only, question-answering (QA) auxiliary experiments.

For closed-source models, we select four models from two representative families based on their promising performance in public visual reasoning benchmarks (Lu et al. 2023; Liu et al. 2023b): 1) GPT-4o and GPT-4o-mini (OpenAI 2024) and 2) Gemini 1.5 (Pro) and Gemini 1.5 (Flash) (DeepMind 2023).

We experiment with two *structural* variants of closed-source models: forward and backward chaining (Jurafsky and Martin 2009). In *forward chaining (FC)*, the model constructs evidence from an image and connects this evidence with the optimal candidate answer (Wang et al. 2024). The forward chaining approach first prompts models to generate JSON files with attributes (e.g., name, relation, description) for each visible puzzle element, which can later be used as a reference in the final prompting. As a representative of *backward chaining (BC)* from question and image towards the answer, we test *belief graphs* (Kassner et al. 2023) where a model derives and evaluates explanations for each candidate answer. Belief graphs excavate additional information by recursively evaluating the truth assignments of premises generated for each answer candidate. The truth assignments are then optimized with an SAT solver, yielding the most probable answer. This approach is evaluated on a random subset of 50 puzzles.

Human Evaluation. To estimate human performance on COLUMBUS, we ask five participants to answer a subset of 103 randomly selected puzzles, consisting of 37 text, 40 icon puzzles, and 13 *overlap* puzzles with both a textual and icon variant.

Model Inputs. We explore four human-curated input levels, each providing the model with increasing information about the puzzle, i.e., its description and details on the nodes or edges of a puzzle's graph. Specifically: 1. no description of the nature of the puzzle, nor the graph; 2. only a description of the nature of the puzzle; 3. description of the nature of the puzzle and the graph nodes; 4. description of the nature of the puzzle and the full graph (both nodes and edges).

Evaluation Protocol. Following other multiple-choice benchmarks (Jiang et al. 2023b; Zhu et al. 2016; de Faria et al. 2023), we use accuracy as the evaluation metric, defined as the percentage of puzzles solved correctly. To extract answers from a model's output, we use regex to check for choice symbols (e.g., "A") if they are present and then perform exact string matching to the correct answer/symbol.

Model	TEXT		ICON	
	Mean	SD	Mean	SD
CLIP	56.15	0.00	52.56	0.00
BLIP-2 OPT (2.7b)	24.74	0.21	24.08	0.13
BLIP-2 OPT (6.7b)	23.95	0.16	25.61	0.00
Fuyu (8b)	32.02	0.00	31.00	0.00
InstructBLIP Vicuna (7b)	51.47	0.13	51.75	0.38
Qwen-VL (7b)	58.02	0.35	63.16	0.55
BLIP-2 Flan-T5-XXL (11b)	68.24	0.07	71.97	0.00
Llava (13b)	58.02	0.09	58.76	0.00
CogVLM (17b)	59.28	0.09	60.11	0.00
Llava (34b)	66.82	0.73	73.13	1.41
GPT-4o	80.89	0.97	**83.34**	1.11
GPT-4o-mini	73.96	0.71	77.69	0.49
Gemini 1.5 (Pro)	71.56	3.71	77.52	**5.08**
Gemini 1.5 (Flash)	64.42	2.00	67.44	2.93
GPT-4o (FC)	**81.28**	1.15	79.20	0.78
GPT-4o-mini (FC)	73.53	0.79	74.36	1.29
Gemini 1.5 (Pro) (FC)	69.98	3.00	72.10	3.17
Gemini 1.5 (Flash) (FC)	72.00	1.42	75.88	3.55
GPT-4o (BC)	64.37	1.63	71.67	4.71
GPT-4o-mini (BC)	45.93	**8.65**	60.00	4.08
Human	98.00	N/A	93.21	N/A

Table 2: Results for each model on COLUMBUS-TEXT and COLUMBUS-ICON. The accuracy's mean and standard deviation (SD) are reported across three runs. The highest and second highest model results are highlighted in **bold** and underlined, respectively. The prompt includes a description of the nature of the puzzle (i.e., prompt 2). We did not test backward chaining for the Gemini models, as they do not output probabilities.

For the larger, more flexible models that produce long explanations for their answers, we use GPT-4o to extract their answers automatically. For model outputs that answer a given puzzle with multiple options, we pick one of them randomly. All models are run three times, and their performance is averaged to account for randomness.

6 Results

We investigate five questions: 1) How well can VLMs solve rebus puzzles that require lateral thinking? 2) Can forward and backward chaining enhance lateral thinking performance? 3) Do VLMs benefit from prompts that supply more information about the puzzle? 4) How does the performance of VLMs vary across different puzzle rules? 5) Can VLMs generate task puzzles directly?

6.1 Overall Performance

Table 2 shows the performance of each model on COLUMBUS-TEXT and COLUMBUS-ICON. The closed-source and larger open-source models perform best on both partitions, while the small, non-instruction-tuned models perform near-randomly. Comparing the mean model performance with text and icons, we see no significant difference. Namely, the accuracy is slightly higher on COLUMBUS-ICON, whereas on the overlapping set of 338 puzzles, we observe a slightly higher accuracy on the textual puzzles. As expected, the best model for each partition is consistently GPT-4o. Yet, none of the models surpass human accuracy, with average gaps of 38.17% on COLUMBUS-TEXT and 30.64% on COLUMBUS-ICON.

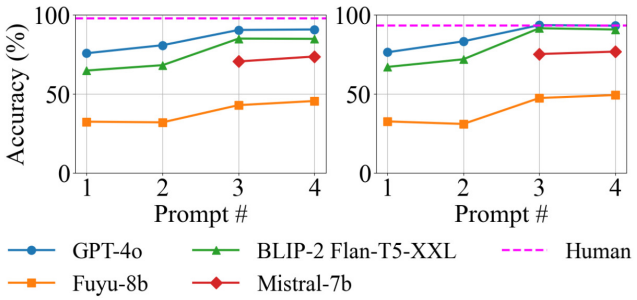

Figure 5: Results for four prompts that supply the model with increasing information for COLUMBUS-TEXT (left) and COLUMBUS-ICON (right) (averaged across three runs). The best-performing model from the following types is shown: open-source non-instruction VLM (Fuyu-8b), open-source instruction VLM (BLIP-2 Flan-T5-XXL), closed-source VLM (GPT-4o), and text-only LLM (Mistral-7b), as well as human accuracy.

6.2 Impact of Structural Reasoning

The two structured variants show opposing results (Table 2). Forward chaining, leading the model to generate graph descriptions in JSON format, yields little effect on the performance of GPT-4o (-1.88%) and Gemini (+2.26%), averaged across both models and partitions. Both models suffer from a gap against human performance. On the contrary, backward chaining yields a 14.1% and 22.86% drop in accuracy for GPT-4o and GPT-4o-mini averaged across the two partitions. We ascribe this to the models lacking a global overview of the image, as each evaluated premise focuses on local parts of a puzzle without cohesively pointing towards a candidate answer.

6.3 Model Sensitivity to Input Information

Can models benefit from a ground-truth structured description of the puzzle provided in their input? Figure 5 shows that adding information about the nature of the puzzle (prompt 2) has little effect (+2.68% and +3.39% for textual and icon puzzles, respectively). Adding a description of the graph nodes (prompt 3) increases the model performance by 11.91% and 14.9% for non-icon and icon puzzles, respectively, reaching over 90% for GPT-4o. However, adding information on the *relational* rules only increases performance 1.47% and 0.56% for COLUMBUS-TEXT and -ICON, respectively. Considering the example in Figure 1, the models extract the text as is (e.g., extract "MO1111ON") and cannot make the lateral connection that words/letters need rearranging. Even GPT-4o struggles consistently with this, such as with certain *direction* rules, as discussed in the next Section.

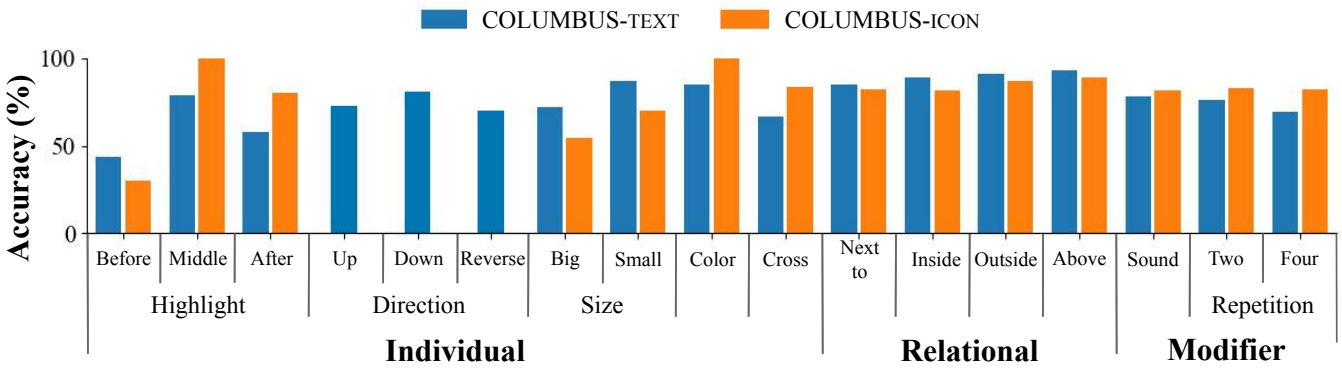

Figure 6: Percentage of puzzles solved by GPT-4o for a single run in COLUMBUS-TEXT and COLUMBUS-ICON for each rule. The prompt used only describes the nature of the puzzle (prompt 2). For COLUMBUS-ICON, the *direction* rules are omitted because these either do not function with icons or become functionally identical to other rules when combined with icons.

6.4 Rule-based Analysis

Figure 6 shows results for how often a puzzle containing a specific rule is solved correctly by the best-performing model (GPT-4o). On COLUMBUS-TEXT, GPT-4o performs the best on the *relational* rules and the worst on *individual* rules (difference of 17.98%). When *individual* rules appear together with *modifier* rules, the model performance is slightly higher (by 3.02%). We see a similar trend for COLUMBUS-ICON, with a gap from *individual* to *relational* and *modifier* rules being 10.96% and 8.21%, respectively. We note that, while the GPT-4o's performance is similar on the two partitions, specific rules are more difficult for this model when represented as text (e.g., *repetition* rules). In contrast, others are more challenging when presented as icons (e.g., *size*). Such biases align with recent work that shows the perceptual sensitivity of VLMs to object visual attributes (Zhang et al. 2024). As for *relational* rules, the model performance on text and icon puzzles is on par.

6.5 VLM Generation of Puzzles

Given the strong generative abilities of VLMs, a natural question arises: can they generate puzzles without the methodology we define in Section 3? To investigate whether the taxonomy-based generation pipeline is necessary, we sample 100 puzzle answers and use DALLE-3 (Betker et al. 2023) to generate corresponding puzzles with the prompt "Try to generate an image for a rebus puzzle on {answer}". Three human annotators are asked first to label whether the puzzles contain sufficient visible elements to support solving the puzzle and then select the better one between the two puzzles (the one generated by our method and the one by DALLE-3). Our results show that 98% of the rebuses generated by our pipeline contain a sound and complete list of elements, compared to only 44% for DALLE-3. Additionally, the puzzles from our pipeline were preferred over those from DALLE-3 84% of the time, with an 11% tie rate. DALLE-3 struggles as a rebus generator for two main reasons Figure 7 (Betker et al. 2023): 1) Noisy details: unlike the concise rebuses our pipeline produces, DALLE-3 often creates very complex puzzles that include irrelevant information (e.g., the click icon in the right of Figure 7). 2) Abstract representation: DALLE-3 struggles to represent abstract ideas, such as "after", whereas our carefully designed rule-based taxonomy can handle these concepts precisely.

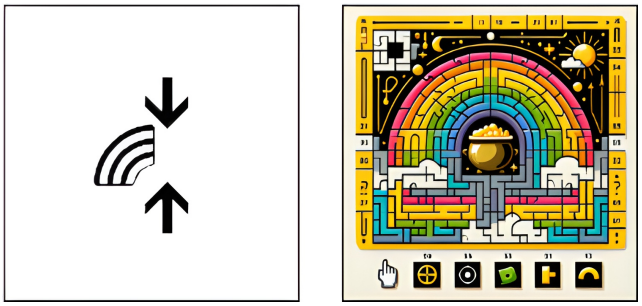

Figure 7: Rebus images for *end of the rainbow*, generated by our automated pipeline (left) and by DALLE-3 (right).

7 Conclusions

This paper introduces COLUMBUS, a synthetic multiple-choice benchmark comprising 1005 rebus puzzles designed to assess visual lateral thinking. Experiments revealed a substantial gap between human and vision language model performance, which narrowed when models received graph descriptions, suggesting they primarily rely on puzzle elements rather than the spatial relationships between them. The models particularly struggled with text-rearrangement rules requiring flexible, puzzle-specific abstractions. Meanwhile, the scale, difficulty, and balance across puzzle types of COLUMBUS is still limited. Future work should extend its methodology to create more comprehensive versions by incorporating diverse multimodal formats, refining graph structures to control difficulty, and varying perceptual dimensions like color, positioning, and size. Special emphasis should be placed on puzzles demanding abstraction and creative thinking, with additional mechanisms such as synonyms and related concepts to counteract reliance on direct word matching.

References

Agrawal, A.; Lu, J.; Antol, S.; Mitchell, M.; Zitnick, C. L.; Batra, D.; and Parikh, D. 2016. VQA: Visual Question Answering. arXiv:1505.00468.

Bai, J.; et al. 2023a. Qwen Technical Report. *arXiv preprint arXiv:2309.16609*.

Bai, J.; et al. 2023b. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966*.

Barrett, D. G. T.; Hill, F.; Santoro, A.; Morcos, A. S.; and Lillicrap, T. 2018. Measuring abstract reasoning in neural networks. arXiv:1807.04225.

Bavishi, R.; Elsen, E.; Hawthorne, C.; Nye, M.; Odena, A.; Somani, A.; and Tarlar, S. 2023. Fuyu-8B: A Multimodal Architecture for AI Agents. https://www.adept.ai/blog/fuyu-8b/. Accessed: 2024-04-15.

Benzoni, G. 2017. *The History of the New World: Benzoni's Historia del Mondo Nuovo*. Penn State University Press.

Betker, J.; Goh, G.; Jing, L.; Brooks, T.; Wang, J.; Li, L.; Ouyang, L.; Zhuang, J.; Lee, J.; Guo, Y.; et al. 2023. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3): 8.

Bitton-Guetta, N.; Bitton, Y.; Hessel, J.; Schmidt, L.; Elovici, Y.; Stanovsky, G.; and Schwartz, R. 2023. Breaking Common Sense: WHOOPS! A Vision-and-Language Benchmark of Synthetic and Compositional Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2616–2627.

Bongard, M. M. 1968. *The Recognition Problem*. Foreign Technology Div Wright-Patterson AFB Ohio, Tech. Rep.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Chollet, F. 2019. On the Measure of Intelligence. arXiv:1911.01547.

Chung, H. W.; et al. 2022. Scaling Instruction-Finetuned Language Models. _eprint: 2210.11416.

Dai, W.; Li, J.; LI, D.; Tiong, A.; Zhao, J.; Wang, W.; Li, B.; Fung, P. N.; and Hoi, S. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In Oh, A.; Neumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 49250–49267. Curran Associates, Inc.

De Bono, E. 1971. *The Use of Lateral Thinking*. Intl Center for Creative Thinking. ISBN 978-0-14-013788-0.

De Bono, E. 2016. *Lateral thinking: a textbook of creativity*. Penguin Life. ISBN 978-0-241-25754-8.

de Faria, A. C. A. M.; de Castro Bastos, F.; da Silva, J. V. N. A.; Fabris, V. L.; de Sousa Uchoa, V.; de Aguiar Neto, D. G.; and dos Santos, C. F. G. 2023. Visual Question Answering: A Survey on Techniques and Common Trends in Recent Literature. arXiv:2305.11033.

DeepMind. 2023. Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.11805.

Gagn, C. L.; Spalding, T. L.; and Schmidtke, D. 2019. LADEC: The Large Database of English Compounds. 51(5): 2152–2179.

Gritsevskiy, A.; Panickssery, A.; Kirtland, A.; Kauffman, D.; Gundlach, H.; Gritsevskaya, I.; Cavanagh, J.; Chiang, J.; Roux, L. L.; and Hung, M. 2024. REBUS: A Robust Evaluation Benchmark of Understanding Symbols. arXiv:2401.05604.

Hernandez, J.; and Varkey, P. 2008. Vertical versus lateral thinking. 34: 26–8.

Huang, S.; Ma, S.; Li, Y.; Huang, M.; Zou, W.; Zhang, W.; and Zheng, H. 2024. LatEval: An Interactive LLMs Evaluation Benchmark with Incomplete Information from Lateral Thinking Puzzles. In Calzolari, N.; Kan, M.-Y.; Hoste, V.; Lenci, A.; Sakti, S.; and Xue, N., eds., *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 10186–10197. Torino, Italia: ELRA and ICCL.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023a. Mistral 7B. arXiv:2310.06825.

Jiang, Y.; Ilievski, F.; and Ma, K. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. *arXiv preprint arXiv:2404.16068*.

Jiang, Y.; Ilievski, F.; Ma, K.; and Sourati, Z. 2023b. BRAINTEASER: Lateral Thinking Puzzles for Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 14317–14332. Singapore: Association for Computational Linguistics.

Jiang, Y.; Zhang, J.; Sun, K.; Sourati, Z.; Ahrabian, K.; Ma, K.; Ilievski, F.; and Pujara, J. 2024. MARVEL: Multidimensional Abstraction and Reasoning through Visual Evaluation and Learning. arXiv:2404.13591.

Johnson, J.; Hariharan, B.; van der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; and Girshick, R. 2017. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jurafsky, D.; and Martin, J. H. 2009. *Speech and language processing*. Prentice Hall series in artificial intelligence. London [u.a.]: Prentice Hall, Pearson Education International, 2. ed., [pearson international edition] edition. ISBN 0-13-504196-1, 978-0-13-504196-3.

Kassner, N.; Tafjord, O.; Sabharwal, A.; Richardson, K.; Schütze, H.; and Clark, P. 2023. Language Models with Rationality. In *EMNLP*.

Leskovec, J.; Rajaraman, A.; and Ullman, J. D. 2014. *Finding Similar Items*, 68122. Cambridge University Press.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org. Place: , Honolulu, Hawaii, USA,.

Li, Z.; and Søgaard, A. 2022. QLEVR: A Diagnostic Dataset for Quantificational Language and Elementary Visual Reasoning. In Carpuat, M.; de Marneffe, M.-C.; and Meza Ruiz, I. V., eds., *Findings of the Association for Computational Linguistics: NAACL 2022*, 980–996. Seattle, United States: Association for Computational Linguistics.

Li, Z.; Wang, X.; Stengel-Eskin, E.; Kortylewski, A.; Ma, W.; Van Durme, B.; and Yuille, A. L. 2023b. Super-CLEVR: A Virtual Benchmark to Diagnose Domain Robustness in Visual Reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14963–14973.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023a. Visual Instruction Tuning. In Oh, A.; Neumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 34892–34916. Curran Associates, Inc.

Liu, Y.; Duan, H.; Zhang, Y.; Li, B.; Zhang, S.; Zhao, W.; Yuan, Y.; Wang, J.; He, C.; Liu, Z.; et al. 2023b. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

Lu, P.; Bansal, H.; Xia, T.; Liu, J.; Li, C.; Hajishirzi, H.; Cheng, H.; Chang, K.-W.; Galley, M.; and Gao, J. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

MacGregor, J. N.; and Cunningham, J. B. 2008. Rebus puzzles as insight problems. 40(1): 263–268.

Malinowski, M.; and Fritz, M. 2015. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. arXiv:1410.0210.

Małkiński, M.; and Mańdziuk, J. 2023. A review of emerging research directions in Abstract Visual Reasoning. *Information Fusion*, 91: 713–736.

Nie, W.; Yu, Z.; Mao, L.; Patel, A. B.; Zhu, Y.; and Anandkumar, A. 2020. BONGARD-LOGO: a new benchmark for human-level concept learning and reasoning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Raven, J. C. 1941. STANDARDIZATION OF PROGRESSIVE MATRICES, 1938. *British Journal of Medical Psychology*, 19(1): 137–150.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Salvi, C.; Costantini, G.; Bricolo, E.; Perugini, M.; and Beeman, M. 2016. Validation of Italian rebus puzzles and compound remote associate problems. 48(2): 664–685.

Threadgold, E.; Marsh, J. E.; and Ball, L. J. 2018. Normative Data for 84 UK English Rebus Puzzles. 9: 2513. Place: Switzerland.

Wang, W.; Fang, T.; Li, C.; Shi, H.; Ding, W.; Xu, B.; Wang, Z.; Bai, J.; Liu, X.; Cheng, J.; et al. 2024. CANDLE: iterative conceptualization and instantiation distillation from large language models for commonsense reasoning. *arXiv preprint arXiv:2401.07286*.

Wang, W.; Lv, Q.; Yu, W.; Hong, W.; Qi, J.; Wang, Y.; Ji, J.; Yang, Z.; Zhao, L.; Song, X.; Xu, J.; Xu, B.; Li, J.; Dong, Y.; Ding, M.; and Tang, J. 2023. CogVLM: Visual Expert for Pretrained Language Models. arXiv:2311.03079.

Webb, T. W.; Dulberg, Z.; Frankland, S. M.; Petrov, A. A.; O'Reilly, R. C.; and Cohen, J. D. 2020. Learning representations that support extrapolation. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Zhang, C.; Gao, F.; Jia, B.; Zhu, Y.; and Zhu, S.-C. 2019. RAVEN: A Dataset for Relational and Analogical Visual rEasoNing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, J.; Hu, J.; Khayatkhoei, M.; Ilievski, F.; and Sun, M. 2024. Exploring perceptual limitation of multimodal large language models. *arXiv preprint arXiv:2402.07384*.

Zhang, S.; et al. 2022. OPT: Open Pre-trained Transformer Language Models. arXiv:2205.01068.

Zhang, W.; Zhang, C.; Zhu, Y.; and Zhu, S. 2020. Machine Number Sense: A Dataset of Visual Arithmetic Problems for Abstract and Relational Reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1332–1340.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv:2306.05685.

Zhu, Y.; Groth, O.; Bernstein, M.; and Fei-Fei, L. 2016. Visual7W: Grounded Question Answering in Images. In *IEEE Conference on Computer Vision and Pattern Recognition*.