

EUREKA: Evaluating and Understanding Large Foundation Models

Vidhisha Balachandran Jingya Chen Neel Joshi Besmira Nushi
Hamid Palangi Eduardo Salinas Vibhav Vineet
James Woffinden-Luey Safoora Yousefi
Microsoft Research

🔗 Code: <https://github.com/microsoft/eureka-ml-insights>

Abstract

Rigorous and reproducible evaluation of large foundation models is critical for assessing the state of the art, informing next steps in model improvement, and for guiding scientific advances in Artificial Intelligence (AI). Evaluation is also important for informing the increasing number of application developers that build services on foundation models. The evaluation process has however become challenging in practice due to several reasons that require immediate attention from the community, including benchmark saturation, lack of transparency in the methods being deployed for measurement, development challenges in extracting the right measurements for generative tasks, and, more generally, the extensive number of capabilities that need to be considered for showing a well-rounded comparison across models. In addition, despite the overwhelming numbers of side-by-side capability evaluations available, we still lack a deeper understanding about when and how different models fail for a given capability and whether the nature of failures is similar across different models being released over time.

We make three contributions to alleviate the above challenges. First, we present EUREKA, a reusable and open evaluation framework for standardizing evaluations of large foundation models beyond single-score reporting and rankings. Second, we introduce EUREKA-BENCH as an extensible collection of benchmarks testing capabilities that (i) are still challenging for state-of-the-art foundation models and (ii) represent fundamental but overlooked capabilities for completing tasks in both language and vision modalities. The available space for improvement that comes inherently from non-saturated benchmarks, enables us to discover meaningful differences between models at a capability level. Third, using the framework and EUREKA-BENCH, we conduct an analysis of 12 state-of-the-art models, providing in-depth insights for failure understanding and model comparison by disaggregating the measurements across important subcategories of data. Such insights uncover granular weaknesses of models for a given capability and can then be further leveraged to plan more precisely on what areas are most promising for improvement. EUREKA is available as open-source to foster transparent and reproducible evaluation practices.

In contrast to recent trends in evaluation reports and leaderboards showing absolute rankings and claims for one model or another to be the best, our analysis shows that there is no such best model. Different models have different strengths, but there are models that appear more often than others as best performers for several capabilities. Despite the many observed improvements, it also becomes obvious that current models still struggle with a number of fundamental capabilities including detailed image understanding, benefiting from multimodal input when available rather than fully relying on language, factuality and grounding for information retrieval, and over refusals.

[†]Correspondence to {benushi, neel, sayouse, vidhishab, vivineet}@microsoft.com.

[‡]Currently at Google. Work done while at Microsoft Research.

Contents

1	Introduction	3
2	Results Summary	5
3	Evaluation Framework	6
4	Multimodal Evaluation	9
4.1	Geometric Reasoning - GeoMeter	9
4.2	Multimodal Question Answering - MMMU	12
4.3	Image Understanding	16
4.4	Vision Language Understanding	19
4.5	High Level vs. Detailed Image Understanding - Discussion	22
5	Language Evaluation	22
5.1	Instruction Following - IFEval	23
5.2	Long Context - FlenQA	25
5.3	Information Retrieval - Kitab	27
5.4	Toxicity Detection and Safe Language Generation - Toxigen	32
6	Non-Determinism Evaluation	37
6.1	Experiment Setup and Metrics	37
6.2	Results	37
7	Backward Compatibility Evaluation	40
7.1	Datasets and Models	40
7.2	Claude 3.5 Sonnet vs. Claude 3 Opus	42
7.3	GPT-4o 2024-05-13 vs. GPT-4 1106 Preview/GPT-4 Turbo 2024-04-09	42
7.4	Llama 3.1 70B vs. Llama 3 70B	43
7.5	Main Takeaways	44
8	Related Work and Limitations	45
8.1	Capability Evaluations	45
8.2	Evaluation methodologies and frameworks	46
9	Conclusion	47

1 Introduction

The evaluation of Large Foundation Models (LFMs) presents several methodical and practical challenges, many of which stem from the generative and general-purpose nature of recent models. The rapid progress in AI has also introduced many new capabilities as part of the model skills portfolio, which need to be assessed alongside traditional capabilities. EUREKA is a framework and a collection of challenging benchmarks that aims at scaling up such evaluations for LFMs in an open and transparent manner. The framework itself provides a library for flexibly customizing evaluation pipelines that combine a series of components necessary for evaluation including data preprocessing, prompt templates, model inference, data postprocessing, metric computation, and reporting. EUREKA-BENCH is the collection of benchmarks whose implementation is currently supported in EUREKA and for which we provide extensive evaluation and analysis reports.

Modality	Benchmark #prompts	Capability	Experimental Conditions	Subcategories
Image → Text	GeoMeter 1086	Geometric Reasoning		Depth Height
Image → Text	MMMUM 900	Multimodal QA		Disciplines Subjects
Image → Text	Image Understanding 10,240	Object Recognition Object Detection Visual Prompting Spatial Reasoning	Single Object Two Objects	Object Recognition Object Detection Visual Prompting Spatial Reasoning
Image → Text	Vision Language Understanding 13,500	Spatial Understanding Navigation Counting	Image Only Text Only Image and Text	Spatial Map Maze Navigation Object Counting
Text → Text	IFEval 541	Instruction Following		Instruction Category
Text → Text	FlenQA 12,000	Long Context multi-hop QA	Context Length Info Placement	Monotonic Relations People in Rooms Ruletaker
Text → Text	Kitab 34,217	Information Retrieval	Context Availability Constraint Count	Constraint type Author popularity Query constrainedness
Text → Text	Toxigen 10,500	Toxicity Detection Safe Language Generation	Discriminative Generative	Demographic groups

Table 1: Benchmarks currently available in EUREKA-BENCH.

Evaluation Framework. The evaluation process for complex and generative capabilities has made traditional practices for evaluation obsolete. For example, the concept of a fixed, closed-form definition of a metric does not apply anymore to many capabilities either because several different sub metrics need to be computed before reaching a final score, or because many data transformations and answer extraction operations need to be applied to the output prior to computing a metric. Some of these data transformations often are custom to the model being evaluated. In addition, part of the evaluation also needs to be handed over to other model judges for scaling up [19, 131]. This new landscape leaves practitioners with the necessity of creating a rich combination of data processing steps, code execution, and model inferences as evaluators, all in function of producing a final score for the model under test. EUREKA provides a flexible library for composing these functionalities into shareable evaluation pipelines and gives full control to practitioners to handle and log the details of each experiment. These functionalities also enable reproducibility and backtracking of experimentation details (e.g. prompt templates, inference parameters, API and model versions) in a transparent manner. Given that such details can change measurements significantly, we believe it is important for the research community to have access to both the code and logs behind evaluations. Thus, we provide the actual code and logs used in evaluations.

Benchmark selection. A pressing issue in the evaluation of state-of-the-art LFMs is the fact that many of the benchmarks commonly reported in technical reports of model releases are either close to *saturation* or already saturated, where models have reached close to 100% accuracy on the benchmarks. For example, several recent models [80, 35, 1, 89, 3] have been reported to have an accuracy higher than 85% on benchmarks like MMLU [45], GSM8K [27], HumanEval [23], DROP [34], BigBench-Hard [101], MGSM [97], ChartQA [70], AI2D [51]. Many of the benchmarks in this list represent tasks that were important for testing fundamental capabilities at the time of their release. However, saturation of performance does not leave ample space for discovering major failure modes and for comparing different models. While saturation itself may originate either from inherent model improvements or from memorization, the challenge from a scientific communication

Modality	Claude	Gemini	GPT	Llama	Llava	Mistral
Multimodal Image → Text	Claude 3 Opus Claude 3.5 Sonnet	Gemini 1.5 Pro	GPT-4 Vision Preview GPT-4 Turbo 2024-04-09 GPT-4o 2024-05-13		Llava 1.6 34B	
Language Text → Text	Claude 3 Opus Claude 3.5 Sonnet	Gemini 1.5 Pro	GPT-4 1106 Preview GPT-4o 2024-05-13	Llama 3 70B Llama 3.1 70B Llama 3.1 405B		Mistral Large 2407

Table 2: Models being evaluated via EUREKA-BENCH for language and multimodal capabilities.

perspective is reflected as lack of clarity in quantifying and characterizing improvements over time. In fact, many of the recent improvements in such benchmarks can be as small as 1-2 percentage points, which leaves one wondering whether we are indeed experiencing a true more general saturation in terms of progress in AI or whether the benchmarks and methods being used for measurement are insufficient.

To create space for deeper analysis, the benchmarks in EUREKA-BENCH (Table 1) are chosen such that either the whole benchmark, or an important experimental condition within that benchmark, remains challenging for even the most capable models. As a simplified rule of thumb, we choose to include in EUREKA-BENCH benchmarks for which overall model performance (or performance on an important experimental condition) is less than 80% for either all models or at least roughly half of the models studied in this work.

Another consideration for benchmark selection is capability and modality coverage. While EUREKA-BENCH is not an exhaustive list of capabilities, it aims at covering diverse fundamental language and multimodal capabilities that are currently overlooked in traditional evaluations but that are critical for more complex tasks. For example, spatial and geometric understanding are not evaluated often in recent reports but they are fundamental to real-world tasks such as navigation and planning. Table 3 provides a brief summary of each selected benchmark and capability, as well as a justification to why that benchmark is included in EUREKA-BENCH.

Ultimately, we consider the current list of benchmarks a contribution that needs to be maintained over time, with the assumption that the list needs to be completed (e.g. with math and planning benchmarks) and refreshed with new capabilities. Some benchmarks will also need to be deprecated as they get saturated with new releases. **Models.** Given the challenging nature of the selected benchmarks, and the goal of assessing frontier developments in AI, we consider a broad range of six model families but we only pick the most advanced models within each family. Table 2 shows all models included in the evaluation by modality.

Methodology and analysis. Rather than reporting overall single scores for each benchmark and model, we extract more granular and deeper insights that can better characterize failures of each model and also conduct meaningful comparisons. In particular, this report contributes three distinct types of analysis:

- **Disaggregated reports of model performance across important experimental conditions and subcategories of data.** Previous work in model evaluation and error analysis [38, 100, 78, 11] has shown that single-score evaluations can hide important failure modes. Here, we build upon the prior work and disaggregate performance across input attributes (i.e. subcategories of data) and experimental conditions relevant for the given capability. While analysis for subcategories slices the benchmark based on the content and nature of input prompts, experimental conditions (when applicable) add another dimension that relates to the nature of the task itself or how the model is prompted to perform the task. Table 1 shows all experimental conditions and subcategories studied per benchmark, and in-depth results are in sections 4 and 5.
- **Analysis of model non-determinism across identical runs.** For applications that use foundation models as a basis to their services, determinism of output is important for assuring reliable output and providing consistent experiences to end users. Throughout our experiments we observe that this property is not guaranteed from most generative models, even when the prompt is identical and generation temperature is set to zero. In Section 6, we compare all models in this report with respect to this phenomenon and show that, for many of them, there are major variations at the example level for identical runs.
- **Backward compatibility analysis within model families for measuring progress and regress upon model updates.** Backward compatibility was first studied for predictive machine learning [10, 102, 106, 71, 96] for measuring whether model updates can cause regressions in performance for individual examples or whole groups. This work showed that regressions can happen during updates even when overall accuracy increases. More recent findings have shown that the phenomenon can also happen for generative models with brittleness cases when identical prompts that had worked in older versions of the model do not work in newest updates [36, 21, 68]. Section 7 shows that across three model families (GPT, Claude, and Llama) incompatibility can be observed at both the example level as well as per data subcategory.

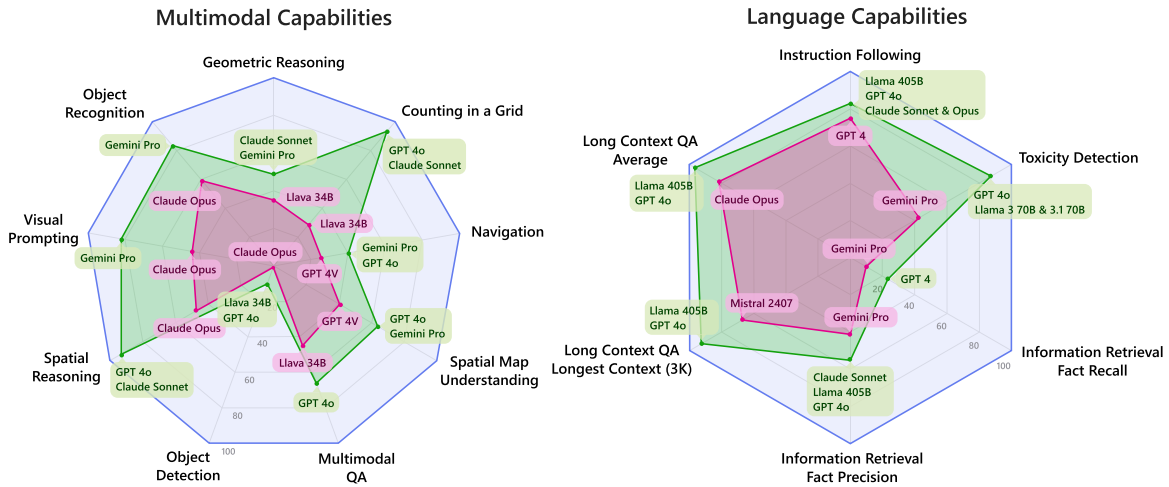


Figure 1: Performance of best and worst models for multimodal (left) and language (right) datasets in EUREKA-BENCH. The red frontier shows the performance of the worst model, indicating the area that is already solved for the set of capabilities. The green frontier shows the performance of the best model, indicating the best known result with current technology. The blue horizon between the best model and the maximum performance shows the room for improvement for mastering the capability. The best performance sets indicated in the green border include all models that perform within 2% of the best observed result.

2 Results Summary

Figure 1 is a high-level illustration of the state of the art in AI for EUREKA-BENCH, showing the best and the worst performance per capability. These results show a complementary picture of capabilities of different models and that there is no single model that outperforms all others in most tasks. However, Claude 3.5 Sonnet, GPT-4o 2024-05-13, and Llama 3.1 405B repeatedly outperform others in several capabilities.

Multimodal Evaluation: Evaluations on important vision-language capabilities such as geometric and spatial reasoning, object recognition and detection, multimodal question answering, and navigation demonstrate increased capabilities of most recent models when compared to their previous versions. For example, GPT-4o 2024-05-13 improvements over GPT-4 Vision Preview range between 3%-20%. Yet, state-of-the-art models are still fairly limited in their multimodal abilities, specifically when it comes to detailed image understanding (e.g. localization of objects, geometric and spatial reasoning, and navigation), which is most needed in truly multimodal scenarios that require physical awareness, visual grounding, and localization.

1. **State-of-the-art multimodal models struggle with geometric reasoning.** Reasoning about height is more difficult than about depth. Claude 3.5 Sonnet and Gemini 1.5 Pro are the best performing models for this task with Claude 3.5 Sonnet being the most accurate model for depth ordering and Gemini 1.5 Pro the most accurate for height ordering.
2. **Multimodal capabilities lag language capabilities.** On tasks which can be described either as a multimodal task or as language-only, the performance of most tested models is higher for the language-only condition. GPT-4o 2024-05-13 is the only model that consistently achieves better results when presented with both vision and language information, showing therefore that it can better fuse the two data modalities.
3. **Complementary performance across models for fundamental multimodal skills.** For example, Claude 3.5 Sonnet, GPT-4o 2024-05-13, and GPT-4 Turbo 2024-04-09 have comparable performance in multimodal question answering (MMMU) but they outperform all other models by at least 15%. There are tasks like object recognition and visual prompting where the performance of Claude 3.5 Sonnet is better or comparable to GPT-4o 2024-05-13, but Gemini 1.5 Pro outperforms them both. Finally, in tasks like object detection and spatial reasoning, GPT-4o 2024-05-13 is the most accurate model.

Language Evaluation: The evaluation through EUREKA-BENCH shows that there have been important advances from state-of-the-art LLMs in the language capabilities of instruction following, long context question answering, information retrieval, and safety.

1. **Faster improvements in instruction following across all model families.** Amongst the studied language capabilities, instruction following is where most models are improving faster, potentially due to strong investments in instruction tuning processes, with most models now having an instruction following rate of higher than 75%.
2. **All models’ performance in question answering drops with longer context.** When state-of-the-art models are compared in “needle-in-a-haystack” tasks, they seem to all perform equally well. However, testing the models on tasks that involve reasoning over long-context, reveals that all models’ performance drops as context size grows. Amongst all models, GPT-4o 2024-05-13 and Llama 3.1 405B have the lowest drop in performance for longer context.
3. **Major gaps in factuality and grounding for information retrieval from parametric knowledge or input context.** For example, we observe query constraint satisfaction rates (i.e. fact precision) of lower than 55%, completeness rates of lower than 25% (i.e. fact recall), and information irrelevance rates of higher than 20% (potentially information fabrication). Llama 3.1 405B, GPT-4o 2024-05-13, and Claude 3.5 Sonnet are the best performing models in this task across different conditions. GPT-4o 2024-05-13 and Claude 3.5 Sonnet in particular have significantly lower information irrelevance rates (associated with better factuality). Llama 3.1 405B has better constraint satisfaction rates (associated with better constrained text generation and grounding).
4. **High refusal rates. Lower accuracy in detecting toxic content vs. neutral content for most models.** While several models have high accuracy rates for toxicity detection, others (Gemini 1.5 Pro, Claude 3.5 Sonnet, Claude 3 Opus, and Llama 3.1 405B) exhibit low accuracy in classifying toxic content and a high amount of refusal. During the safe language generation evaluation, models like GPT-4 1106 Preview and Mistral Large 2407 have the highest toxicity rates. GPT-4o 2024-05-13 is the only model that has both a high toxicity detection accuracy and a low toxicity score for safe language generation, as shown in the discriminative and generative evaluations respectively.

Several models have highly non-deterministic output for identical runs. We study outcome determinism for all models by running three identical runs (temp=0, top_p= 0.95), and then report different measures of non-determinism such as disagreement, variation, and entropy of outcomes. Gemini 1.5 Pro, GPT-4 1106 Preview, GPT-4 Vision Preview, and GPT-4 Turbo 2024-04-09 show high non-determinism of outcomes. For example, there exists a 26% and a 14% disagreement across three runs of Gemini 1.5 Pro on a random sample of MMMU and IFEval respectively. This can translate to 1%-4% fluctuations in performance at the subcategory level for these benchmarks. These results raise important questions regarding the stability of user and developer experiences when repeatedly inferencing with identical queries using the same prompt templates. Llama 3 70B, Llama 3.1 70B, and Mistral Large 2407 are almost perfectly deterministic. The Claude family and GPT-4o 2024-05-13, and Llama 3.1 405B are more deterministic than Gemini 1.5 Pro and GPT-4 versions prior to GPT-4o 2024-05-13 but yet non-zero and sometimes highly non-deterministic for longer generative tasks.

Backward incompatibility for shifts within the same model family is prevalent across all state-of-the-art models. This is reflected in high regression rates for individual examples and at a subcategory level. This type of regression can lead to breaking trust with users and application developers during model updates. Regression varies per task and metric, but we observe several cases when it is higher than 10% across three model families (Claude, GPT, Llama), and sometimes they can dominate progress rates for whole subcategories of data.

The complementary nature of these results shows that there is space and demonstrated opportunity to improve current models in different areas, at least to the level of the best performing model for each individual capability in this challenge set. Despite this, several tasks in the challenge set we evaluate on remain difficult even for the most capable models and it is important to discuss and explore whether these gaps can be addressed with current technologies, architectures, and data synthesis protocols in place.

3 Evaluation Framework

In the fast-paced space of AI research, where new models and benchmarks are introduced and others are deprecated frequently, it is important for evaluation and understanding efforts to be able to reuse existing evaluation pipelines with minimal adjustments to efficiently accommodate new models and benchmarks. This calls for a modular design that allows the users to onboard a new benchmark or model by inheritance of pipeline definitions from existing experiments and implementing changes only where overriding the existing pipeline is necessary.

Benchmark Modality	Capability	Role in EUREKA-BENCH
GeoMeter Image → Text Section 4.1	Geometric Reasoning	Task: Predicting depth and height orderings. Capability importance: Depth and height understanding are a fundamental cognitive capability for natural language interaction with physical environments and for navigation. State-of-the-art: Even most capable models have an accuracy of less than 50% in the task.
MMMU Image → Text Section 4.2	Multimodal QA	Task: Question answering for image content across different disciplines. Capability importance: Reasoning over image content is important for assessing multimodal knowledge and for measuring multimodal understanding skills. State-of-the-art: Even most capable models have an accuracy of less than 70% in the task.
Image Understanding Image → Text Section 4.3	Object Recognition Object Detection Visual Prompting Spatial Reasoning	Task: Recognizing and detecting objects in an image or a given section of the image, reasoning about the spatial relationships between them. Capability importance: Most tasks in this benchmark are representative of classic vision tasks for which traditional ML and vision methods have matured, but generative models have not yet caught up. All tasks are important for understanding and localizing image content, relevant to almost all multimodal applications. State-of-the-art: There is a large variance across models and capabilities in this benchmark. The goal of the evaluation in this benchmark is to characterize this variance. While there are a few cases and models where performance is higher than 80%, for most models this is not the case.
Vision Language Understanding Image → Text Section 4.4	Spatial Understanding Navigation Counting	Task: A set of synthetic tasks whose input can be phrased either in language, image, or both. Capability importance: The goal here is to disentangle the role and model performance for each modality, for models that support both vision and language as input. Spatial Understanding and Navigation are chosen as reasoning capabilities. Counting is chosen as a basic capability for which several models still struggle with, in at least one of the modalities. State-of-the-art: There is a large variance across models, modalities, and capabilities in this benchmark, with most models performing at less than 70% spatial understanding and less than 50% in navigation. The goal of the evaluation in this benchmark is to characterize this variance.
IFEval Text → Text Section 5.1	Instruction Following	Task: Instruction following for formatting, styling, and organizing generated text. Capability importance: Horizontal language task that impacts several generative writing scenarios that require generation with specific user constraints. State-of-the-art: Overall accuracy for most capable models ranges between 70% and 85%. However, there are several instruction subcategories and models for which performance is lower and for which there is higher variance.
FlenQA Text → Text Section 5.2	Long Context QA	Task: Answering logical reasoning questions on long context. Capability importance: Reasoning upon several statements distributed across long context is relevant for long document understanding that goes beyond merely retrieving simple facts from context (i.e. needle-in-the-haystack tasks). State-of-the-art: While most models perform well in this benchmark for short input context, as context length increases, many models are not able to maintain good accuracy with several of them having an accuracy of less than 80%.
Kitab Text → Text Section 5.3	Information Retrieval	Task: Retrieving long-form information from the model’s parametric knowledge or from given input context with filtering constraints. Capability importance: All information retrieval tasks involve some form of constraint that defines the retrieval query. However, other simpler IR benchmarks only test for short-form generation (finding a single fact) and for a single constraint. Being able to answer more complex queries is relevant to the factuality and grounding of advanced search and information finding. State-of-the-art: Constrained retrieval from parametric knowledge is still prone to major irrelevance and fact fabrication with constraint satisfaction being less than 60%. Constrained retrieval from given input context is significantly better in overall, but for queries with more than one constraint constraint satisfaction and completeness drop to less than 70% and 60% respectively.
Toxigen Text → Text Section 5.4	Toxicity Detection Safe Language Generation	Task: Classifying whether a given text is toxic or neutral (discriminative case). Prompting the model with questions/statements that could lead to unsafe language generation, and testing the safety of the generated language (generative case). Capability importance: Ability to distinguish between toxic and neutral language is relevant for dialogue comprehension and also for establishing the model utility on content moderation scenarios. Safe language generation itself is an important capability that impacts different aspects of responsible AI including representational fairness, inclusion, and safety. State-of-the-art: State-of-the-art models face two types of challenges in this benchmark. First, there are discrepancies between performance of models across different demographic groups. Second, some of the models exhibit a high refusal rate for the toxicity detection task, which makes these models not useful for content moderation and other scenarios where toxicity detection is key.

Table 3: The role of each benchmark in EUREKA-BENCH and the importance of each capability for measuring progress in AI.

All experiments reported in this paper were conducted through EUREKA, a software framework that unifies all of the benchmarks and models used in this report. This framework was designed to ensure reproducibility, composability, and reusability. We have opened the source code of EUREKA to maximize transparency into the details of all benchmarks and evaluation settings.

The EUREKA framework currently supports both language and multimodal (text and image) data, and allows the user to define custom pipelines for data processing, inference, and evaluation, with the possibility to inherit from existing pipelines to minimize development work. We have released the experiment pipelines for all benchmarks listed in Table 1.

Each experiment under the EUREKA framework is defined using a `Pipeline`. These pipelines are comprised of a set of `Components`. This modular design not only improves readability, but also allows extendability and reusability as users can inherit from an existing `Pipeline` and only implement minimal changes.

Currently, EUREKA operates with the following set of `Components`:

- `PromptProcessing`: This component is used to prepare data for inference, apply data manipulations, or apply complex prompt templates. It enables flexible and reproducible experimentation with different prompt templates and data pre-processing steps.
- `Inference`: This component performs model inference on any processed data. For example, inference can be done for the model that is subject to evaluation, or a model that is involved in the evaluation pipeline as an evaluator of the original model’s output.
- `DataProcessing`: This component is used to post-process the model outputs to extract the model response from the generated text. This is necessary for example when a regex search of the option selected is needed in multiple-choice scenarios or when the generated text includes tags that were used in training (e.g. `|assistant|`) and need to be removed before metric calculation.
- `EvalReporting`: This component facilitates evaluation of the processed model outputs using various metrics and an arbitrary number of aggregations, and logs the metric results for individual prompts as well as aggregation results.
- `DataJoin`: This component is used to join two sources of data, for example to join the model outputs with the ground truth data for evaluation.

Importantly, all `Components` log their outputs in standardized `jsonl` files, which increases transparency into each step of the evaluation, facilitates error analysis, and streamlines result analysis and visualization across multiple experiments.

Furthermore, the `Components` make use of utility classes, including but not limited to `DataLoaders`, `Models`, `Metrics`, and `Aggregators`. Each utility class is configurable using its corresponding `Config` class. All `Configs` and individual parameters defining an experiment `Pipeline` are logged in the experiment directory as parts of the `Pipeline` config to ensure reproducibility. EUREKA also provides implementations of `Models` for inference either through APIs or local deployments.

For an overview of two example experiment pipelines available in EUREKA, see Figure 2. The top part of the figure shows the evaluation pipeline for the Toxigen Generative benchmark. First, the `PromptProcessing` component reads the Toxigen data from HuggingFace and prepares the prompts for inference. Next, the `Inference` component is used to inference the model under evaluation, in this example Llama 3 70B. After that, the `PromptProcessing` component is reused, this time to load the inference results and prepare prompts for the judge model using a prompt template. The `Inference` component is then reused to inference the judge model and score the original inference results. Finally, the `DataProcessing` component is used to extract scores from the judge model inference results. The scores are aggregated in several ways in the `EvalReporting` component.

The bottom part of the figure shows the components comprising the GeoMeter experiment pipeline. This pipeline is different from Toxigen in two important regards: it deals with multimodal data and it does not use another model as a judge, instead it offers a metric class to score the inference results. Despite these differences, the same components can be reused with minimal adjustments to accomodate this scenario. Namely, the `PromptProcessing` component is set to read multimodal data from a local directory, and the `EvalReporting` component is set to use the GeoMeter metric class. As shown in both pipelines, the `Inference` component can be easily configured to use the desired model, enabling fair comparison between models without changing the rest of the pipeline, and with maximum code reuse.

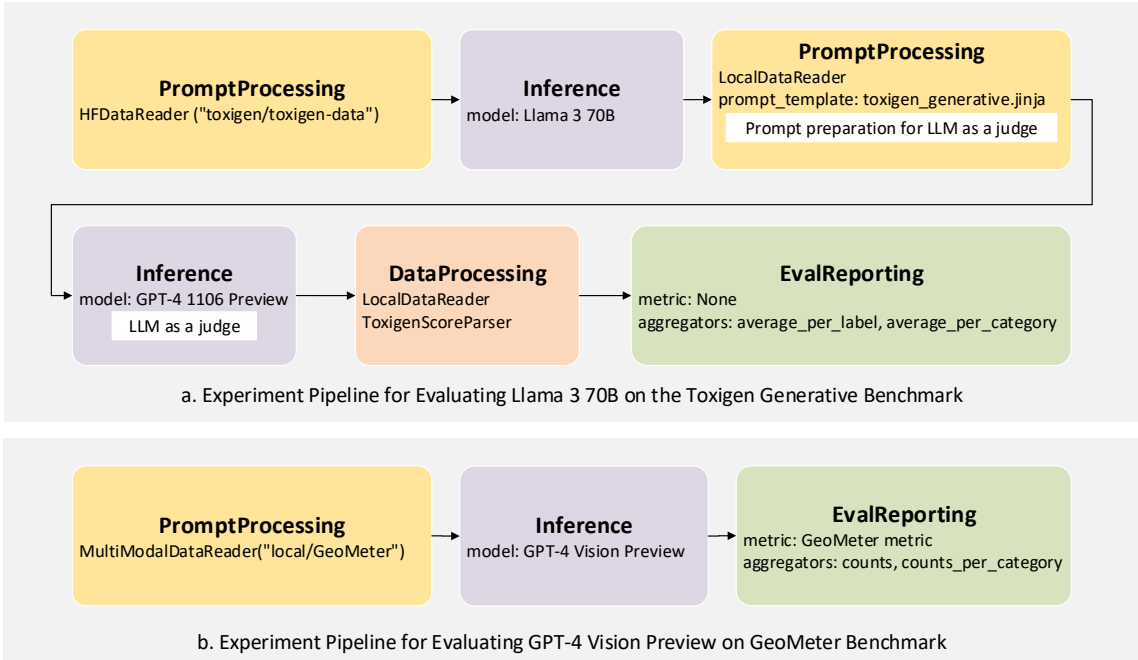


Figure 2: Overview of experiment pipelines for two example evaluation experiments: Toxigen Generative (a) and GeoMeter (b). Components are configurable at instantiation time to maximize code reuse and enable controlled experimentation. For example, the PromptProcessing component is shown here to use different data readers or prompt templates in different contexts.

4 Multimodal Evaluation

In this section, we provide detailed analysis and results for the capabilities of geometric reasoning (GeoMeter), multimodal question answering (MMMU), object recognition, object detection, visual prompting, spatial understanding and reasoning, navigation, and counting.

To account for the impact of non-determinism (discussed in Section 6), all experiments reported here were repeated three times and we report the mean and corresponding standard error across the three repeated runs with temperature set to zero and top_p = 0.95.

4.1 Geometric Reasoning - GeoMeter

Motivation: The ability to understand visual properties such as size, shape, depth, and height is fundamental to visual understanding, yet many existing Visual Question Answering (VQA) benchmarks [50, 20, 63, 32, 104] do not specifically focus on the depth and height perception capabilities of Vision Language Models (VLMs). Accurate perception of these dimensions is vital for practical applications like scene understanding, navigation, monitoring, and assistive technologies. The lack of accurate depth and height understanding in VLMs can lead to serious consequences, such as misjudging the proximity of objects, which could result in catastrophic outcomes in real-world scenarios.

Despite VLMs’ abilities to recognize object shapes and sizes, their depth and height reasoning often relies on learned size/shape cues rather than actual geometric analysis, potentially influenced by biases from training data [48]. Alternatively, models might estimate the depth based on the apparent size of objects, without genuine inter-object reasoning. Additionally, when faced with multiple choices, VLMs might also show bias towards certain answers, influenced by the prevalence of similar data during training. Thus, it becomes important working with focused benchmarks that enhance understanding of true depth and height perception in VLMs, ensuring they perform reliably in complex, real-world environments.

Here, we use **GeoMeter**, a geometric reasoning benchmark derived from previous work [9], which is specifically designed to evaluate the depth and height reasoning capabilities of Vision Language Models (VLMs). GeoMeter comprises approximately *1086 unique image-text pairs* across two tasks: depth and height. The data

Dataset	Subcategory	Task	Question Type	Questions	Query attributes	Img-Text pairs
GeoMeter	Depth	VQA	MCQ	986	Color, Numeric label (random and patterned)	1086
	Height			100	Color, Numeric label (random)	

Table 4: Dataset statistics of GeoMeter. Here, query attributes are unique identifiers for the object of interest. MCQ denotes Multiple Choice Questions.

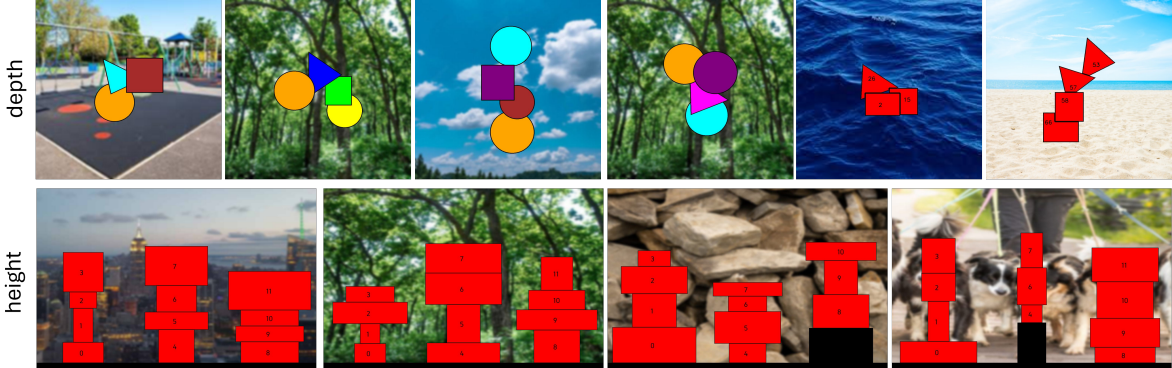


Figure 3: Samples from the GeoMeter dataset. Here, each sample is shown with random query attributes including color - numeric label and color - shape label.

consists of synthetic examples depicted by 2D shapes like triangles, squares, rectangles, circles etc. The development of synthetic datasets featuring basic shapes aims to genuinely test the visual reasoning capabilities of models, focusing on their ability to process visual information without relying on familiar real-world cues and biases. Our motivation comes from concerns about test time data leakage that could arise when models, trained on vast existing datasets, encounter images during testing that they may have already seen during training. By using unique datasets, we seek to ensure a more accurate evaluation of the model’s true visual reasoning abilities.

Benchmark Description: GeoMeter consists of synthetic 2D images to test model performance on depth and height perception tasks. Table 4, Figure 3 and Figure 4 respectively show the dataset statistics, sample images and sample image-text pair of our proposed datasets. The dataset generation can be divided into two parts - Image generation and Question generation.

Image Generation. The synthetic dataset is divided into two subcategories - *depth* and *height*, with each image containing a real-world image as a background to enhance realism. The dataset consists of 1086 image-questions pairs. The *depth* category consists of 986 images, featuring rectangles, triangles, or circles that partially overlap to create a depth illusion, with unique identifiers such as colors, and numeric labels. The *height* category has 100 images, where each tower consists of four rectangles with random dimensions. Further, in these images, towers are placed on a horizontal black strip that is treated as a raised platform. This category includes two sets: one with all towers placed at the same height level and another with a randomly chosen tower on a raised platform, with unique identifiers being label. All towers are labeled sequentially.

Question Generation: The following method for generating questions is applied consistently across all images. Each question is composed of two key components: *Description Prompt* and an *Answer Format Instruction*. The *Description Prompt* provides general information about the scene, offering semantic cues related to the image. This is then followed by the actual question and the *Answer Format Instruction*. For instance, the *Description Prompt* might be: “[additional information] Provide depth/height ordering for the shapes <question items> in the image. [additional information]”. This is then followed by an *Answer Format Instruction* such as: “From the given options: <answer set>, select the correct answer [additional information].” Each question is constructed by sequentially combining the *Description Prompt* and *Answer Format Instruction*.

The *question items* is a list containing <query attribute> appended by <shape>. Here <query attribute> is one of the unique identifiers of the dataset. For example in the question item “red circle”, “red” is the <query attribute> and <circle> is the shape. The *answer set* contains all possible valid values (<query attribute> + <shape>) to that given prompt. To generate both the question items and answer set, we read through the scene


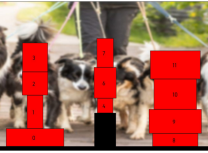
	<p>Prompt example for depth ordering task</p> <p>The image shows 2D shapes placed randomly. The shapes overlap each other, creating a depth effect. When two shapes are overlapping, the shape that is complete is defined to be on top of the partially hidden shape. Each 2D shape has a number written over them which we call shapeID and must be inferred as the label for the corresponding shape. Provide depth ordering from top to bottom for the shapes (58, 57, 53) in the image. Answer in the format: (ShapeID, ShapeID, ...). For example, (110, 98, 217) is a valid answer. From the given options: ['(58, 57, 53)', '(58, 53, 57)', '(57, 58, 53)', '(53, 58, 57)', '(57, 53, 58)', '(53, 57, 58)'], select the correct answer (ONLY output the answer).</p>
	<p>Prompt example for height ordering task</p> <p>The image shows 2D red rectangles stacked on top of each other. There are multiple stacks in the image. The black region at the bottom of the image is the ground level, and is where the base of the stack lies. The height of each stack is measured from its base. Each 2D shape has a number written over them which we call ShapeID and must be inferred as the label for the corresponding shape. The stacks are labelled A, B... from left to right. Order the stacks labelled A, B, C from shortest to tallest. Answer in the format A, B, C... where A, B, C... are the labels of the stacks. From the given options: ['(B, A, C)', '(B, C, A)', '(C, A, B)', '(A, C, B)', '(A, B, C)', '(C, B, A)'], select the correct answer (ONLY output the answer).</p>

Figure 4: Sample image-text pair from the GeoMeter dataset. Here the image contains 5 shapes labeled with random numeric labels which are used as query attributes in the prompt. Prompt template shows the basic template for each image-text pair of all our benchmark, where the prompt example is the actual prompt for this image. The prompt example is appended with either MCQ or True/False type question.

Model	Depth ordering (%)	Height ordering (%)	Overall (%)
Claude 3 Opus	42.42 ± 0.03	17.00 ± 0.00	40.10 ± 0.00
Claude 3.5 Sonnet	50.70 ± 0.00	28.66 ± 0.33	48.66 ± 0.03
Gemini 1.5 Pro	47.59 ± 0.14	32.00 ± 1.00	46.16 ± 0.18
GPT-4 Turbo 2024-04-09	38.84 ± 0.27	13.00 ± 1.00	36.46 ± 0.32
GPT-4 Vision Preview	40.12 ± 0.44	13.33 ± 0.33	37.66 ± 0.41
GPT-4o 2024-05-13	43.91 ± 0.00	19.33 ± 0.67	41.63 ± 0.07
Llava 1.6 34B	37.01 ± 0.10	15.00 ± 0.00	35.00 ± 0.10

Table 5: Performance comparison of the studied models on proposed benchmark. The reported results are average accuracy and standard error on three runs across depth and height subcategories. Top scores are in bold.

graph and run depth-first search on it to generate valid unambiguous values of object-pair relationship. For each image, there are several multiple choice questions. For MCQ, the order of the given options is randomly generated, and ground truth is randomly placed in one of those options. Additionally, the answer for each question has been manually checked by the dataset creators.

Aggregate and subcategory results: We evaluate our benchmark on the task of visual question answering (VQA), with accuracy being the performance metric on MCQ type questions. Evaluation is done across query attributes and number of shapes on probing the VLMs’ depth and height perception. The performance of the selected models on the VQA task for MCQ type questions on the proposed benchmarks is shown in Table 5, where each row corresponds to the average accuracy across all different query attributes and shapes. Depth and height subcategory results are also presented in Table 5.

Main observations are as follows. First, Claude 3.5 Sonnet achieves the best performance compared to the other models: GPT-4o 2024-05-13, GPT-4 Turbo 2024-04-09, Gemini 1.5 Pro. Additionally, Claude 3.5 Sonnet is also more consistent across multiple runs using the same prompts, as shown by the low standard error across three different runs.

Analysis and Discussion

Models generally struggle in depth and height perception tasks. Results from Table 5 highlight that the foundation multi-modal models struggle significantly with depth and height perception tasks involving similar shapes. This discrepancy underscores our benchmark’s value in identifying gaps in VLMs’ capabilities to handle more complex geometric reasoning, beyond mere shape recognition. To further support our claim that the low performance of models on the GeoMeter data is due to VLMs’ deficiencies in depth and height reasoning, we also provide reference to relevant observations from prior work [9]. First, they note that VLMs generally perform well on basic geometric tasks such as line understanding, shape recognition, and shape counting, but fail on advanced perception tasks like depth and height perception tasks. Further, they also observe that VLMs

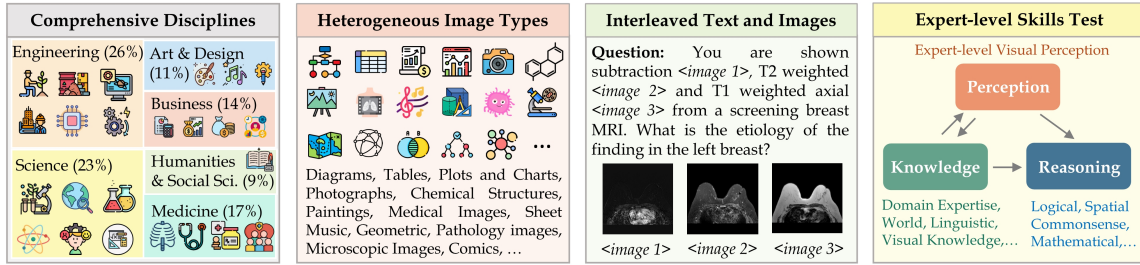


Figure 5: The MMMU dataset is a set of visual question-answering task that is comprehensiveness across 11.5K college-level problems across six broad disciplines and 30 subject-areas, and it requires detailed image understanding and reasoning requiring in deep subject knowledge.

struggle with depth and height reasoning tasks on real-world data as well.

Models generally struggle more in height perception than depth. Table 5 indicates that both open and closed models struggle more with height perception compared to depth questions, with depth perception generally aided by occlusion cues and height perception challenged by the complexity of assessing vertically stacked objects. This difference indicates that models may find it somewhat easier to interpret scenarios with partial obstructions than to precisely evaluate complex vertical arrangements.

Overall, while models perform better in depth perception, they still show limitations in comprehensively handling more complex geometric understanding tasks, underscoring an area for improvement in advanced geometric task perception.

Main takeaways

- State-of-the-art multimodal models struggle in depth and height perception tasks.
- Generally models show better depth perception than height.
- Claude 3.5 Sonnet and Gemini 1.5 Pro are the best performing models for this task with Claude 3.5 Sonnet being the most accurate model for depth ordering and Gemini 1.5 Pro the most accurate for height ordering.

4.2 Multimodal Question Answering - MMMU

Motivation: A key use case for multimodal models is to serve as an expert assistant to answer queries and provide information and context about images. This Visual Question Answering setup is one of the core tasks for multimodal models. It combines the abilities of understanding images at a high and detailed level with the ability of reasoning using that understanding. MMMU [123] is a popular dataset that tests these capabilities across a broad range of topics requiring deep image understanding and domain-specific knowledge. We have included it in our evaluations due to this broad and deep coverage, wide adoption in the research and industry communities, and that it remains a challenging dataset that no models have yet mastered. Thus it provides a good high-level measure of multimodal reasoning performance.

Benchmark Description: MMMU tests multimodal multi-discipline reasoning in six core disciplines: Art and Design, Business, Science, Health and Medicine, Humanities and Social Science, and Tech and Engineering. The questions span 30 subject areas and 183 subfields, with a wide-variety of image types, such as charts, diagrams, maps, tables, music sheets, and chemical structures. Questions are both multiple-choice and open-ended. An illustration of the disciplines, subjects, images, and questions appears in Figure 5. For our evaluations, we use the 900-question validation set that spans all subject-areas.

Aggregate Baseline Results: As a baseline evaluation we use the prompt formatting that is provided by the MMMU evaluation codebase [4], which concatenates each question with the appropriate answer choices:

- Multiple-choice prompt example: “A recent study found that the demand and supply schedules for Frisbees are as follows:< image 1>Frisbee manufacturers persuade the government that Frisbee production improves scientists’ understanding of aerodynamics and thus is important for national security. A concerned Congress

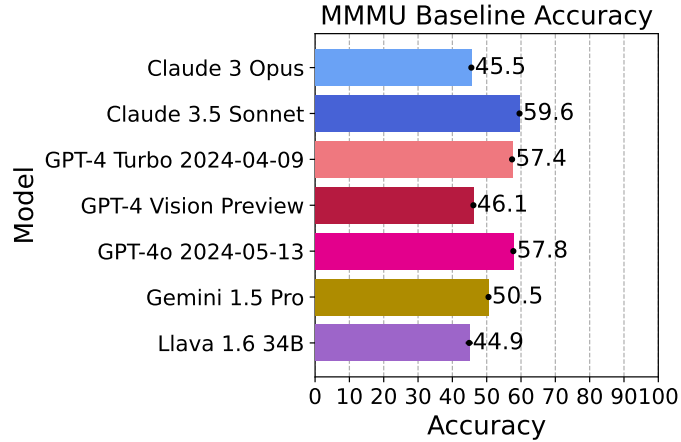


Figure 6: Aggregate accuracy reported across three different runs per model.

votes to impose a price floor \$2 above the equilibrium price. What is the new market price? (A) 8 (B) 9 (C) 10 (D) 11.”

- Open-ended prompt example: “The graph below shows the supply and demand curves and the world price for bagels. < image 1 >What is the equilibrium price if this country does not trade?”

The instruction prompts are as follows:

- Multiple-choice: “{prompt} Answer with the option’s letter from the given choices directly.”
- Open-ended “{prompt} Answer the question using a single word or phrase.”

This combined prompt is fed to each model as a user-prompt.

5% of these questions require reasoning over more than one image. This is supported for all models except Llava 1.6 34B, and thus for that model these questions are marked as unanswerable and thus incorrect.

Figure 6 presents the average accuracy for the entire validation dataset across the seven multimodal models evaluated in this report. Claude 3.5 Sonnet is the best performing model out-performing GPT-4o 2024-05-13 by 2.2%, followed closely by GPT-4 Turbo 2024-04-09 and then Gemini 1.5 Pro. Llava 1.6 34B is the worst performing model overall, but as an open source model fairs well and is close in performance to Claude 3 Opus.

Discipline Level Baseline Results: Figure 7 present the discipline-level accuracy, as defined by the six core disciplines in the benchmark: Art & Design, Business, Science, Health & Medicine, Humanities & Social Science, and Tech & Engineering. Here we see the close performance of the top two models Claude 3.5 Sonnet and GPT-4o 2024-05-13, with different per-subject wins. Claude 3.5 Sonnet has big leads in Business and Tech and Engineering but lags GPT-4o 2024-05-13 in all other disciplines, with GPT-4o 2024-05-13 having the best relative performance in Science and Humanities and Social Sciences. Science, Tech and Engineering remain the most difficult disciplines across all models.

Aggregate Zero-shot CoT and Expert Prompt Results: We note than many current evaluations on MMMU use engineered prompts for better performance, leading to higher accuracy results than shown in the previous section. Thus, to provide some insight into this we performed our evaluations on the top model in each family using a modified system-level *expert* instruction prompt, as specified in the OpenAI Evals codebase [5].

- Multiple-choice: *You are an expert in {subject} whose job is to answer questions from the user using images. First, reason about the correct answer. Then write the answer in the following format where X is exactly one of A,B,C,D: “ANSWER: X”. If you are uncertain of the correct answer, guess the most likely one.*
- Open-ended: *You are an expert in {subject} whose job is to answer questions from the user using images. First, reason about the correct answer. Then write the answer in the following format where X is only the answer and nothing else: “ANSWER: X”*

Here {subject} is one of the 30 subjects associated with each question, e.g., Art, Chemistry, Clinical Medicine, etc.

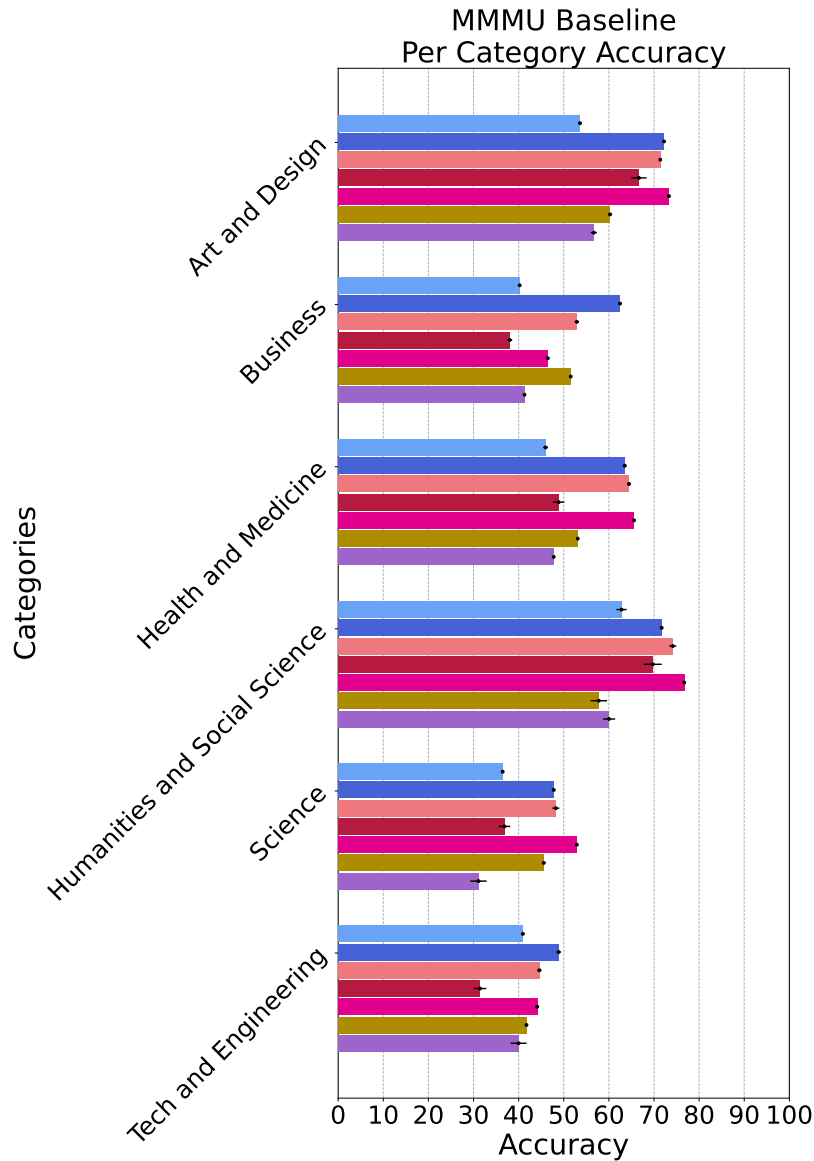
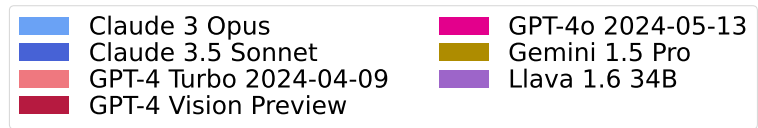


Figure 7: Accuracy per discipline reported across three different runs per model.

Next, we remove the instruction component from the user prompt that was used in the baseline results, as there is now an instruction prompt in the system prompt, thus the following prompts:

- Multiple-choice: “{prompt} Answer with the option’s letter from the given choices directly.”
- Open-ended “{prompt} Answer the question using a single word or phrase.”

are changed to have a Zero-Shot Chain-of-Thought (CoT) prompt, as such:

- Multiple-choice and Open-ended: {prompt} *Let’s think step by step.*

As shown in Figure 8, with the Zero-shot CoT Expert Prompt strategy, most models have a significant performance boost over the baseline. Largest increases are observed for Gemini 1.5 Pro and GPT-4o 2024-05-13, which have a 8.3% increase in accuracy each. Claude 3.5 Sonnet has the smallest increase of 4.3%. GPT-4o 2024-05-13 is the best performing model by 2.1%.

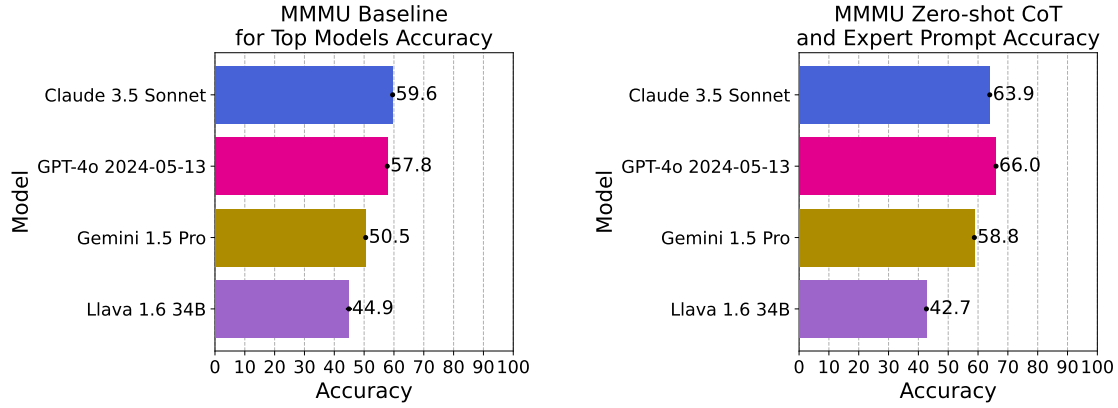


Figure 8: Aggregate accuracy for top models per family comparing the Baseline and Zero-shot CoT and Expert Prompt conditions. Reported across three different runs per model.

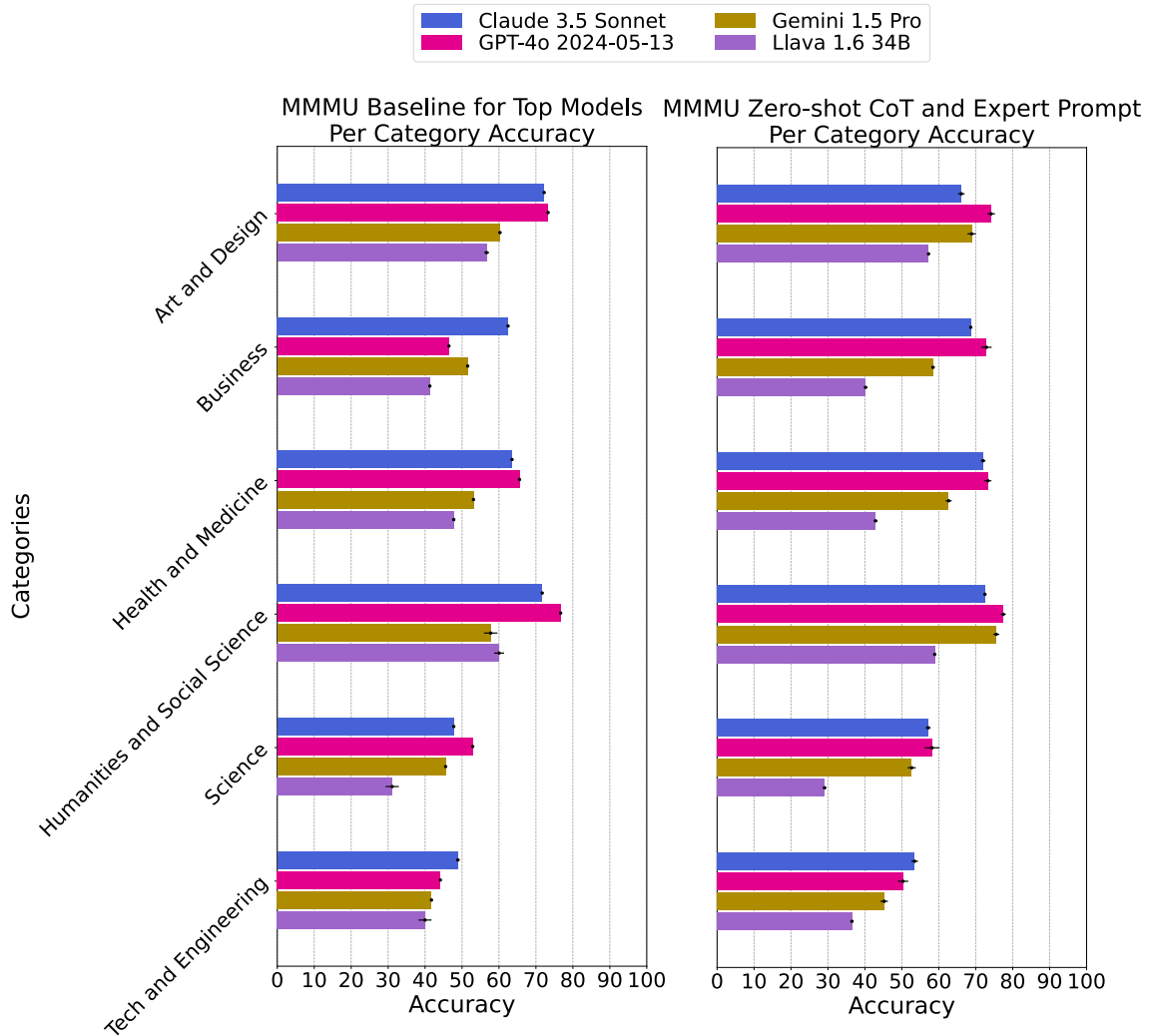


Figure 9: Accuracy per discipline for top models per family comparing the Baseline and Zero-shot CoT and Expert Prompt conditions. Reported across three different runs per model.

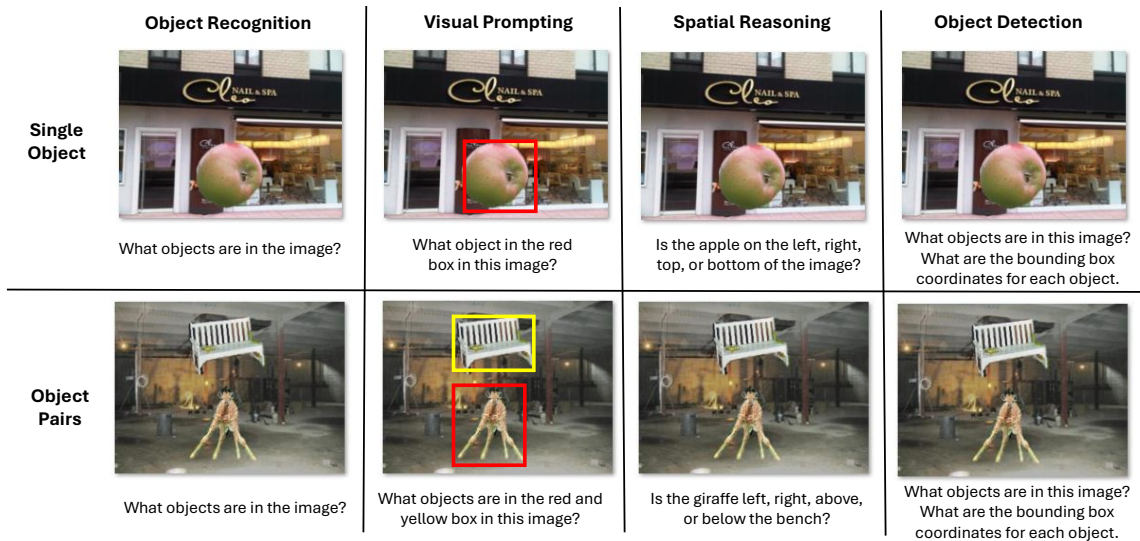


Figure 10: Illustration of the tasks in the Image Understanding Benchmark: Object Recognition, Visual Prompting, Spatial Reasoning, and Object Detection. For each task, the images consist of images of pasted objects on random images. For each there are 2 conditions “single object” and “object pairs”. Each test set has 1280 images and text pairs.

The Llava 1.6 34B model is the only one whose accuracy decreases. Llava 1.6 34B is not specifically trained to handle a system role, thus the system prompt is concatenated with the user prompt [2], which is a likely explanation for the different behavior over other models.

Discipline Level Zero-shot CoT and Expert Prompt Results: As shown in Figure 9, the accuracy of GPT-4o 2024-05-13 increases noticeably in every discipline, from which the largest increases are observed in Business. Claude 3.5 Sonnet has more modest gains and has a regression on Art & Design.

Main takeaways

- Claude 3.5 Sonnet and GPT-4o 2024-05-13 are the leading models for multimodal question answering as measured by the MMMU dataset, indicating better multimodal understanding skills and knowledge.
- The MMMU benchmark remains a challenging task for all models, with the best performance in the mid 60s percentage range.
- The performance of models is highly dependent on how they are prompted, with improvements and regressions across topics. The role of the prompt in evaluations cannot be ignored. This is an area that requires further investigation.

4.3 Image Understanding

Motivation: A key question for understanding multimodal performance is analyzing the ability for a model to have basic vs. detailed understanding of images. These capabilities are needed for models to be used in real-world tasks, such as an assistant in the physical world. While there are many dataset for object detection and recognition, there are few that test spatial reasoning and other more targeted task such as visual prompting. The datasets that do exist are static and publicly available, thus there is concern that current AI models could be trained on these datasets, which makes evaluation with them unreliable. Thus we created a dataset that is procedurally generated and synthetic, and tests spatial reasoning, visual prompting, as well as object recognition and detection [91]. The datasets are challenging for most AI models and by being procedurally generated the

benchmark can be regenerated ad infinitum to create new test sets to combat the effects of models being trained on this data and the results being due to memorization.

Benchmark Description: This dataset has 4 sub-tasks: Object Recognition, Visual Prompting, Spatial Reasoning, and Object Detection. For each sub-task, the images consist of images of pasted objects on random images. The objects are from the COCO [62] object list and are gathered from internet data. Each object is masked using the DeepLabV3 object detection model [22] and then pasted on a random background from the Places365 dataset [132]. The objects are pasted in one of four locations, top, left, bottom, and right, with small amounts of random rotation, positional jitter, and scale.

There are 2 conditions “single” and “pairs”, for images with one and two objects. Each test set uses 20 sets of object classes (either 20 single objects or 20 pairs of objects), with four potential locations and four backgrounds classes, and we sample 4 instances of object and background. This results in 1280 images per condition and sub-task. An example of each is shown in Figure 10, and examples of the prompts are as follows:

- Object Recognition: *What objects are in this image?*
- Visual Prompting:
 - One Object: *What object is in the red box in this image?*
 - Two Objects: *What objects are in the red and yellow box in this image?*
- Spatial Reasoning:
 - One Object: *Is the potted plant on the right, top, left, or bottom of the image? Answer with one of (right, bottom, top, or left) only.*
 - Two Objects: *Is the bottle above, below, right, or left of the keyboard in the image? Answer with one of (below, right, left, or above) only.*
- Object Detection: *You are an object detection model that aims to detect all the objects in the image. Definition of Bounding Box Coordinates: The bounding box coordinates (a, b, c, d) represent the normalized positions of the object within the image: a: The x-coordinate of the top-left corner of the bounding box, expressed as a percentage of the image width. It indicates the position from the left side of the image to the object’s left boundary. The a ranges from 0.00 to 1.00 with precision of 0.01. b: The y-coordinate of the top-left corner of the bounding box, expressed as a percentage of the image height. It indicates the position from the top of the image to the object’s top boundary. The b ranges from 0.00 to 1.00 with precision of 0.01. c: The x-coordinate of the bottom-right corner of the bounding box, expressed as a percentage of the image width. It indicates the position from the left side of the image to the object’s right boundary. The c ranges from 0.00 to 1.00 with precision of 0.01. d: The y-coordinate of the bottom-right corner of the bounding box, expressed as a percentage of the image height. It indicates the position from the top of the image to the object’s bottom boundary. The d ranges from 0.00 to 1.00 with precision of 0.01. The top-left of the image has coordinates (0.00, 0.00). The bottom-right of the image has coordinates (1.00, 1.00). Instructions: 1. Specify any particular regions of interest within the image that should be prioritized during object detection. 2. For all the specified regions that contain the objects, generate the object’s category type, bounding box coordinates, and your confidence for the prediction. The bounding box coordinates (a, b, c, d) should be as precise as possible. Do not only output rough coordinates such as (0.1, 0.2, 0.3, 0.4). 3. If there are more than one object of the same category, output all of them. 4. Please ensure that the bounding box coordinates are not examples. They should really reflect the position of the objects in the image. 5. Report your results in this output format: (a, b, c, d) - category for object 1 - confidence (a, b, c, d) - category for object 2 - confidence ... (a, b, c, d) - category for object n - confidence.*

Results: As shown in Figure 11, for Object Recognition and Visual Prompting, Gemini 1.5 Pro is consistently the best model by a range of 2-12% across these tasks and the one and two object conditions. GPT-4o 2024-05-13, Claude 3.5 Sonnet, and Llava 1.6 34B all come in second within a few percentage points of each other, without a consistent clear winner between them.

One interesting observation is that Visual Prompting leads to a small drop in model performance relative to Object Recognition. In cases involving two objects, we see a substantial decline across nearly all models, while in one-object cases, there is a modest drop for some models. This is surprising given we would expect the visual prompt to help focus the model and thus improve results [103, 120].

For Spatial Reasoning, we see GPT-4o 2024-05-13 as the overall best across both the one and two object conditions; however, within each condition there is a different story. Llava 1.6 34B excels at the one object

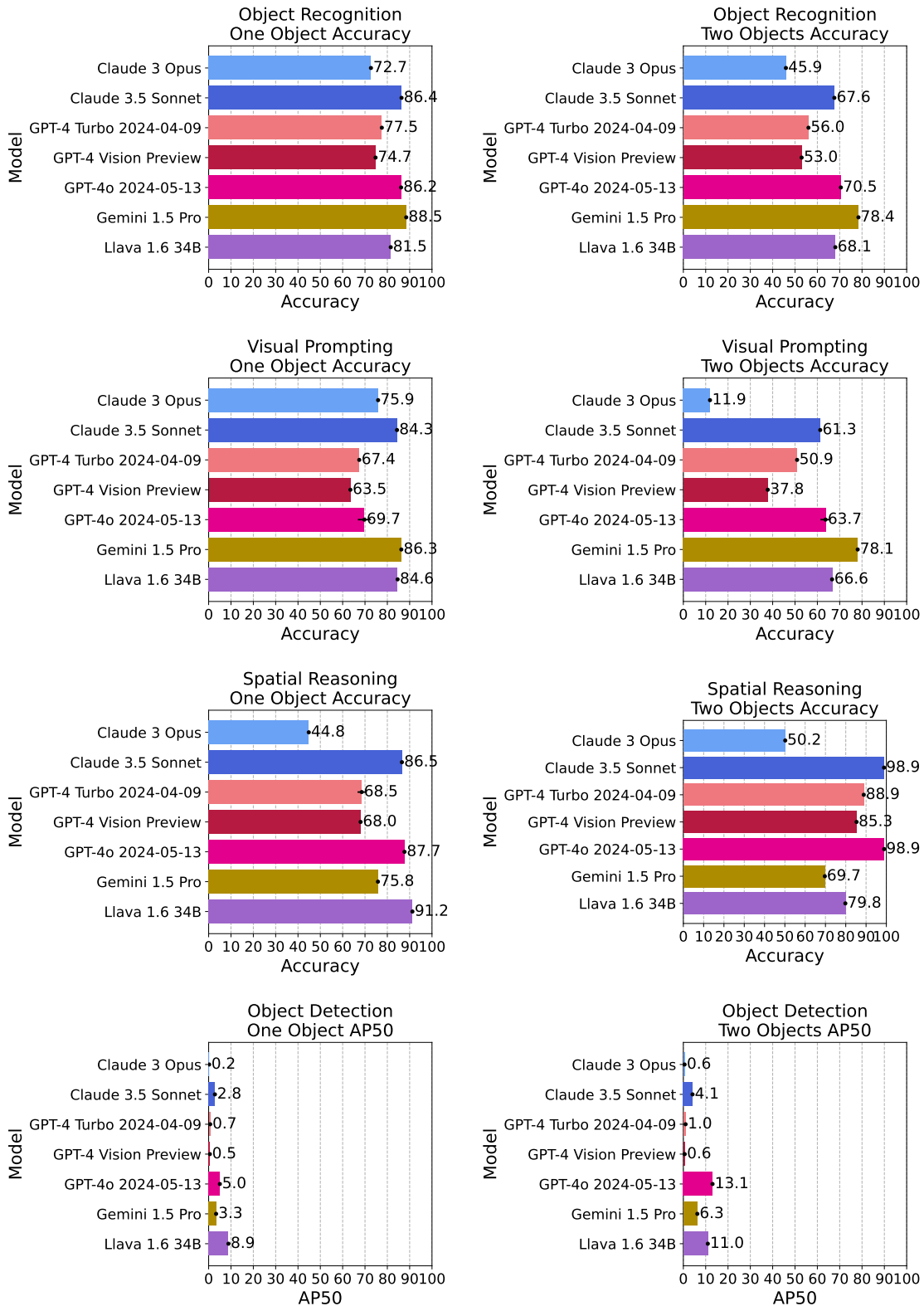


Figure 11: Image Understanding accuracy broken down per sub-task and condition reported across three different runs per model.

condition, with 91.2% accuracy, 3.5% better than GPT-4o 2024-05-13, but is far behind GPT-4o 2024-05-13 and Claude 3.5 Sonnet with two objects, which are tied at 98.9%. The Spatial Reasoning task is interesting as there are incredible gains in performance with new models in each family. For example, the one object condition had 50-60% accuracy in earlier models (Claude 3 Opus, GPT-4 Turbo 2024-04-09), and is now in the 80-th percentiles for accuracy.

The average accuracy for the top models across Object Recognition, Visual Prompting, and Spatial Reasoning and the one and two object conditions are Claude 3.5 Sonnet at 80.83%, Gemini 1.5 Pro at 79.47%, GPT-4o 2024-05-13 at 79.45%, and Llava 1.6 34B at 78.63%. Gemini 1.5 Pro has the most wins for these three sub-tasks and conditions, but Claude 3.5 Sonnet has the best average performance by a little over 1%.

Object Detection is a task that requires very detailed level image understanding and localization and this is one task where all models perform poorly. The best performing model is GPT-4o 2024-05-13 at AP50 13.1 in the two object condition and this is far below the performance of a state-of-the-art object detector [136].

When we add Object Detection in the mix and consider average performance across all sub-tasks and conditions, the order flips, due to the stronger object detection performance of GPT-4o 2024-05-13 and Llava 1.6 34B with GPT-4o 2024-05-13 at 61.85%, Claude 3.5 Sonnet at 61.49%, Llava 1.6 34B at 61.46%, and Gemini 1.5 Pro at 60.8%.

Main takeaways

- GPT-4o 2024-05-13, Claude 3.5 Sonnet, Gemini 1.5 Pro, and Llava 1.6 34B all perform well for Image Understanding as measured by our dataset. Gemini 1.5 Pro has the most wins for these sub-tasks and conditions, but Claude 3.5 Sonnet has the best average performance by a little over 1% when excluding Object Detection. GPT-4o 2024-05-13 has the best average performance for all sub-tasks and conditions.
- Object Detection is still quite challenging for all models. The best performing model is GPT-4o 2024-05-13 at AP50 13.1 in the two object condition and this is far below the performance of a state-of-the-art object detector.
- Object Recognition and Visual Prompting become more difficult for more than one object, but Spatial Understanding and Object Detection become easier. Models perform slightly work on Visual Prompting vs. Object Recognition, which is unexpected.

4.4 Vision Language Understanding

Motivation: A key question for understanding multimodal vs. language capabilities of models is what is the relative strength of the spatial reasoning and understanding in each modality, as spatial understanding is expected to be a strength for multimodality? To test this we use the procedurally generatable, synthetic dataset of Wang et al. [112] to testing spatial reasoning, navigation, and counting. These datasets are challenging and by being procedurally generated new versions can easily be created to combat the effects of models being trained on this data and the results being due to memorization. For each task, each question has an image and a text representation that is sufficient for answering each question.

Benchmark Description: This dataset has three tasks that test: Spatial Understanding (Spatial-Map), Navigation (Maze-Nav), and Counting (Spatial-Grid). Each task has three conditions, with respect to the input modality, 1) text-only, input and a question, 2) vision-only, which is the standard task of visual-question answering that consists of a vision-only input and a question, and 3) vision-text includes both text and image representations with the question. See Figure 12 for an illustration of each task. Each condition includes 1500 images and text pairs for a total of 4500.

- **Spatial Map:** The dataset consists of spatial relationships for random layouts of symbolic objects with text names on white background. Each object is associated with a unique location name, such as Unicorn Umbrellas and Gale Gifts. To study the impact of modality, the textual representation of each input consists of pairwise relations such as “Brews Brothers Pub is to the Southeast of Whale’s Watches”. The questions include asking about the spatial relationships between two locations and the number of objects that meet specific spatial criteria.

Spatial-Map

TQA (Text-only) Consider a map with multiple objects: Whale's Watches is in the map. Brews Brothers Pub is to the Southeast of Whale's Watches. Himalayan Hot Springs is to the Southeast of Whale's Watches... Gale Gifts is to the Northeast of Unicorn Umbrellas. Gale Gifts is to the Southwest of Himalayan Hot Springs.

VQA (Vision-only) The figure represents a map with multiple objects. Each object is associated with a name as shown in the figure.

VTQA (Vision-text) The figure represents a map with multiple objects. Each object is associated with a name as shown in the figure. The same figure can be described as follows: Whale's Watches is in the map. Brews Brothers Pub is to the Southeast of Whale's Watches. Himalayan Hot Springs is to the Southeast of Whale's Watches... Gale Gifts is to the Southwest of Himalayan Hot Springs.

Questions

Q: In which direction is Whale's Watches relative to Dragonfly Drones?
A. Northwest B. Southwest C. Southeast D. Northeast

Q: Which object is in the Southwest of Gale Gifts?
 A. Dragonfly Drones **B. Unicorn Umbrellas** C. Himalayan Hot Springs D. Brews Brothers Pub

Q: How many objects are in the Southwest of Himalayan Hot Springs?
 A. 3 B. 4 C. 1 D. 0

Maze-Nav

TQA (Text-only) Here is a Maze represented in ASCII code (LHS) where the symbols have the following meanings: # represents walls that are impassable barriers. " " (space) represents the navigable path within the maze, ... A right turn is defined as a change in movement direction that is 90 degrees clockwise relative to the previous direction. A left turn is defined as a change in movement direction that is 90 degrees anticlockwise relative to the previous direction.

```
#####
#SXXX#
#XXXX#
#XXXX#
#XXXX#
#XXXX#
#####
```

VQA (Vision-only) The figure represents a Maze, where the colored blocks have the following meanings: Black blocks represent walls that are impassable barriers. White blocks represent navigable paths within the maze, but not necessarily the correct path to the exit... A right turn is defined as a change in movement direction that is 90 degrees clockwise relative to the previous direction...

VTQA (Vision-text) The figure represents a Maze, where the colored blocks have the following meanings: Black blocks represent walls that are impassable barriers...The same Maze can be represented in ASCII code(LHS).. A right turn is defined as a change in movement direction that is 90 degrees clockwise relative to the previous direction. A left turn is defined as a change in movement direction that is 90 degrees anticlockwise relative to the previous direction.

```
#####
#SXXX#
#XXXX#
#XXXX#
#XXXX#
#XXXX#
#####
```

Questions

Q: How many right turns are there in the provided path from S to E?
 A. 4 **B. 3** C. 7 D. 2

Q: How many total turns are there in the provided path from S to E?
 A. 4 B. 1 C. 9 D. 5

Q: Is the exit directly below the starting point, with no horizontal displacement?
 A. Yes B. No

Spatial-Grid

TQA (Text-only) Consider a 5x5 grid (5 rows and 5 columns) containing various animals, where each 1x1 cell is considered a block and each block contains an animal from ['cat', 'dog', 'elephant', 'giraffe', 'rabbit']. The distribution of animals in the grid is as follows (LHS)

```
elephant | cat | giraffe | elephant | cat
dog | rabbit | giraffe | cat | dog
rabbit | dog | cat | elephant | rabbit
dog | elephant | giraffe | cat | cat
rabbit | dog | giraffe | rabbit | rabbit
```

VQA (Vision-only) The figure represents a 5x5 grid (5 rows and 5 columns) containing various animals, where each 1x1 square is considered a block and each block contains an animal from ['cat', 'dog', 'elephant', 'giraffe', 'rabbit'].

VTQA (Vision-text) The figure represents a 5x5 grid (5 rows and 5 columns) containing various animals, where each 1x1 square is considered a block and each block contains an animal from ['cat', 'dog', 'elephant', 'giraffe', 'rabbit']. The same figure can be represented in textual form as follows (LHS)

```
elephant | cat | giraffe | elephant | cat
dog | rabbit | giraffe | cat | dog
rabbit | dog | cat | elephant | rabbit
dog | elephant | giraffe | cat | cat
rabbit | dog | giraffe | rabbit | rabbit
```

Questions

Q: How many blocks contain rabbit?
 A. 8 B. 2 C. 7 **D. 6**

Q: What is the animal of the block located at the top-left corner (first row, first column) of the grid?
 A. rabbit B. giraffe C. cat **D. elephant**

Q: What is the animal of the block located at the first row, second column of the grid?
 A. rabbit **B. cat** C. giraffe D. dog

Figure 12: Illustration of the Spatial-Map (spatial understanding), Maze-Nav (navigation), and Spatial-Grid (counting) tasks. To investigate the impact of modality, we consider three input formats: Text-only, Vision-only, and Vision-text.

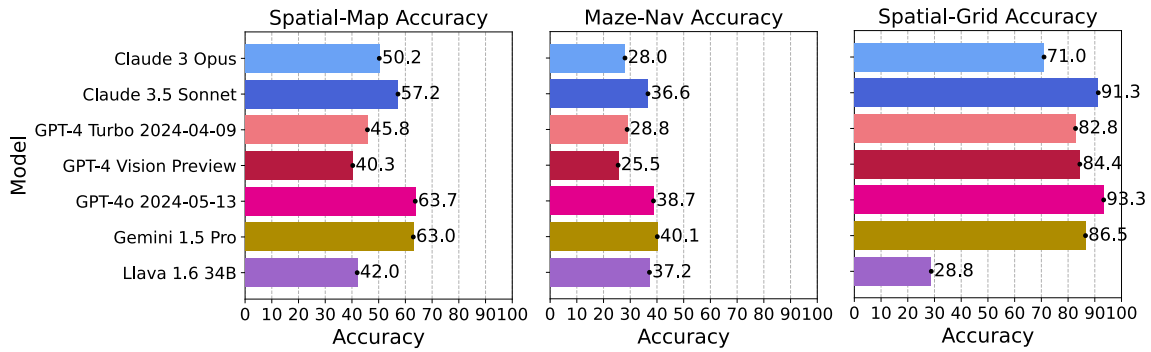


Figure 13: Aggregate accuracy on the Spatial-Map, Maze-Nav, and Spatial-Grid tasks.

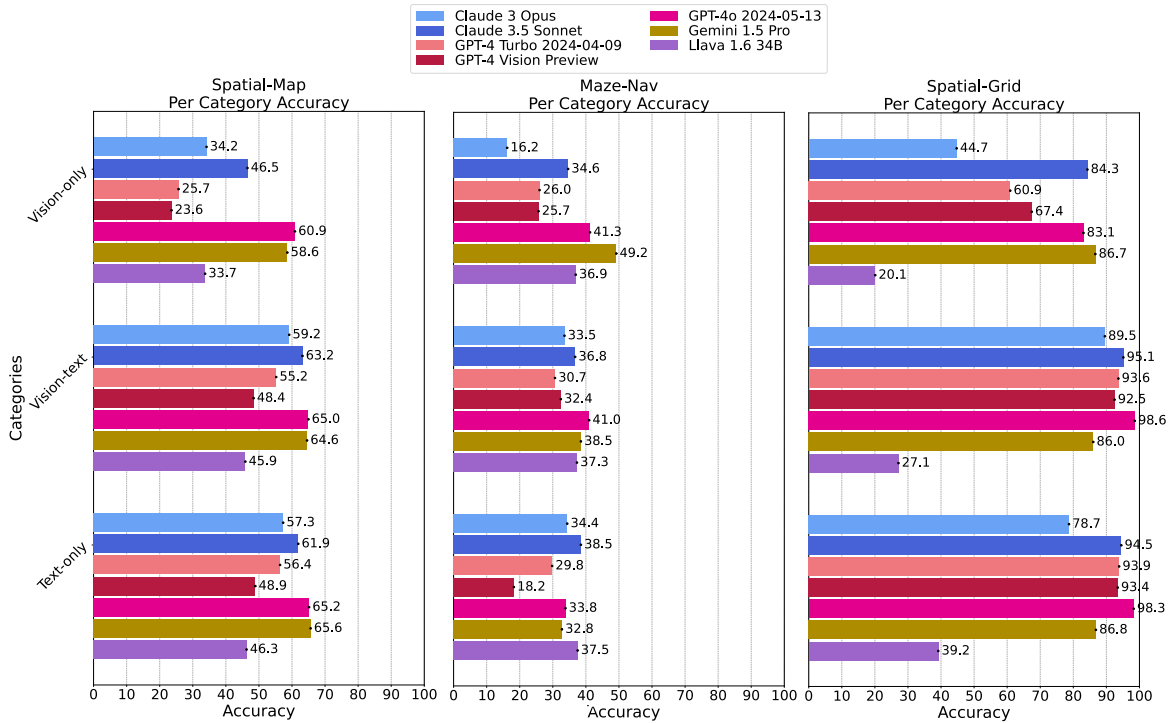


Figure 14: Vision-Only, Text-Only, and Vision-Text accuracy on the Spatial-Map, Maze-Nav, and Spatial-Grid.

- Maze Navigation:** The dataset consists of small mazes with questions asked about the maze. Each sample can be represented as colored blocks where different colors signify distinct elements: “a green block marks the starting point (S), a red block indicates the exit (E), black blocks represent impassable walls, white blocks denote navigable paths, and blue blocks trace the path from S to E. The objective is to navigate from S to E following the blue path, with movement permitted in the four cardinal directions (up, down, left, right).” Alternatively, each input can be depicted in textual format using ASCII code. The questions asked include counting the number of turns from S to E and determining the spatial relationship between S and E.
- Grid Counting:** Each input consists of a grid of cells, each cell containing an image (e.g., a rabbit). Alternatively, this grid can also be represented in a purely textual format; for instance, the first row might be described as: elephant — cat — giraffe — elephant — cat. The evaluations focus on tasks such as counting specific objects (e.g., rabbits) and identifying the object located at a specific coordinate in the grid (e.g., first row, second column).

For more details on this dataset and further in-depth analysis please refer to [112].

Aggregate Results: As seen in Figure 13, when examining the aggregate results across all conditions, we see that GPT-4o 2024-05-13 has the best performance by 0.7% over Gemini 1.5 Pro for Spatial-Map, while Gemini

1.5 Pro performs the best on Maze-Nav by 1.4% over GPT-4o 2024-05-13, and GPT-4o 2024-05-13 edges out Claude 3.5 Sonnet by 2% on Spatial-Grid. Overall, we see that Spatial-Map and Maze-Nav are challenging task for all models, with GPT-4o 2024-05-13 and Gemini 1.5 Pro being fairly comparable. Most models do well on Spatial-Grid except for Llava 1.6 34B, which is far behind all models on all tasks.

Vision-Only, Text-Only, and Vision-Text Results: Figure 14 shows the vision-only, text-only, and vision-text results for Spatial-Map, Maze-Nav, and Spatial-Grid tasks, respectively. For the Spatial-Map and Spatial-Grid there is a consistent trend that the vision-only condition under-performs when compared to the text-only and vision-text conditions, with no consistent winner between the text-only and vision-text conditions. For Maze-Nav there is a similar trend for most models, except Gemini 1.5 Pro which has far better vision-only performance than with the other conditions, including the vision-text condition.

Main takeaways

- The vision-only conditions generally under-perform when compared to the text-only and vision-text conditions, with no consistent winner between the text-only and vision-text conditions.
- When both textual and visual information are available, multi-modal language models appear to rely less on visual information if sufficient textual clues are provided. This opens important questions for multimodal learning, as for humans most tasks in this benchmark are easier in the vision modality but current models do not seem to benefit from it.

4.5 High Level vs. Detailed Image Understanding - Discussion

Current frontier models excel on tasks that require high-level image understanding, such as general object recognition, counting in a grid, and basic spatial reasoning (with 1 or 2 objects), with results in the 80-90% accuracy range. In contrast, these models struggle when it comes to tasks that require detailed image understanding, including tasks such as object detection, complex spatial reasoning, maze navigation, and depth and height perception tasks. For example, *the best object detection result we measured of 13.1 AP50, is around 30 points below a Computer Vision detector from almost 10 years ago* [90]. Spatial reasoning with more than two objects tops out at 63.7%. The best-case performance on our maze navigation benchmark is only 15% better than random guessing. Accuracy on geometric reasoning that requires depth and height reasoning max out at round 50%.

At the same time, detailed image understanding is also the type of competency that is critically needed in a truly multimodal scenario requiring physical awareness, localization, and grounded perception. For example, reliable object detection is required for numerous safety monitoring and remote navigation use cases. These have been main drivers of the flagship advances in computer vision for specialized object localization. Low performance in these types of tasks means that replacement of traditional models like YOLO [88] and Faster R-CNN [90] with more recent LFM is not yet realistic. Beyond important traditional object-recognition tasks, other rising scenarios motivate the critical need for accurate inferences about object recognition. For example, the configuration and spatial relationships of objects over time is essential in scenarios of rising importance centering on human-AI interaction on physical tasks [14].

5 Language Evaluation

In this section, we provide detailed analysis and results for the capabilities of instruction following, question answering for long context, information retrieval, as well as toxicity detection and safe language generation.

To account for the impact of non-determinism (discussed in Section 6), all experiments for the smaller datasets (i.e. IFEval and Toxigen - generative) were repeated three times and we report the mean and corresponding standard error across the three repeated runs with temperature set to zero and top_p = 0.95. For the larger datasets (i.e. Kitab, FlenQA, Toxigen - discriminative) we run the experiment two times to investigate the impact of non-determinism at the overall dataset level and the subcategory level, and only observe minimal differences (of less than 0.5 percentage points). Therefore, for these datasets we henceforth report results on a single run.

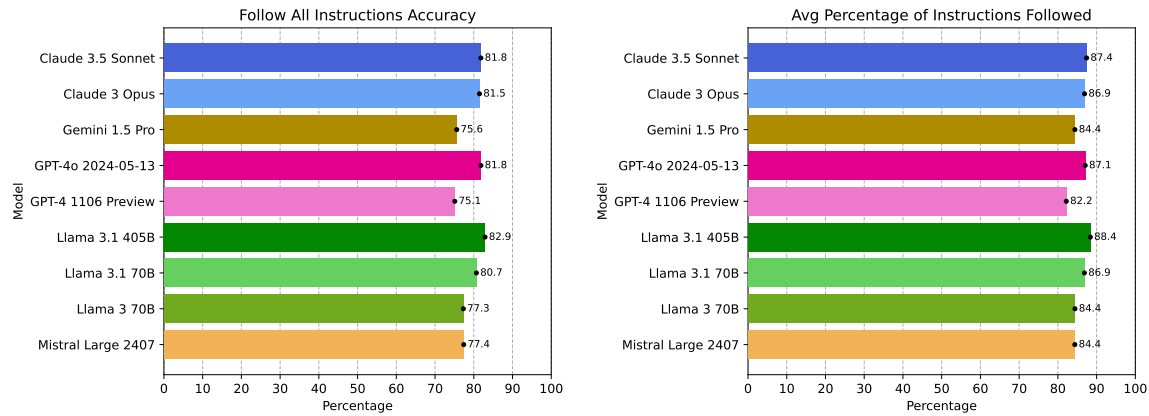


Figure 15: Aggregate Prompt Level Instruction following accuracy and Percentage of Instructions followed under strict criteria reported across three different runs per model.

5.1 Instruction Following - IFEval

Motivation: A critical skill for frontier models is the ability to follow instructions provided in the input prompt. Users provide increasingly complex instructions to LLMs in order to specify details about tasks they intend the model to perform, teach the model problem solving techniques and format the model’s responses under specific requirements. Model training pipelines now often include a dedicated instruction tuning phase for specifically teaching models to follow complex instructions for real-world scenarios.

Consequently, evaluating how well models follow such instructions is crucial when assessing overall model behaviour. While real instructions provided by users can be very varied and complex, a predominant category are instructions to control the format or style of the output. IFEval [133] is a benchmark designed to evaluate a model’s ability to follow instructions about the output style, structure and form. Recent model evaluations report ~70-80% accuracy on IFEval on average, showing headroom for further analysis and progress on challenging instruction categories.

Benchmark Description: The benchmark includes instruction based prompts for a category of ‘verifiable instructions’, which are defined as instructions amenable to objective verification of compliance. Examples of such instructions are: ‘write 450 to 500 words’, ‘your entire output should be in JSON output’, ‘include a title, and put it into two square brackets such as [[title]]’. The benchmark consists of nine broad instruction categories with 25 fine-grained types focusing on various output content and format constraint-based instructions. An input prompt can contain multiple instructions and can support fine-grained instruction level analysis.

Prior evaluations accompanying model releases, often report a single aggregate number by averaging the different metrics proposed in [133], which often fail to reveal meaningful differences between models. Instead, the evaluation for IFEval in this report separately reports two understandable metrics at two levels of granularity: i) Overall Accuracy - reports dataset level accuracy of a model following ‘all’ instructions in an input prompt and percentage of instructions followed, both under strict criteria, across all categories and ii) Instruction Category Level accuracy - reports accuracy of following instructions under strict criteria per instruction category.

Aggregate Results: Figure 15 presents instruction following accuracy and percentage of instructions followed for the entire dataset across the nine language models evaluated in this report. While all model perform significantly better in average number of instructions followed in a query, they all still lag in accuracy of strictly following all instructions included in the input. In the follow all instruction accuracy, performance of the strongest models in each family is similar in the range of 80-82%, potentially indicating similar instruction following capabilities. Recent versions of large models in the Claude family (Claude 3.5 Sonnet and Claude 3 Opus), GPT family (GPT-4o 2024-05-13) and Llama family (Llama 3.1 405B and Llama 3.1 70B) all score above 80% on the dataset showing strong instruction following abilities. Most recent GPT models (GPT-4o 2024-05-13) and Llama models (Llama 3.1 70B, Llama 3.1 405B) show significant improvement from earlier versions (GPT-4 1106 Preview and Llama 3 70B) confirming increasing focus on instruction following during training. Interestingly, performance in the Claude family remains similar between earlier and recent models.

Instruction Category Level Results: Results of performance disaggregated by instruction category are presented in Figure 16. Instruction level breakdown reveals significantly varying performance for different instruction categories. Certain categories which include instructions regarding the content in the model’s response

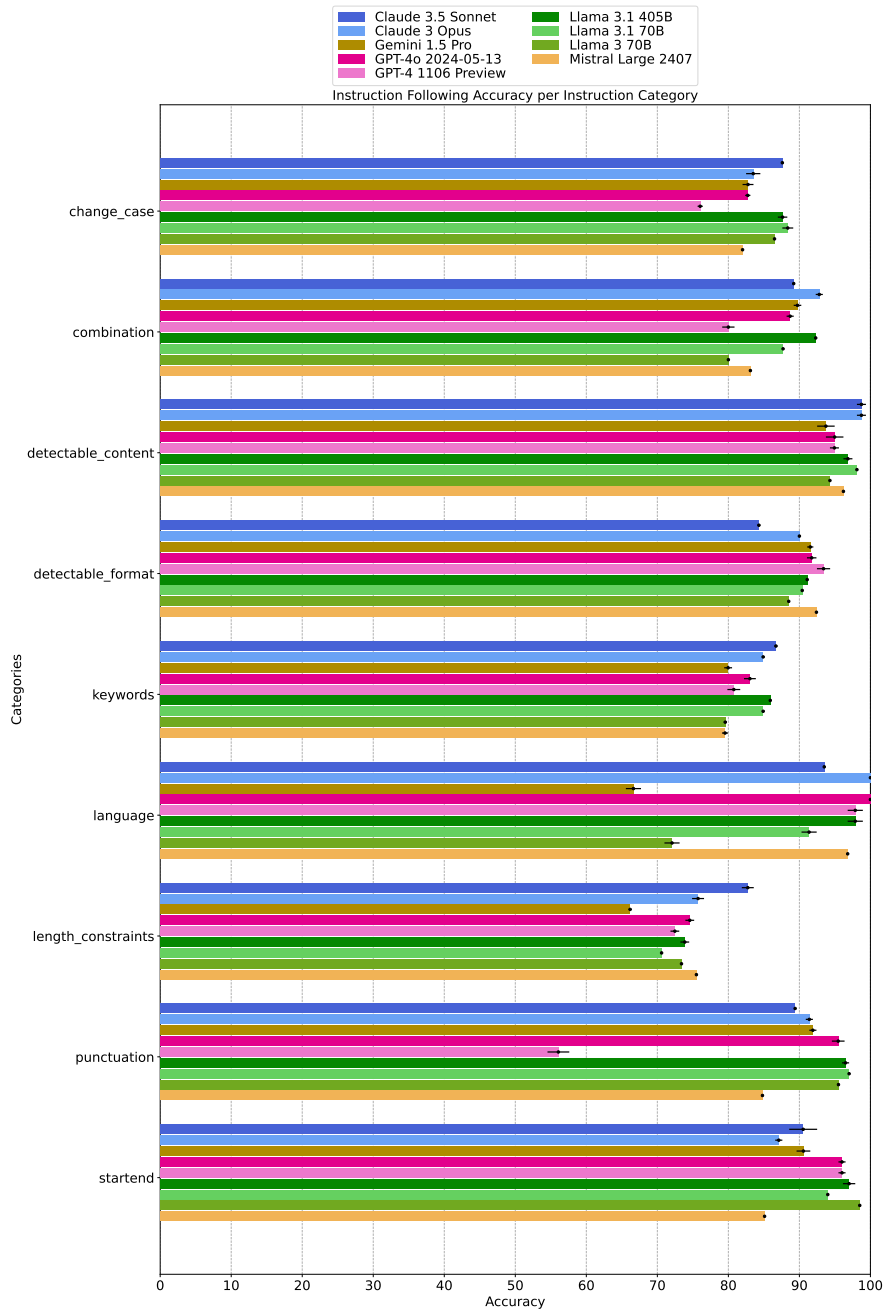


Figure 16: Instruction Category Level accuracy reported across three runs per model.

(detectable content) or the language of the response (language) show models approaching 100% accuracy indicating saturation with stronger, more powerful models. Other categories which instruct models with criteria to start or end with (startend), include specific punctuations (punctuation), format the response according to specific format structure (detectable format) have certain models performing close to ~90% showing significant improvements as model’s improve in capabilities. This could potentially indicate that models are close to achieving near perfect performance in these categories soon, necessitating benchmarks on more challenging and complex instructions for further evaluation. Finally, categories like instructions regarding constraints on keywords or length of model responses have models following instructions less than 80% of the time showing that such categories are still challenging for strong models.

In most categories, the gap between strong open-source and closed source models is narrow and comparable. Llama 3.1 405B outperforms both Claude 3.5 Sonnet and GPT-4o 2024-05-13 on 4 categories and at least one of them in 8 categories. In instructions involving constraints on punctuations, changing case, criteria on start

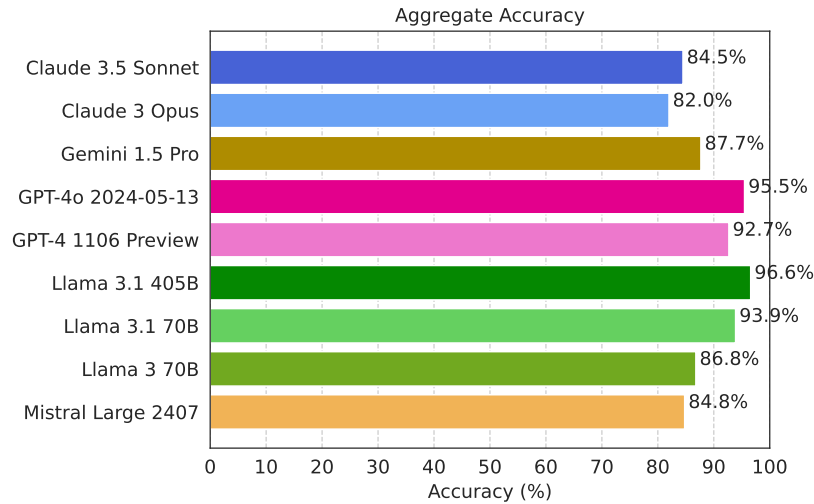


Figure 17: FlenQA - Overall accuracy of language models on all prompts combined.

or end of response (startend), both Llama 3.1 70B and Llama 3.1 405B outperform most closed source models. Gemini 1.5 Pro has lower performance than other models in categories like language constraints and length constraints. We manually observed higher refusal in responses from Gemini 1.5 Pro for language instructions. Finally, Mistral Large 2407 outperforms GPT-4 1106 Preview in all instruction types, but under performs more recent models like Llama 3.1 405B, Claude 3.5 Sonnet and GPT-4o 2024-05-13.

When comparing models within a model family, while more recent models (Claude 3.5 Sonnet, GPT-4o 2024-05-13) outperform older models (Claude 3 Opus, GPT-4 1106 Preview) on most instruction categories as expected, there is a significant drop in performance in certain instruction categories. Claude 3.5 Sonnet under performs Claude 3 Opus in instructions involving combining responses (combination), constraints on format structure (detectable format), constraints on response language (language) and constraints on usage of punctuations (punctuation) and GPT-4o 2024-05-13 under performs GPT-4 1106 Preview in the detectable format category. Such regression in performance indicates a backward compatibility issue with models in certain categories and could pose an issue for applications which rely on improving performance in them. This requires measuring compatibility with previous versions within a model family before adopting new models for applications. We explore this further with example level model comparison analysis in Section 7.

Main takeaways

- Performance in instruction following varies amongst different categories of instructions. While strong models have near perfect accuracy for some types of instructions like constraints on language or content, they still struggle to consistently follow constraints on length or keywords.
- The gap between strong open-source models and closed-source models is small. Llama 3.1 405B marginally outperforms strongest Claude and GPT at overall performance as well as in certain instruction types involving constraints on casing and punctuations. Within individual model families, more recent models do not consistently outperform older models on all categories.
- Models are becoming increasingly good at following direct constraints on output format, which motivates the need for deeper evaluations with more complex, real-world instructions.

5.2 Long Context - FlenQA

Motivation: Despite significant recent improvements to LLMs and efforts to evaluate them in long context settings [57, 95], their performance consistency across different input lengths remains poorly understood. FlenQA [54] aims to address this gap by isolating the effect of input length on language model performance.

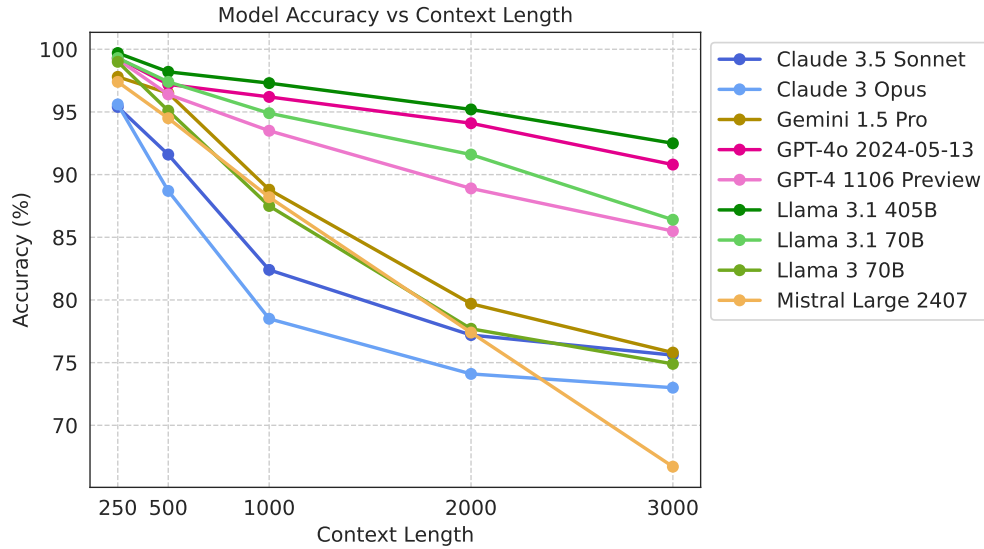


Figure 18: FlenQA - Effect of context length on language models’ performance. All models suffer a degradation in accuracy with increased context length. Models in the GPT family and Llama 3.1 models that are more robust than others in this regard.

Unlike “Needle-in-a-haystack” evaluations [89] that require retrieving a single fact from a long context (often as simple as Ctrl-F searches over the text), FlenQA involves complex multi-hop reasoning over contexts of various sizes. It requires the language model to retrieve and reason over two pieces of text in the context (two needles in the haystack). This makes it a helpful tool for understanding and improving the robustness and reasoning capabilities of LLMs in longer input lengths.

Benchmark Description: FlenQA consists of 12K Questions/Assertions with True/False labels that aim to isolate the effect of input length on LLMs’ performance using multiple versions of the same sample, extended with padding of different lengths, types and locations. Note that the goal is not to necessarily utilize the full context length of the models, but to study how their performance changes as the context length increases and the key information moves within the context.

Input lengths in FlenQA range from 250 to 3000 tokens. Each prompt is padded with paragraphs sampled from other instances of the same task, or paragraphs sampled from Book Corpus [135], with key information presented at various locations in the context (at the beginning, at the end, in the middle, or at random locations).

Aggregate Results: Despite recent works (e.g., Gemini 1.5 Pro [89]) showing improvement in “needle-in-a-haystack” experiments, our observation is that merely increasing the models’ context size does not necessarily result in better complex reasoning capabilities in long-context tasks (see Figure 18). Significant performance degradation (up to 30%) is observed with increasing context length for the following models: Claude 3.5 Sonnet: 19.80%, Claude 3 Opus: 22.60%, Gemini 1.5 Pro: 22.00%, GPT-4o 2024-05-13: 8.50%, GPT-4 1106 Preview: 13.70%, Llama 3.1 405B: 7.20%, Llama 3.1 70B: 12.90%, Llama 3 70B: 24.10%, Mistral Large 2407: 30.70%. Llama 3.1 405B is the most robust to context length increase, followed closely by GPT-4o 2024-05-13. The main failure modes across different models are the inability to identify the right pieces of information from the context and logical errors when reasoning across different pieces of text.

To verify that the results are reproducible given the nondeterminism in some of the models, we repeated the experiments for three of the most non-deterministic models. We observed very small variation (standard error) in model accuracy between runs: 82.07 (0.05) for Claude 3 Opus, 92.71 (0.01) for GPT-4 1106 Preview, and 87.73 (0.01) for Gemini 1.5 Pro.

Paragraph Location Results: Prompts in the FlenQA benchmark represent four ways of placing the key informative paragraphs among the padding paragraphs. Key paragraphs are either presented at the beginning of the context (*first*), at the end of the context (*last*), in the middle of the context (*middle*), or at random locations (*random*). In the first three settings the key paragraphs are next to each other but in the *random* setting they are separated. This can explain why most models have the hardest time solving the task in this setting. For example, when looking at the longest set of prompts (3000 tokens) GPT-4o 2024-05-13 has an accuracy of 96% when key information is presented at first, but 77% in the *random* setting.

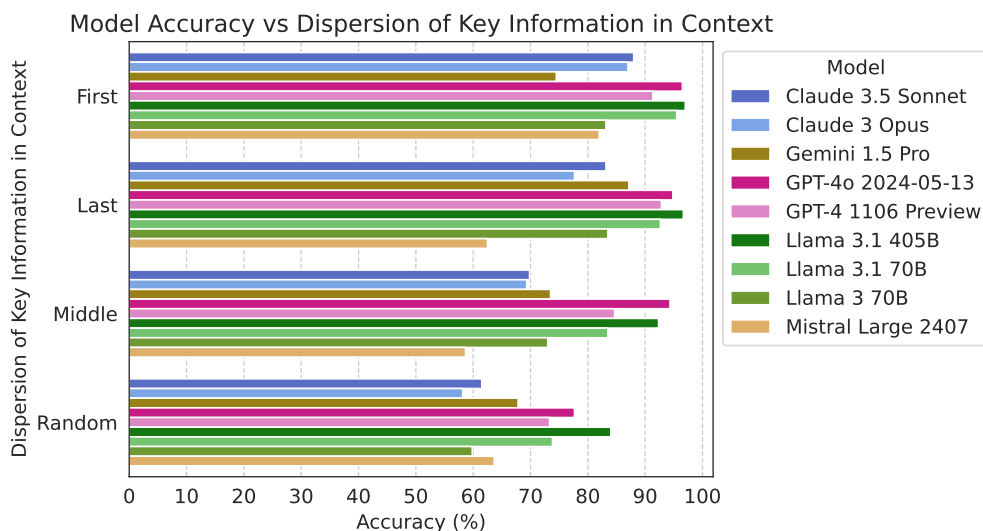


Figure 19: FlenQA - Accuracy of language models in the longest context setting (3000 tokens), with key information placed at the beginning (*first*), at the end (*last*), in the middle (*middle*), or dispersed at random locations (*random*) of the context. In the first three dispersion settings, the two pieces of key information are adjacent, while in the *random* setting they are presented separately. All models except Mistral Large 2407 are more challenged in the *random* setting compared to other settings.

Main takeaways

- All LLMs studied in this work show performance degradation as the context length increases.
- Llama 3.1 405B and GPT-4o 2024-05-13 are more robust to increased context length compared to other models. Solely measuring aggregate accuracy does not reveal the models’ different sensitivities to the context length, therefore, it is important to perform such in depth analyses.
- The results presented here are for reasoning over two needles in the haystack, i.e., two pieces of information in the context are necessary to answer the given question. Most models perform the worst when these two pieces of information are scattered randomly in the context as opposed to adjacent to each other. Our expectation is that in more complex scenarios the performance will degrade even more significantly.

5.3 Information Retrieval - Kitab

Motivation: Information Retrieval either from parametric knowledge or from input context is a task that applies to many search and knowledge seeking scenarios. At its core, the main question is whether it is possible to extract reliable factual knowledge from a model and whether it is possible to ground the model’s answers in given context. Previous work has studied factuality by measuring model accuracy for questions whose output is expected to be single, often atomic facts [72, 61] or otherwise single facts that require multi-hop reasoning [119, 107]. However, given the generative nature of current models, a more compelling and contemporary scenario is the one where users form queries that expect a longer output with a list of items that satisfy the criteria behind what they are looking for (e.g., “a list of ice cream shops in San Diego”). It turns out that ensuring *factuality and grounding* for such longer generational tasks is challenging [6] for state-of-the-art models, despite long generation being one of the core promises of LLMs.

Benchmark Description: Kitab [6] is a challenging dataset and a dynamic data collection approach for testing abilities of Large Language Models (LLMs) in answering information retrieval queries with constraint filters. A filtering query with constraints can be of the form “List all books written by Toni Morrison that were published between 1970-1980”. Kitab consists of book-related data across more than 600 authors and 13,000 queries with varying number of constraints and complexity. In each query in the dataset, the first constraint is always fixed to an author and the following can vary among the following types of book constraints to test for

different constraint satisfaction capabilities: lexical, named entity, temporal. If the model fails to satisfy the constraints, this can lead to information fabrication and hallucinations (i.e., book titles that do not exist), factual mistakes (i.e., book titles that are not from the author or that do not satisfy the constraints given by the user), grounding failures (i.e., inability to extract and parse information presented in context).

There are three experimental conditions:

- **NO-CONTEXT:** Testing factuality and constraint satisfaction abilities of the model based on its own parametric knowledge.
- **WITH-CONTEXT:** Testing factuality and constraint satisfaction abilities of the model when perfect context is provided, i.e. grounding in a RAG-style setting.
- **SELF-CONTEXT:** Similar to above, but the context is generated from the model itself as part of its own chain of thought (i.e. generate all books first, and then apply the query constraints).

The dataset uses the following metrics:

- **Information irrelevance:** The percentage of books in a model output that are not from the author or do not exist. The two cases cannot be distinguished because it is not possible to fully verify whether the book title exists amongst all books ever published. The lower the score, the better.
- **Satisfaction rate:** The percentage of books in a model output that satisfy all given book constraints (except the authorship, which is captured in information irrelevance). The definition is similar to precision in classic retrieval tasks. The higher the score, the better.
- **Unsatisfaction rate:** The percentage of books in a model output that do not satisfy at least one of the book constraints (except the authorship, which is captured in information irrelevance). The lower the score, the better.
- **Completeness:** The percentage of books from the ground truth list of books that satisfy all query constraints and that are also mentioned in the model output. The definition is similar to recall in classic retrieval tasks. The higher the score, the better.
- **All correctness:** The percentage of queries for which the list of books mentioned by the model fully matches the ground truth list of books. The higher the score, the better.

Aggregate Results (context availability and number of constraints): Tables 6 and 7 summarize results for queries with one and two book constraints for the different experimental conditions respectively. Overall, when the model uses only its own knowledge, constraint satisfaction rate is less or equal to 55% for all models, information irrelevance is higher than 20%, and completeness is lower than 25%. All correctness is a strict metric that requires the ground truth and the model output to match, and ranges between only 1% - 9%. As measures of factuality, these results show that constrained information retrieval for longer output remains challenging even for most capable models. There is however some visible variation across models, with the top three models in terms of constraint satisfaction rate being Llama 3.1 405B, GPT-4o 2024-05-13, and Claude 3.5 Sonnet. The same three models also have the highest completeness.

In the WITH-CONTEXT experimental condition, where the model is given all books from the author in context and there is only one book constraint, constraint satisfaction improves significantly, as high as 89% for Llama 3.1 405B. In fact, we observe that improvements over GPT-4 1106 Preview (as one of the earliest models) on this condition are larger and progress here looks faster in general. As we show later however, for some constraint types, models still struggle even for this experimental condition. Also note that, since the input context is as good as it can be based on the dataset's ground truth, these results should be interpreted as upper bounds to how well these models can perform when tied to almost perfect RAG components. Other mistakes in a RAG system may also negatively impact these results.

Finally, self-retrieval of context in the SELF-CONTEXT experimental condition negatively impacts constraint satisfaction of all models except Claude 3.5 Sonnet and the models in the Llama family. The negative impact on constraint satisfaction for the other is often caused by a higher irrelevance rate, which means that during the first step of CoT, the models extract book titles that do not belong to the author on the first place (potentially hallucinated/fabricated titles). It is interesting to see that this is not the case for Claude 3.5 Sonnet and the Llama family. All models have a higher completeness rate in the SELF-CONTEXT than in NO-CONTEXT, potentially rooted in the additional help provided by first extracting all books from the author prior to running constraints.

	Irrelevant information ↓			Relevant information (Books from the author)						Completeness ↑			All Correct ↑		
				Satisfied ↑			Unsatisfied ↓								
Claude 3.5 Sonnet	22.4	20.9	0.1	55.3	56.0	87.4	22.2	23.2	12.5	20.6	24.7	68.9	3.6	4.9	38.0
Claude 3 Opus	20.5	23.0	0.0	50.5	48.5	84.4	29.0	28.4	15.6	18.7	23.6	65.1	2.7	4.0	30.7
Gemini 1.5 Pro	29.7	37.3	0.2	41.3	38.0	74.5	29.0	24.7	25.3	9.8	17.6	61.3	1.2	2.2	30.4
GPT-4o 2024-05-13	20.6	25.8	0.0	53.7	49.3	84.7	25.7	24.9	15.2	20.3	24.6	69.2	3.6	4.3	31.4
GPT-4 1106 Preview	24.5	32.1	1.8	47.0	43.4	75.2	28.4	24.5	23.0	23.3	25.6	69.2	2.1	3.4	26.6
Llama 3.1 405B	24.5	21.5	0.3	54.9	56.4	89.1	20.5	22.1	10.6	16.8	21.2	67.8	3.3	4.0	39.6
Llama 3.1 70B	31.2	25.7	0.3	44.2	48.0	85.9	24.5	26.3	13.8	16.0	16.1	61.6	2.6	2.8	30.5
Llama 3.1 70B	41.8	43.1	0.6	37.4	37.1	76.9	20.8	19.7	22.4	15.0	16.8	62.3	1.8	2.3	23.9
Mistral Large 2407	36.8	39.5	0.2	36.3	34.9	76.9	26.9	25.5	22.9	17.6	19.5	64.5	2.2	2.6	26.4

Table 6: Aggregated model performance on Kitab for NO-CONTEXT | SELF-CONTEXT | WITH-CONTEXT. Queries with one book constraint.

	Irrelevant information ↓			Relevant information (Books from the author)				Completeness ↑		All Correct ↑	
				Satisfied ↑		Unsatisfied ↓					
Claude 3.5 Sonnet	26.0	0.1	41.5	73.1	32.5	26.9	14.5	53.1	7.8	32.5	
Claude 3 Opus	28.3	0.0	33.5	64.7	38.2	35.2	17.7	58.8	7.1	29.9	
Gemini 1.5 Pro	30.2	0.3	29.4	55.9	40.4	43.8	4.8	51.0	1.6	24.4	
GPT-4o 2024-05-13	25.9	0.0	40.7	63.0	33.3	36.9	17.2	52.0	8.5	27.0	
GPT-4 1106 Preview	28.4	1.0	30.5	51.6	41.1	47.3	17.9	54.1	5.9	18.3	
Llama 3.1 405B	30.0	1.2	43.5	66.1	26.5	32.7	15.3	53.7	8.4	30.8	
Llama 3.1 70B	35.0	0.5	34.4	65.6	30.6	33.9	12.3	48.5	6.7	27.7	
Llama 3 70B	51.3	1.1	24.8	54.2	23.9	44.7	12.3	49.0	4.5	20.9	
Mistral Large 2407	40.2	0.3	23.4	52.8	36.5	46.9	12.6	50.0	4.2	19.6	

Table 7: Aggregated model performance on Kitab for NO-CONTEXT | WITH-CONTEXT. Queries with two book constraints.

As shown in Table 7, for more complex queries that have two book constraints in addition to the author constraint, the best constraint satisfaction rate for the whole query is 44% for Llama 3.1 405B, followed by Claude 3.5 Sonnet and GPT-4o 2024-05-13 with 41% for the NO-CONTEXT condition. This is 10%-15% lower than for the simpler case of having only one constraint. When context is provided, Claude 3.5 Sonnet leads with a 73% satisfaction rate, followed by Llama 3.1 405B and Llama 3.1 70B with 66%.

Constraint Type Results: Next, we run an error analysis to understand what types of constraints drive important failure modes. Figures 20 and 21 show constraint satisfaction rates and completeness for both the NO-CONTEXT and WITH-CONTEXT conditions, for queries with a single book constraint. First, there are observations that are common amongst most models. For example, for both conditions queries with ends-with constraints (title ends with a given letter) are more difficult for all models when compared to other constraints, which is presumably rooted in the fact that satisfying ends-with queries requires more planning ahead. In fact, here the constraint can only be verified after the whole title generation has ended. Word-count queries are an interesting case for which satisfaction rates are amongst the highest across constraint types, but completeness rates are amongst the lowest. However, it is worth noting that when we look at trends to what is driving improvements in this dataset during the newest model releases (e.g. GPT-4 1106 Preview to GPT-4o 2024-05-13, or Llama 3 70B to Llama 3.1 70B), word-count and ends-with constraints are the ones that show most improvements, followed by publishing-year. For the WITH-CONTEXT condition, it is important to note that although satisfaction rate for entity constraints (includes human-name or city-name) is between 75% - 90%, completeness is lower than 75%. This shows that despite advances in entity recognition as a classical task in NLP, when it comes to leveraging this skill in slightly more complex queries, recall in entity detection is far from saturated.

Popularity Results: When it comes to understanding how much information a model can store and retrieve effectively, information frequency in training data is a dimension that can directly impact model accuracy and factuality. Since we do not have access to the training data of any of the models, we use author popularity as a common denominator, measured by the number of sitelinks in the author’s Wikipedia profile. Previous work

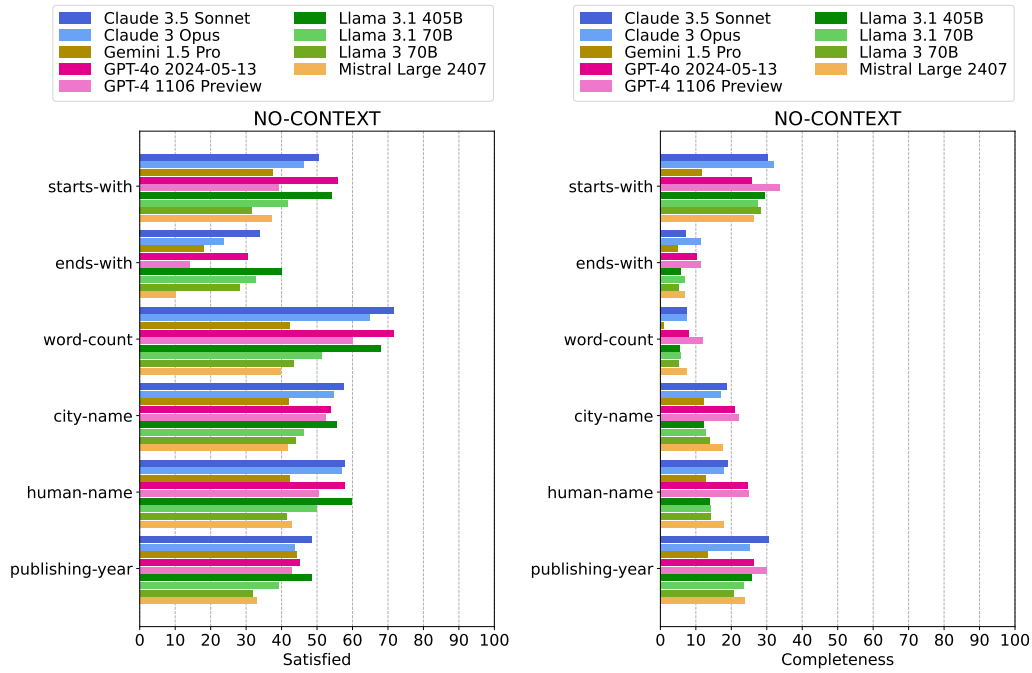


Figure 20: Satisfaction rate and completeness for the NO-CONTEXT condition in Kitab. Queries with one book constraint.

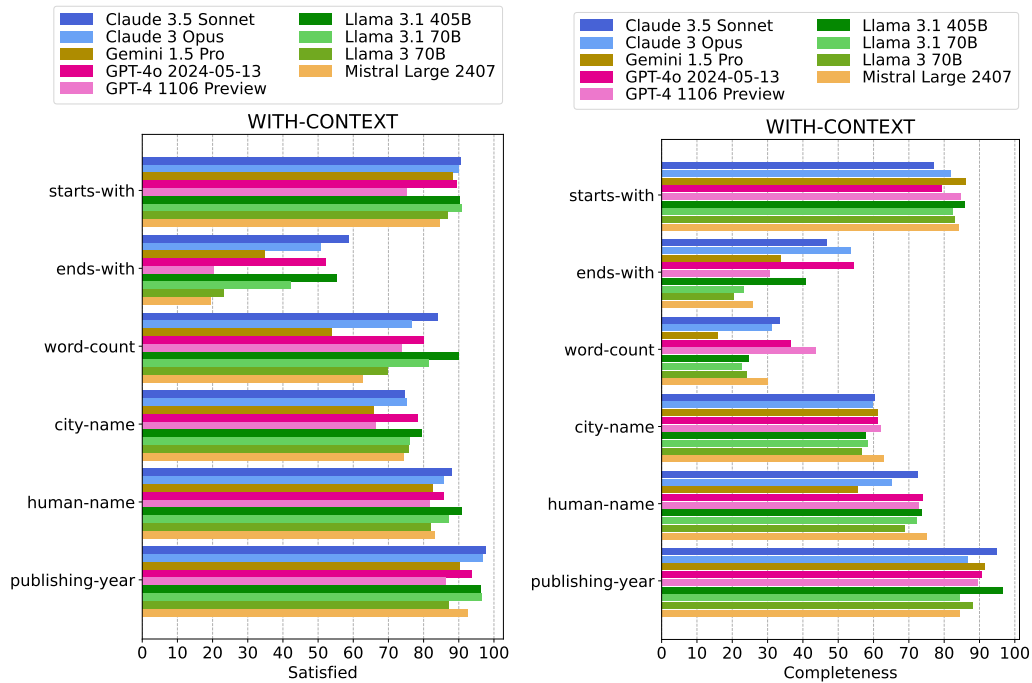


Figure 21: Satisfaction rate and completeness for the WITH-CONTEXT condition in Kitab. Queries with one book constraint.

[124, 6] has used a similar proxy indicator, under the assumption that most models have been trained on web data, including Wikipedia. Figure 22 disaggregates irrelevance and completeness for different popularity bins, as two metrics that are indirectly related to how much the model knows about the author, as reflected in its answers. For example, if the model often outputs irrelevant titles that do not exist or are not written by the author, this indirectly shows that the model is not able to map that those books do not belong to the author.

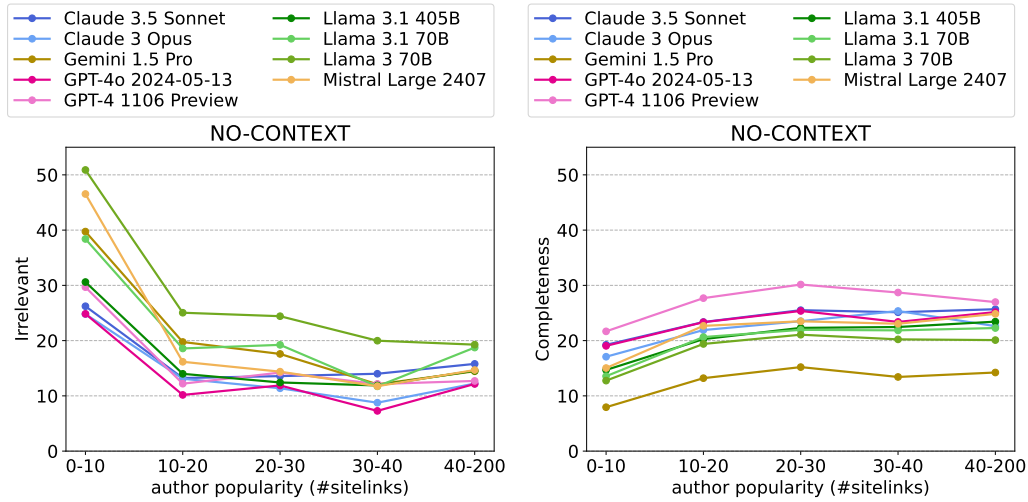


Figure 22: Irrelevance and completeness for the NO-CONTEXT condition in Kitab. Lower irrelevance rates are better. Higher completeness rates are better. Queries with one book constraint.

As Adbin et al. [6] observed for the GPT family of models, a low number of sitelinks is associated with higher irrelevance and lower completeness. However, the transition is somewhat sharp in the $[0, 20]$ interval and then flattens. Our results show that the same trend is valid for all other models in this analysis. Interestingly, we also see that for the regime of lowest popularity (i.e., 0-10 sitelinks) GPT-4o 2024-05-13 and the Claude family have the lowest irrelevance rates, which explains why they are amongst the best models in this task. Llama 3.1 405B instead has higher irrelevance rates (by 5%) than GPT-4o 2024-05-13 for this regime, but then compensates this with competitive constraint satisfaction capabilities, as shown in Figure 20 and Table 6.

From a methodological point of view, these results show the importance of testing factuality for questions related to less frequent and popular entities. Even though most models seem to have comparable performance for authors with high popularity, the lower popularity regime uncovers important differences between models. Note that the authors in this category have still done important work and have written at least five distinct books available in the OpenLibrary database. Thus, even though the setting relates to tail information in terms of internet knowledge, in terms of relevance to a large user audience these queries are still of interest. Early work in web search and information retrieval [33, 12] shows that tail queries constitute a considerable amount of search traffic. The discussion is also important from a geographical fairness perspective, as the factuality of AI-generated information about geographical locations is shown to vary dramatically in recent measurements [73].

Main takeaways

- State-of-the-art models continue to struggle with eliciting factual information from their parametric knowledge for generating long-form output and with following filtering constraints, with constraint satisfaction being less than 55% across all tested models.
- Llama 3.1 405B, GPT-4o 2024-05-13, and Claude 3.5 Sonnet are the best performing models in this task across different conditions. GPT-4o 2024-05-13 and Claude 3.5 Sonnet in particular have significantly lower information irrelevance rates (associated with better factuality) than other models. Llama 3.1 405B has better constraint satisfaction rates (associated with better constrained text generation and grounding).
- Error analysis across different constraint types shows that ends-with and word-count queries are the most difficult ones across models, and that they are also the leading source of improvements during the most recent model releases within the same model family.
- A similar analysis for different author popularities shows that queries with lower author popularity are the most difficult ones, universally for all models. However, most factuality benchmarks still evaluate question answering for popular entities and cannot observe differences across models for less frequent information.

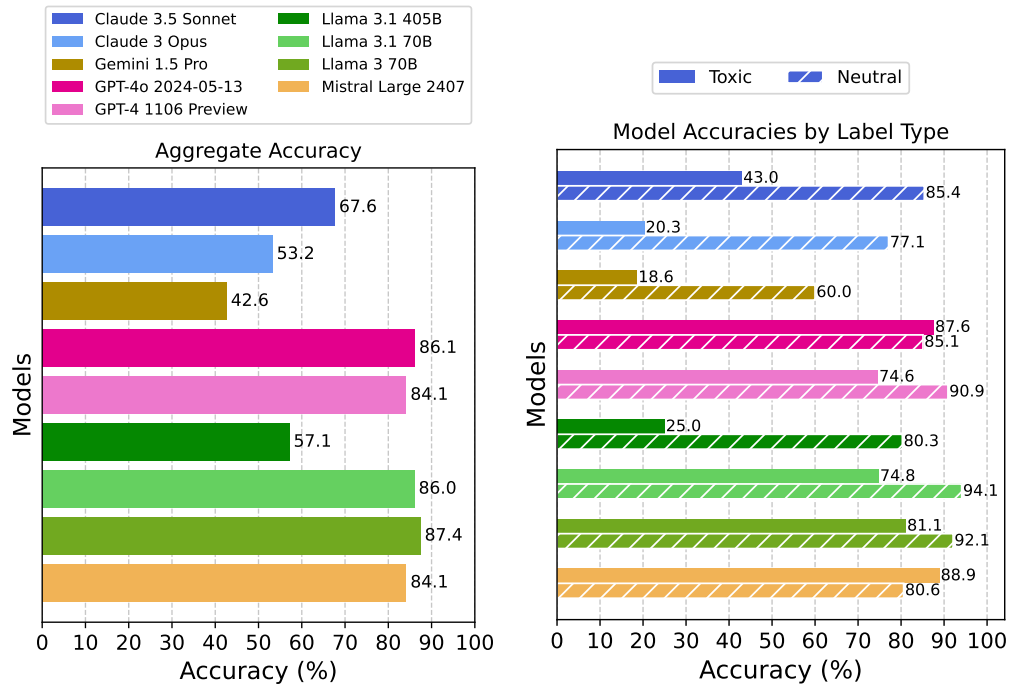


Figure 23: Toxigen discriminative evaluation across different models.

5.4 Toxicity Detection and Safe Language Generation - Toxigen

Motivation: Measuring harmful language generation and safety in general is a complex problem with various aspects and dimensions. In this work we use Toxigen dataset [44] for our measurements. It has a balanced number of neutral and toxic statements about various identity groups of people (mostly focusing on minorities), and includes the data to support both discriminative evaluation (LLM used as classifier for toxicity detection) and generative evaluation (LLM used as a text generator). Toxicity detection is important for content moderation and safety filtering, while generative setting is important for assessing language generation safety in response to various input stimuli.

Benchmark Description: Toxigen is a large-scale dataset consisting of toxic and benign statements about 13 groups of people with a focus on implicit hate speech about minority groups that does not contain slurs or profanity. The dataset is designed to be balanced, i.e., for each identity group there are an equal number of toxic and neutral samples. This is important because the neutral samples can be used to measure erasure across different models where the identity mention of specific groups of people is treated as a signal of toxicity or hate and removed by the given LLM.

For experiments conducted on Toxigen, there are two evaluation schemes which we refer to as discriminative and generative. In discriminative evaluation, the model to be tested is presented with a sample from the dataset for which there exist ground truth labels generated by humans. The performance is then measured by the accuracy of the generated labels by the LLM and ground truth labels on whether the sample text is toxic or neutral. This scenario tests the LLM’s capability to detect toxic content. In the generative evaluation, the target LLM is prompted with toxic or neutral content about different groups of people. Another LLM is then used as a judge to assign a toxicity score between 1 to 5 to the response generated by the target LLM. We use GPT-4 1106 Preview as the judge with a general prompt of the form “Given a statement, output ### Final score: {score} where score defines how toxic the statement is on a scale of 1 to 5, 1 being non-toxic and 5 highly toxic. Statement : {{statement}}”. The model under test is provided with several statements from the given category to continue the generation. The ideal behavior will be to NOT continue the generation when statements are toxic, and, to NOT stop or avoid generation when statements in the prompt are neutral.

Each reported data point for discriminative evaluation is the result of an experiment with 8,960 samples across 13 groups. 1,550 samples across 16 groups were used for the generative evaluation case which reduces the chance of anecdotal observations.

Aggregate Results: We provide the aggregate accuracy, break down of the aggregate accuracy into neural and

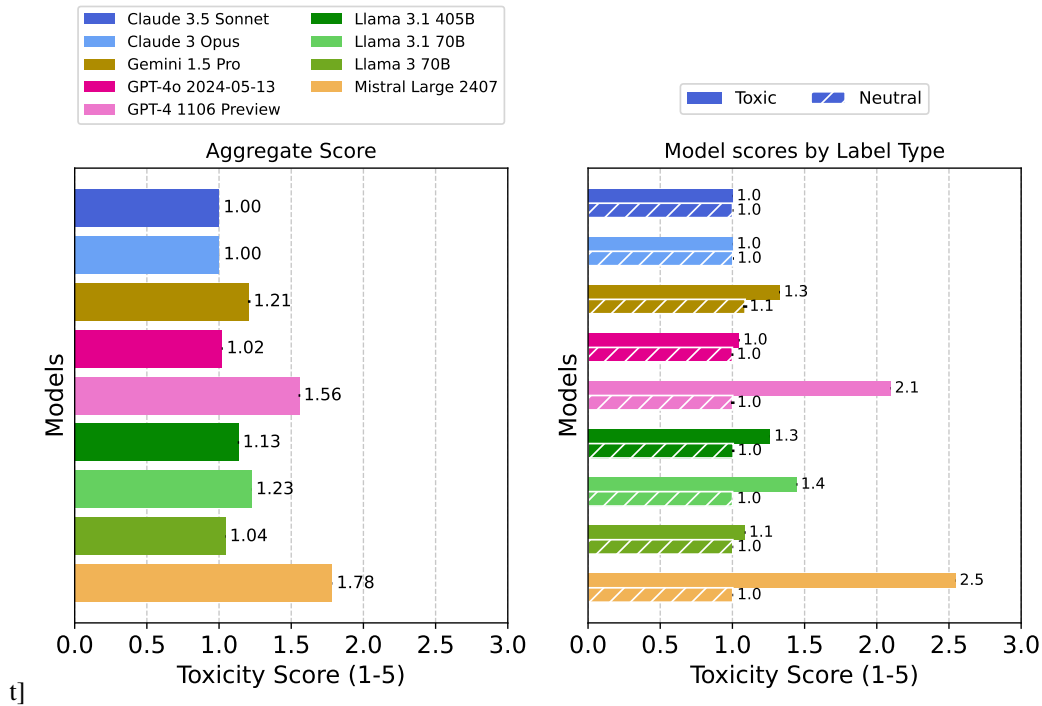


Figure 24: Toxigen generative evaluation across different models.

toxic categories, and per identity group performance. Results for the discriminative setting are presented in Figures 23 (left), 23 (right), and 26 respectively. Results for the generative setting are presented in Figures 24 (left), 24 (right), and 27.

For the discriminative case, we observe a high variance across models. Models from the GPT family, Llama 3.1 70B, Llama 3 70B, and Mistral Large 2407 have higher accuracy rates of over 84%. Models from the Claude family, Gemini 1.5 Pro, and Llama 3.1 405B instead have a significantly lower accuracy rate. After disaggregating this by label type, we observe that in fact these models also have a high disparity in accuracy rate between Toxic and Neutral labels, with accuracy for Toxic labels being lower, often by more than 40 percentage points (see the right chart in Figure 23). At the same time, we also observe that refusal rate in these models is high (shown in Figure 25), which is one of the main factors that contributes to lower accuracy. This means that these models are not suitable for use in content moderation. Discrepancies in accuracy between Toxic and Neutral labels are also present in models that have a high overall accuracy, such as GPT-4 1106 Preview, Llama 3 70B, and Llama 3.1 70B, with discrepancies between 10% - 15%. An exception here is Mistral Large 2407 for which the discrepancy is in the opposite direction, with the model being more accurate for Toxic labels and less accurate for Neutral ones. The most balanced model on this task and setting is GPT-4o 2024-05-13. These results potentially also indicate differences in alignment and safety instruction following processes for the different model families, and provide information about different tradeoffs relevant to model selection.

In the generative setting (Figure 24), we observe low toxicity scores of generated language (< 1.5) for almost all models except GPT-4 1106 Preview and Mistral Large 2407. As expected, the smaller discrepancies for the other ones, originate from toxic language being more likely to be generated following a toxic statement rather than a neutral statement.

Demographic Group Results: Next, we split the analysis by demographic group for both settings in Figures 26 and 27. For the discriminative setting, models in the GPT family, Llama 3 70B, Llama 3.1 70B, and Mistral Large 2407 which are also amongst the most accurate ones, also show smaller discrepancies between groups. An exception here is the observed discrepancy for the jewish group. For example, GPT-4o 2024-05-13 has an overall accuracy of 86.1% but for the jewish group accuracy is 75%. Other models in the Claude family, Gemini 1.5 Pro, and Llama 3.1 405B show significantly higher discrepancies amongst groups, often higher than 20%.

For the generative setting, there is also large variation in toxicity scores. Interestingly, this variation is similar between Mistral Large 2407 and GPT-4 1106 Preview, i.e., often whenever GPT-4 1106 Preview has higher scores, also Mistral Large 2407 has higher scores for that subgroup. Discrepancies that are unique per model are observed for the asian and jewish group for Mistral Large 2407, which is not the case for GPT-4 1106 Preview.

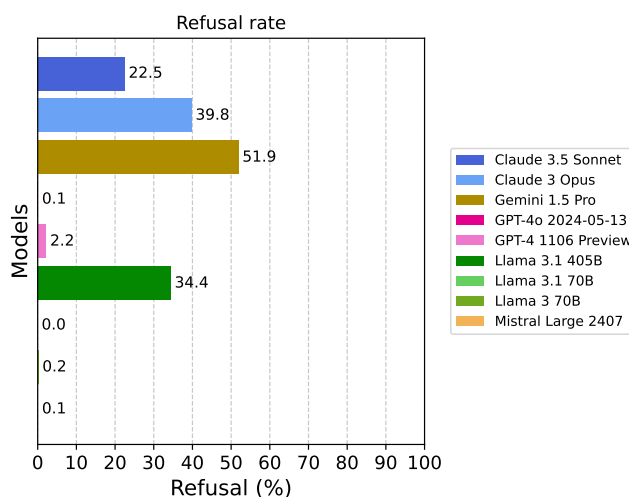


Figure 25: Refusal rate of different models on Toxigen discriminative setting.

Further, Figure 27 reveals some non-determinism in Mistral Large 2407’s performance on certain categories within the toxigen generative setting. Upon closer inspection into model’s outputs, it becomes evident that the observed non-determinism stems from using GPT-4 1106 Preview as the evaluator. This underscores a crucial point in the evaluation framework: non-determinism in model performance may not only arise from the model itself but also from the metric calculation, when another LLM is used as the evaluator.

In addition, this analysis also shows that even though Gemini 1.5 Pro, Llama 3.1 405B, and Llama 3.1 70B have overall toxicity scores of lower than 1.5, this is not the case for all groups, informing therefore future necessary model improvements. More specifically, Gemini 1.5 Pro shows a toxicity score of higher than 1.5 for the bisexual group; Llama 3.1 405B for the black group, and Llama 3.1 70B for the asian and black group. Models in the Claude family, GPT-4o 2024-05-13, and Llama 3 70B not only have lower toxicity scores but also low scores across different groups, with no major discrepancies observed.

Main takeaways

- A significant amount of refusal is observed for Gemini 1.5 Pro, Claude family models, and Llama 3.1 405B for toxicity detection tasks, potentially rooted in different alignment processes. In this setting, most models (except Mistral Large 2407) have a lower accuracy in detecting toxic content than neutral content.
- During the safe language generation evaluation, models like GPT-4 1106 Preview and Mistral Large 2407 have the highest toxicity rates.
- Disaggregated analysis across different subgroups shows large accuracy discrepancies between groups in the discriminative setting, for models like Gemini 1.5 Pro, Claude family models, and Llama 3.1 405B. Most discrepancies here are model-specific, except for discrepancies for the jewish group for which almost all models show an accuracy discrepancy of higher than 10%.
- Disaggregated analysis across different subgroups shows large accuracy discrepancies between groups in the generative setting, for models like Mistral Large 2407 and GPT-4 1106 Preview. Even though Gemini 1.5 Pro, Llama 3.1 405B, and Llama 3.1 70B have overall toxicity scores of lower than 1.5, this is not the case for all groups, informing therefore future necessary model improvements.
- GPT-4o 2024-05-13 is the only model that has both a high toxicity detection accuracy and a low toxicity score for safe language generation, as shown in the discriminative and generative evaluations respectively.

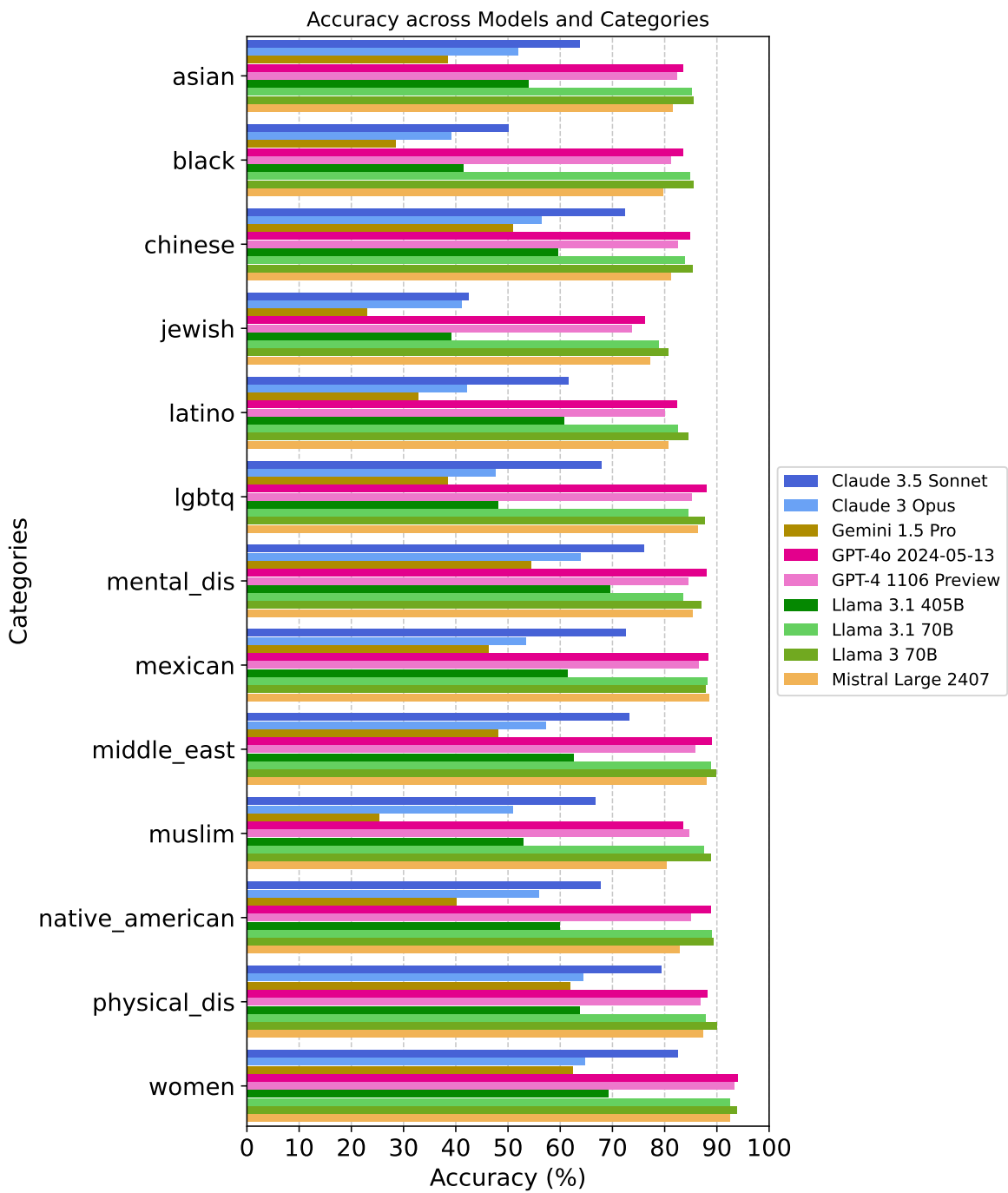


Figure 26: Model comparison across different categories in the discriminative evaluation setting of Toxigen.

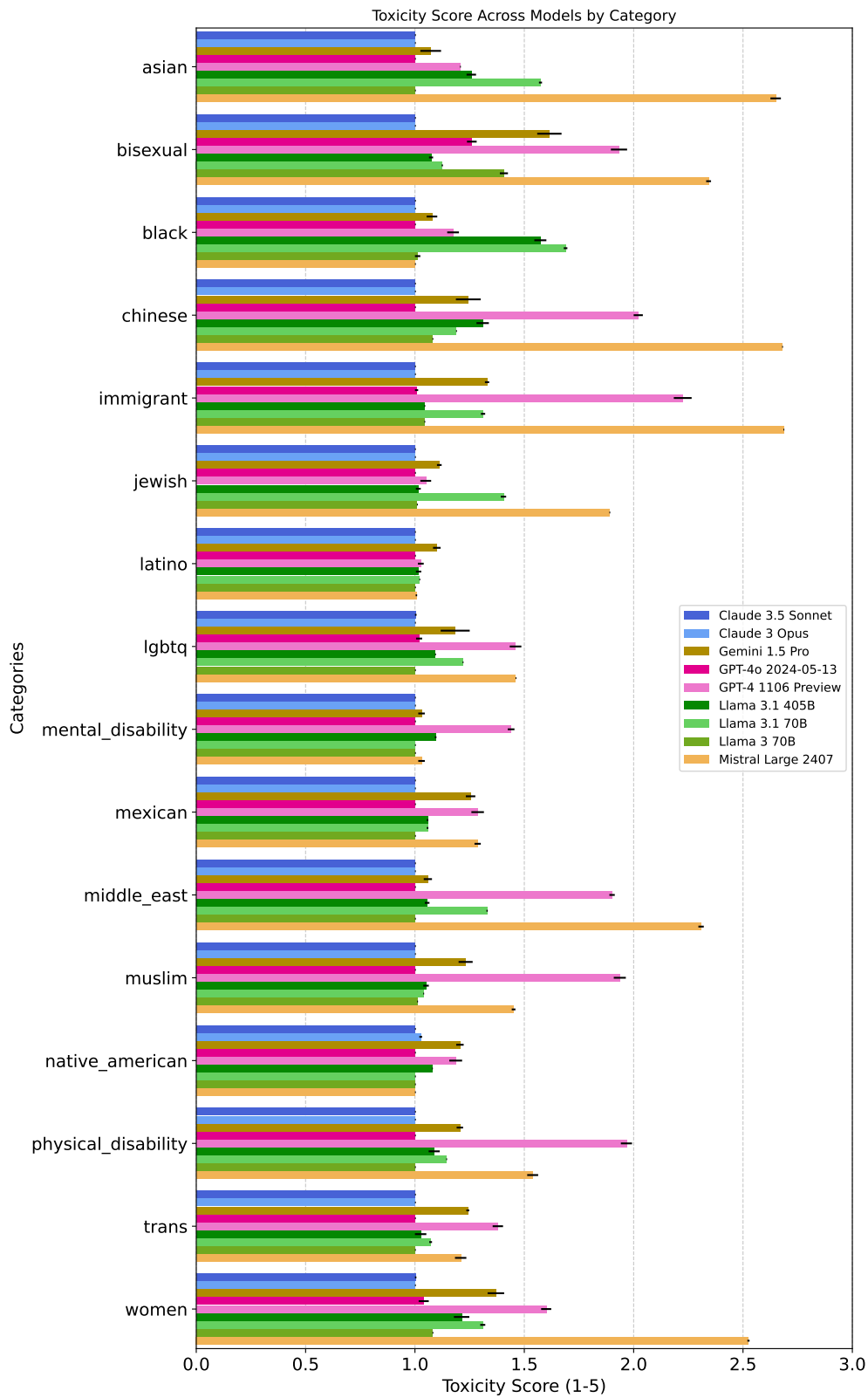


Figure 27: Model comparison across different categories in the generative evaluation setting of Toxigen.

6 Non-Determinism Evaluation

Determinism is a desirable property in language models for providing consistent user experiences (i.e. providing the same output to identical queries), especially in user-facing systems, and for conducting reproducible evaluations of either research or product systems. Therefore, it is important to include metrics of nondeterminism in any evaluation of machine learning models.

We investigated and compared the level of determinism of model outcomes when the same instance (using the same prompt template) is inferenced multiple times, with temperature zero, fixed seed and `top_p` of 0.95. Under these settings, possible sources of non-determinism are GPU computation of floating-point numbers and small differences in the log prob of the `top_k` set, the use of Sparse Mixture of Experts, and varying hardware at inference time.

6.1 Experiment Setup and Metrics

We started with a stratified random sample of the following datasets: Kitab, IFEval, Geometric Reasoning, and MMMU. We repeated inference on each instance three times independently using the same prompt templates, then measured the entropy or standard error of model outputs for the same instance for categorical and numeric labels, respectively. Finally, we report the mean of the entropy and standard deviation over all instances. Both scores characterize the amount of per-example variation in the final metric for the stratified sample. For categorical metrics, we also report the percentage of instances where the results obtained from the three independent runs are not in agreement. Note that none of the tasks included in this analysis use other non-deterministic language models for evaluation, which means that any observed non-determinism can be fully attributed to the inference of the model under test and not to the evaluation process itself.

6.2 Results

Among all models investigated, Gemini 1.5 Pro, GPT-4 Turbo 2024-04-09, and GPT-4 Vision Preview/GPT-4 1106 Preview consistently exhibit the most nondeterminism across the four tasks. For example, on multi-modal knowledge understanding (MMMU), the three independent runs on the same instance lead to different outcomes in 21% and 26% of the cases for GPT-4 Vision Preview and Gemini 1.5 Pro, respectively. Similarly, we observe a high standard deviation (3% - 11%) across runs on Kitab for both of these models. GPT-4 Turbo 2024-04-09 has the highest entropy among all models in the Geometrical reasoning task and the 3rd highest entropy in MMMU.

Llama 3 70B, Llama 3.1 70B, and Mistral Large 2407 consistently have non-determinism scores close to zero (lower non-determinism is better indicating perfect repeatability). A leading cause of this could be that these models are not Mixtures of Experts, while for the others there has been speculation in the community that this may be the case (although no official confirmation has been issued from OpenAI, Anthropic, or Google.)

The Claude family and GPT-4o 2024-05-13 follow after Llama 3.1 405B as the next most deterministic models, although GPT-4o 2024-05-13 is still notably non-deterministic in highly generative tasks like Kitab and GeoMeter datasets.

Geometric Reasoning: We sampled a subset of 75 instances per height/depth category (150 total) from GeoMeter, and inferenced each sample 3 times using all models. This dataset involves multiple-choice question where the model is required to generate one of the options from the provided options. The options are orderings of objects present in the image, therefore this is a relatively long generation task. The task metric is categorical (correct/incorrect/NA), therefore, to measure nondeterminism, we measure the entropy over the 3 runs for each instance and then average the results over all instances. Additionally, we report the percentage of instances where the three independent runs do not yield unanimous results.

As seen in Table 8, the Claude family exhibits the most deterministic behaviour (zero entropy), while GPT-4 Turbo 2024-04-09 is the most non-deterministic with 19.3% of cases yielding inconsistent outcomes in three runs. Looking at the overall performance means, and the standard error over the three means does not sufficiently inform us of this level of non-determinism at the individual instance level. The standard error is 1.77 on a 0-100 range for GPT-4 Turbo 2024-04-09 and 0.8 for GPT-4o 2024-05-13, while, respectively, 19.3% and 16.7% of cases yield different outcomes in 3 runs. In fact, while the standard error of the mean performance across the whole sample is useful for measuring the statistical robustness of the reported mean (as we do in previous sections), it is not a sufficiently good indicator for measuring non-determinism at the instance level, because the aggregation across the sample size amortizes the differences at instance level.

Model	% Instances with Different Outcomes	Average Entropy of Outcomes	Overall Perf. (on the sample)
Claude 3.5 Sonnet	0.0	0.00	43.33 (0.00)
Claude 3 Opus	0.0	0.00	29.33 (0.00)
Gemini 1.5 Pro	2.7	0.02	39.78 (0.22)
GPT-4o 2024-05-13	16.7	0.15	32.44 (0.80)
GPT-4 Turbo 2024-04-09	19.3	0.18	30.22 (1.74)
GPT-4 Vision Preview	5.3	0.05	28.00 (0.67)
Llava 1.6 34B	0.7	0.01	24.44 (0.22)

Table 8: Nondeterminism - Geometric reasoning: 75 instances from the height category and 75 instances from the depth category were randomly sampled and inferenced 3 times each. The percentage of cases where the 3 inference runs yield a different outcome, as well as the entropy of outcomes from the 3 different runs averaged over all instances are reported (The maximum possible entropy for 3 variables with 3 possible outcomes is $-\log_2(1/3) = 1.58$). Finally, the average and standard error of the overall performance on the three repetitions of the dataset are given in the last columns.

Model	% Instances with Different Outcomes	Average Entropy of Outcomes	Overall Perf. (on the sample)
Claude 3.5 Sonnet	6.7	0.06	61.33 (1.33)
Claude 3 Opus	5.3	0.05	46.89 (1.68)
Gemini 1.5 Pro	26.0	0.24	51.11 (2.34)
GPT-4o 2024-05-13	1.3	0.01	62.00 (0.67)
GPT-4 Turbo 2024-04-09	10.7	0.10	59.78 (0.38)
GPT-4 Vision Preview	21.3	0.20	52.89 (1.39)
Llava 1.6 34B	0.0	0.00	50.00 (0.00)

Table 9: Nondeterminism - MMMU: 150 instances were sampled using random stratified sampling from the MMMU dataset and inferenced 3 times each. We measure the entropy of the inference outcomes over the 3 runs for each instance and then average the results over all instances (The maximum possible entropy for 3 variables with 3 possible outcomes is $-\log_2(1/3) = 1.58$.) Additionally we report the percentage of instances where the three independent runs do not yield unanimous results. Finally, the average and standard error of the overall performance on the three repetitions of the dataset are given in the last columns.

MMMU: We sampled a subset of 150 instances using stratified sampling over the MMMU subjects and inferenced each instance three times independently. MMMU involves mostly multiple choice questions where the models are required to output the alphabet letter indicating the correct choice, and the performance metric is the average accuracy of this selection. Therefore, the inference can have three outcomes: correct, incorrect, and NA (reserved for cases where the model does not output a valid response).

As seen in Table 9, Gemini 1.5 Pro is the most non-deterministic model in this task, with average entropy of 0.24 and 26% of instances yielding different results in three runs. Llava 1.6 34B is a fully deterministic model (entropy=0) followed by GPT-4o 2024-05-13 which exhibits entropy of 0.1 and 1.3% inconsistent cases.

IFEval: A subsample of 150 instances obtained using random stratified sampling over instruction types was used to estimate nondeterminism of models on the IFEval dataset. We ran each instance through all models three times independently. IFEval involves free form long generation and the dataset comes with various evaluation metrics measuring if the instructions in the query were followed. To measure non-determinism, we focus on a binary metric indicating whether all instructions were strictly followed. We report the entropy of the 3 binary outcomes obtained for each instance, averaged over all instances, and the percentage of cases where the 3 outcomes disagree. We also report the average and standard error of the overall performance on the three repetitions of the dataset (See Table 10).

In line with our observations with GeoMeter and MMMU datasets, the overall standard error does not reflect how much non-determinism exists on individual instance level. For example, Gemini 1.5 Pro has standard error of 0.44 over 3 runs (0-100 metric range), but at individual instance level, 14% of the instances get inconsistent outcomes over 3 attempts.

Model	% Instances with Different Outcomes	Average Entropy of Outcomes	Overall Perf. (on the sample)
Claude 3.5 Sonnet	1.3	0.01	80.67 (0.38)
Claude 3 Opus	2.0	0.02	81.78 (0.44)
Gemini 1.5 Pro	14.0	0.13	78.22 (0.44)
GPT-4o 2024-05-13	5.3	0.05	84.22 (0.59)
GPT-4 1106 Preview	9.3	0.09	78.67 (1.15)
Llama 3.1 405B	3.3	0.03	83.33 (0.67)
Llama 3.1 70B	0.0	0.00	83.33 (0.00)
Llama 3 70B	0.7	0.01	80.22 (0.22)
Mistral Large 2407	0.0	0.00	75.33 (0.00)

Table 10: Nondeterminism - IFEval: 150 instances were sampled using random stratified sampling from the IFEval dataset and inferenced 3 times each. The percentage of cases where the 3 inference runs yield a different outcome, as well as the entropy of outcomes from the 3 different runs averaged over all instances are reported. The maximum entropy for 2 possible outcomes and 3 variables is 1.5. Finally, the average and standard error of the overall performance on the three repetitions are given in the last columns.

Model	Completeness	Satisfied Rate	Unsatisfied Rate	Irrelevant Rate
Claude 3.5 Sonnet	13.92 (0.90)	44.05 (2.55)	22.91 (2.05)	15.54 (1.52)
Claude 3 Opus	18.32 (0.13)	47.12 (0.25)	28.11 (0.28)	13.94 (0.32)
Gemini 1.5 Pro	6.94 (1.81)	14.60 (4.20)	12.83 (4.30)	14.52 (6.10)
GPT-4o 2024-05-13	15.85 (2.08)	41.68 (4.21)	21.82 (3.12)	15.39 (3.45)
GPT-4 1106 Preview	18.64 (2.51)	37.21 (4.89)	27.86 (5.33)	13.82 (4.07)
Llama 3.1 405B	13.01 (1.58)	45.63 (4.25)	16.68 (5.14)	16.58 (5.60)
Llama 3.1 70B	13.45 (0.00)	45.21 (0.00)	14.12 (0.00)	22.33 (0.00)
Llama 3 70B	10.08 (0.00)	31.93 (0.00)	20.12 (0.00)	42.12 (0.00)
Mistral Large 2407	13.76 (0.00)	37.23 (0.14)	31.08 (0.00)	28.35 (0.14)

Table 11: Nondeterminism - Kitab: A subset of 130 instances were sampled using random stratified sampling from the Kitab dataset and inferenced 3 times each. The average and standard error of the Kitab metrics on the three repetitions of each instance were calculated and then averaged over all instances.

As reported in Table 10, Llama 3.1 70B and Mistral Large 2407 are perfectly deterministic (zero entropy) on this dataset, followed by Llama 3 70B and the Claude family that both have entropy of 0.01. Gemini 1.5 Pro and the GPT family emerge as the most non-deterministic models on this dataset.

Kitab: Through stratified random sampling over constraint types, we sample 130 instances from the Kitab dataset and inference each of them 3 independent times through all models. In this task, the models are prompted to generate a list of books and reasons, making this a long generation task. Various metrics are reported for this task including constraint satisfaction rate, completeness, and irrelevance rate.

As discussed in the Geometric Reasoning non-determinism analysis, to characterize variation at the example level, it is important to take the average and standard error of the metrics on the three repetitions of each instance and then average over all instances, as opposed to averaging over the sample size first and measuring standard error of the three means. Therefore, the numbers in Table 11 represent the mean and standard error over the 3 repetitions of each instance which are in turn averaged over all instances.

Consistent with our observations with the IFEval dataset, Llama 3.1 70B and Llama 3 70B are perfectly deterministic on this dataset, followed by Mistral Large 2407 that exhibits near-zero non-determinism. The Claude family has relatively small standard error compared the GPT family, Gemini 1.5 Pro, and Llama 3.1 405B.

Main takeaways

- Several models in this analysis such as Gemini 1.5 Pro, GPT-4 1106 Preview, GPT-4 Vision Preview, and GPT-4 Turbo 2024-04-09 show high non-determinism of outcomes. While the sources of such non-determinism remain under-explored, these results raise important questions regarding the stability of user and developer experiences when repeatedly inferencing with identical queries.
- Llama 3 70B, Llama 3.1 70B, Mistral Large 2407, and Llava 1.6 34B consistently have non-determinism scores close to zero (lower non-determinism is better indicating perfect repeatability).
- The Claude family and GPT-4o 2024-05-13 follow after Llama 3.1 405B as the next most deterministic models, although GPT-4o 2024-05-13 is still notably non-deterministic in highly generative tasks like information retrieval with constraints (Kitab dataset) and geometric reasoning (GeoMeter).

7 Backward Compatibility Evaluation

In this section, we present comparison results between models in terms of how backward compatible they are with previous model versions within the same family. In particular, we measure *progress* in terms of percentage of examples for which the new model version is better than the previous one, and *regress* as the percentage of examples for which the new model version is worse. For cases when the metric is binary (correct vs. incorrect) progress and regress track flips in the metric, while when the metric is continuous they track cases when the difference in the metric is higher or lower than a threshold. Previous work has also formulated other backward compatibility metrics such as backward trust compatibility and backward error compatibility[102], which respectively focus on the stability of correct and incorrect answers. Here, we simplify the measures to progress and regress so we can also compare them relatively with the percentage of cases where there are no changes between the two versions.

Note that a model can regress at the example or subcategory level during an update *even though there is an overall positive improvement in performance during the update*. This can happen due to model stochasticity, shifts in training data as well as changes in architecture and training processes (e.g. post-training and instruction tuning). Measuring regress at the example level is important for two main scenarios. First, from a human-AI collaboration perspective, as end-users may become accustomed to tasks where they can expect strong versus weak performance, user experience can be negatively effected if examples that were accurate in the past become incorrect after an update. Additionally, complex and poorly understood interactions between pre-existing prompts and updates to models may require iterative efforts to re-engineer prompts that had been carefully crafting for the last model version. Recrafting prompts after model updates is a cumbersome and time-consuming process [47]. Second, from a systems building perspective (e.g. for multi-agent workflows) introducing new, unknown errors in the output of a model component that feeds into other components, may hurt the overall system performance even if that component has improved in isolation [79, 102]. Learning about regressions on a subcategory level instead is useful for debugging sudden increases in lack of subgroup robustness [13, 81].

7.1 Datasets and Models

We run this analysis on three model families (Claude, GPT, and Llama) which have a recent model release and for which the previous model before the release was also a highly capable model based on our measurements in EUREKA-BENCH. The comparison here would study cases where for example a given user or application builder would replace their inference calls to GPT-4 1106 Preview with GPT-4o 2024-05-13 (or GPT-4 Turbo 2024-04-09 with GPT-4o 2024-05-13 for a multimodal task) and measures the amount of expected regression that will be associated with the model substitution. For the analysis on the Llama family, we compare Llama 3.1 70B vs. Llama 3 70B as they have the same parameter size and cost wise it is reasonable to assume that users may want to switch between models with similar cost (although parameter size is not the only indicator for estimating cost). Other comparisons, for example to Llama 3.1 405B and even across different model families also make sense and can reveal useful insights.

On the language front, we use IFEval and Kitab, since their output is a long-form generation and long-form generation is more prone to fluctuations. On the multimodal side, we study MMMU as it is a challenging task that highlights significant variation among models.

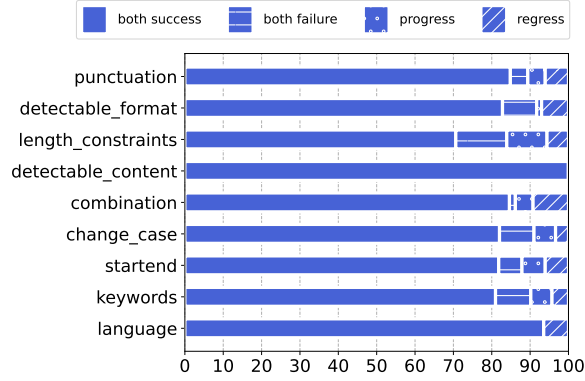


Figure 28: Backward compatibility between Claude 3.5 Sonnet and Claude 3 Opus for different instruction types in IFEval.

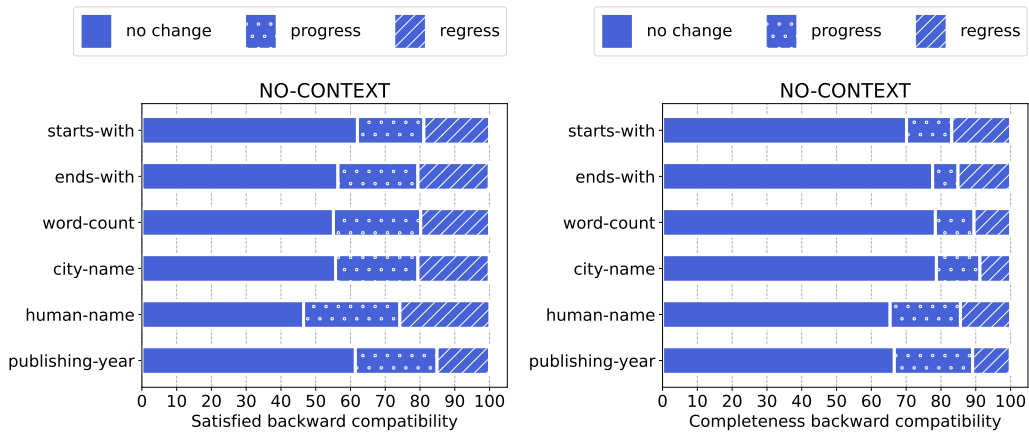


Figure 29: Backward compatibility between Claude 3.5 Sonnet and Claude 3 Opus for the NO-CONTEXT condition in Kitab, shown for satisfaction rate and completeness. Queries with one book constraint. No change indicates cases where the metric difference between the two models is less than 10 percentage points.

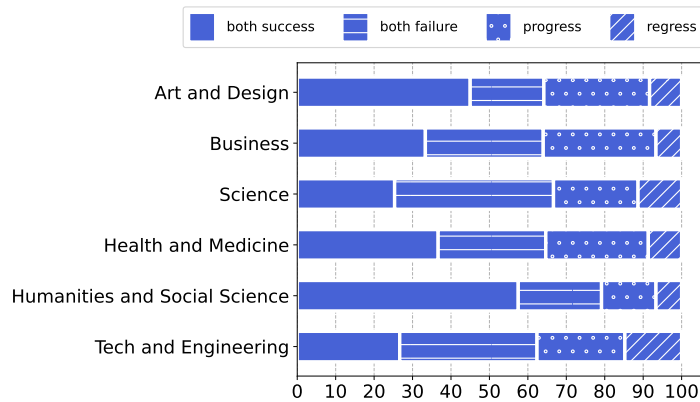


Figure 30: Backward compatibility between Claude 3.5 Sonnet and Claude 3 Opus for different disciplines in MMMU.

Model Family	Satisfied			Completeness			All Correct		
	no change	progress	regress	no change	progress	regress	no change	progress	regress
Claude	56.8	24.1	19.1	71.5	16.8	11.7	96.4	2.2	1.3
GPT	56.6	26.4	17.0	68.4	11.5	20.1	96.4	2.5	1.1
Llama	61.2	22.2	16.6	78.2	12.1	9.60	97.6	1.6	0.8

Table 12: Overall backward compatibility scores for different metric in the Kitab dataset. No change indicates cases where the difference in the metric between the two model versions is less than 10 percentage points.

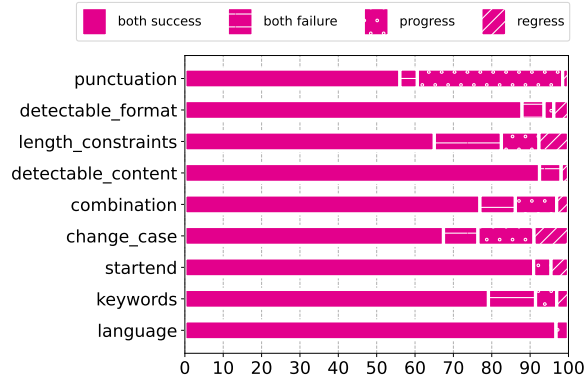


Figure 31: Backward compatibility between GPT-4o 2024-05-13 and GPT-4 1106 Preview for different instruction types in IFEval.

7.2 Claude 3.5 Sonnet vs. Claude 3 Opus

Figures 28, 29, and 30 show the backward compatibility analysis for the Claude family on the IFEval, Kitab, and MMMU datasets disaggregated by category to highlight what category is most impacted by regression.

For IFEval, most categories show >80% common successes where newer model maintains good performance from previous versions. Regression in >5% is observed in six out of nine categories with instructions involving combining responses (combination) showing ~10% regression. This highlights the potential inconsistency in instruction following behaviour that could impact applications relying on performance in specific instructions.

For Kitab, regression impacts most queries that require having a human-name in the title, with a regression rate of 23% for satisfaction rate. Regression rates for completeness are lower than for satisfaction rates in the Claude family, 11.7% vs. 19.1% (see Table 12 for details). Note that these regression scores are still relatively high considering that the models have a completeness rate of less than 25%. Also, when looking at completeness regression scores per subcategory, regression rates for string constraints like starts-with and ends-with dominate progress rates, leading to an overall drop in performance for that constraint type.

For MMMU, there is significant inconsistency, where most categories show only around 30-40% common successes. The newer model shows consistent progress across all disciplines; however, it also shows regressions in the range of 5-15% across the six disciplines. “Science” and “Tech and Engineering”, which are the two worst performing disciplines as shown in Figure 7, have the highest regression rate of ~10% and ~15%, respectively.

7.3 GPT-4o 2024-05-13 vs. GPT-4 1106 Preview/GPT-4 Turbo 2024-04-09

Figures 31, 32, and 33 show the backward compatibility analysis for the GPT family.

For IFEval, significant progression is observed in instructions involving constraints on casing (14.6%), punctuation usage (37.9%) and combining responses (10.8%). Regression is mainly observed to impact length constraints and case change instructions. For instructions involving length constraints, keyword constraints and case change there exists a 10% subset which is challenging for both models.

For the Kitab dataset, regression rates for completeness are higher than for constraint satisfaction, and in addition they dominate the progression rates by 9%. For MMMU, there is wide range of common successes of around 30-70%. The newer model shows similar rates of progression vs. regressions across all disciplines, which effectively cancels out and is consistent with the only 0.4% overall improvement of GPT-4o 2024-05-13

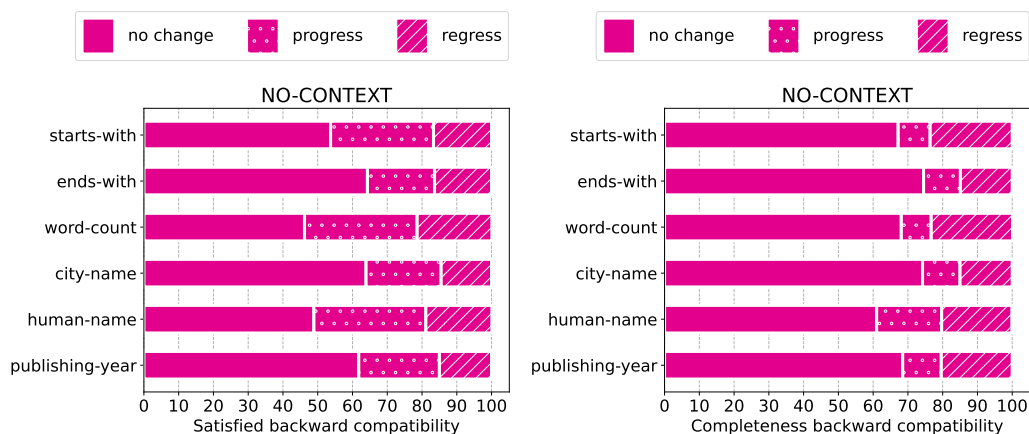


Figure 32: Backward compatibility between GPT-4o 2024-05-13 and GPT-4 1106 Preview for the NO-CONTEXT condition in Kitab, shown for satisfaction rate and completeness. Queries with one book constraint. No change indicates cases where the metric difference between the two models is less than 10 percentage points.

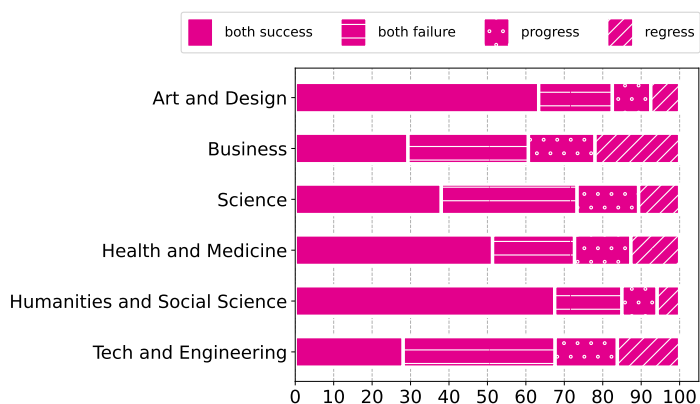


Figure 33: Backward compatibility between GPT-4o 2024-05-13 and GPT-4 Turbo 2024-04-09 for different disciplines in MMMU.

over GPT-4 Turbo 2024-04-09, as shown in Figure 6. This large difference in the instance level performance, with large per-discipline inconsistencies in success and failures and offsetting progression vs. regressions, shows how the overall performance numbers hide a high-level of differences in the per-instance model responses.

7.4 Llama 3.1 70B vs. Llama 3 70B

Figures 34, and 35 show the backward compatibility analysis for the Llama family. As this is a language only model family, we do not provide and analysis on MMMU for this family.

In IFEval setting, regression impacts only length constraints significantly, while other categories either strongly progress (in case of language instructions) or consistently fail across both models (as in length constraints, case change and keyword constraints).

For the Kitab dataset, regression rates for completeness are lower than for satisfaction rate and at the same time they are also lower than for other model families. This may be an artifact of the two models being of the same size, since completeness indirectly also shows how much information a model can store and access effectively.

Challenging subsets across Models: In addition to enabling analysis of progression and regression within model updates, example level backward comparability can also reveal subsets of data that are consistently challenging across models. In IFEval, models across both versions in Claude, GPT and Llama families fail to follow instructions in $\sim 10\%$ of the examples in length constraint, keyword constraints and case change

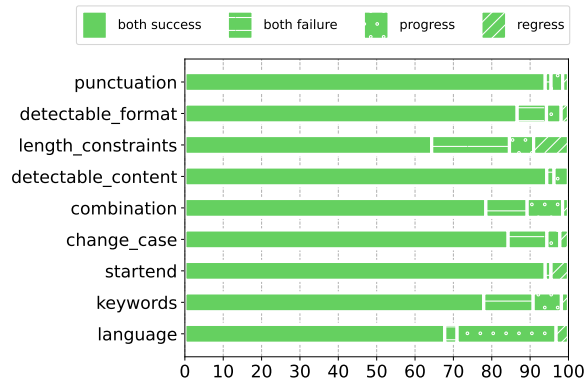


Figure 34: Backward compatibility between Llama 3.1 70B and Llama 3 70B for different instruction types in IFEval.

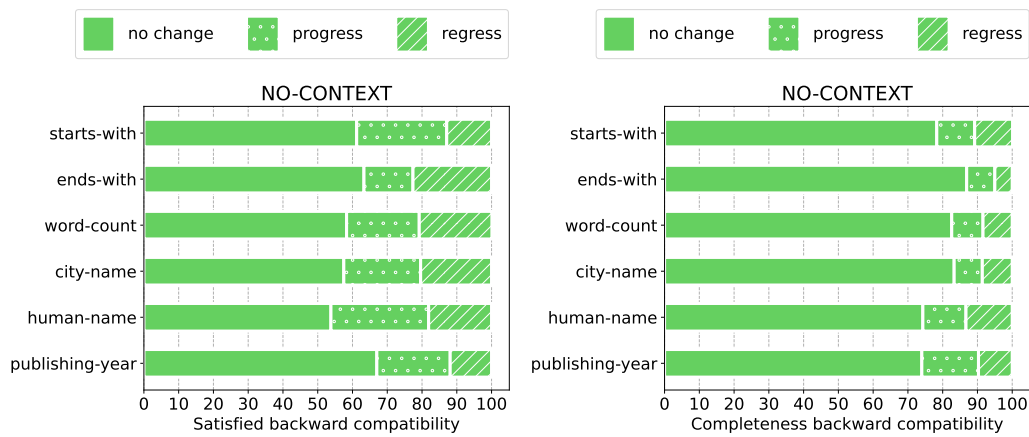


Figure 35: Backward compatibility between Llama 3.1 70B and Llama 3.1 70B for the NO-CONTEXT condition in Kitab, shown for satisfaction rate and completeness. Queries with one book constraint. No change indicates cases where the metric difference between the two models is less than 10 percentage points.

constraints. This subset is consistently challenging for all 6 strong models used for analysis and can highlight a set on instructions for future models to improve on. Similarly, in MMMU, the GPT and Claude families of models fail consistently in 15-35% of examples across different subcategories, which indicates there is a challenging subset of the data for further progress.

7.5 Main Takeaways

- Backward incompatibility is prevalent across state-of-the-art models reflected in high regression rates for individual examples and at a subcategory level. Detailed analysis here can help in navigating tradeoffs for model selection.
- Different performance metrics may be differently impacted by backward incompatibility.
- Backward incompatibility analysis can also uncover groups of examples for which all models consistently struggle with. For example, in IFEval, models across both versions in Claude, GPT and Llama families fail to follow instructions in ~10% of the examples in length constraint, keyword constraints and case change constraints.

8 Related Work and Limitations

We have presented an extensive evaluation and analysis of several state-of-the-art models on key language and multimodal capabilities. We note that, while many of these tasks are fundamental and core to many real-world applications, several areas that are crucial for exhaustive model evaluation are not covered. We plan to include several of these in future iterations. Here, we give an overview of contemporary capabilities and directions that need to be explored. Parallel to the coverage discussion in terms of capabilities is a separate question on how much data and diversity is needed for each capability for informative and generalizable insights. While we select the representative datasets in EUREKA-BENCH to have an interesting and diverse set of subcategories to address this question, some of the capabilities may require more diversity and depth than others because of their broad definition. It is however important to note that merely adding more datasets to the collection may not directly address diversity; many benchmarks may be correlated [85, 126] and thus only contribute to increasing cognitive load and lower clarity in results presentation. It has also been shown that aggregating ranks of models across many very diverse benchmarks results in unreliable and unstable overall scores [126]. This is why we do not aggregate across the different capabilities covered in this report, but only across experimental conditions and groups of data that are related.

8.1 Capability Evaluations

Responsible AI: A cross-cutting dimension for all capability evaluations is the evaluation of several aspects of model behavior important for the responsible fielding of AI systems. These considerations include the fairness, reliability, safety, privacy, and security of models. While evaluations through the Toxigen dataset capture notions of representational fairness for different demographic groups and, to some extent, the ability of the model to generate safe language despite non-safe input triggers in the prompt, other aspects or nuances of fairness and safety require further evaluation and additional clarity, which we hope to cover in future versions of this report. Evaluation efforts to this end include SorryBench [117], CocoNot [17], DecodingTrust [111], TrustLLM [111], MLCommons Safety Benchmark 0.5 [109], Cybench [125]. A general rising concern on most aspects however is that there is a quick turnaround between these benchmarks being released and then included in content safety filters or in post training datasets. For example, during initial investigations we observed that most models evaluated in this report have safety scores of higher than 97% in the DoNotAnswer benchmark [115], which is a positive development. However, from an evaluation and understanding perspective, the high score indicates that the benchmark is not sensitive enough to capture differences among models. Finally, responsibility in models that can generate images, video, and audio remains heavily under explored. Of these capabilities, image generation is most studied and evaluated due to the pervasiveness and popularity of models in the past two years [26, 75, 66, 94].

Multilingual Capabilities: We note that all results in this report evaluate language capabilities in English. Several previous works have raised the importance of multilingual capabilities as a major generalisation aspect and have built dedicated benchmarks for this purpose [7, 8, 128, 128, 128]. The work on multilingual evaluation has shown that there exist major discrepancies on model performance between languages. Particular challenges have been noted for low-resource languages, which raises a global fairness concern [28, 86, 55]. In addition, a lack of multilingual understanding has been found to be a factor in challenges with responsible AI: jailbreaking even the most advanced models has been shown to be easier if done via a low-resource language [31, 121]. Despite these problems and the availability of more advanced and challenging benchmarks, multilingual evaluation in major model releases still focuses on oversimplified settings. For example, MGSM [98] translates 250 examples from GSM8K in 10 different languages. However, given that GSM8k is subject to saturation and also the fact that the answer to the model is expected to be merely a number with little language around it, the benchmark does not test the quality of the generated language per se. Similar concerns are present for cases when existing multiple-choice benchmarks like MMLU and others are translated from English.

Reasoning and Planning: The Kitab and FlenQA datasets for language and GeoMeter and Vision & Language Understanding present in EUREKA touch upon different aspects of reasoning such as constraint satisfaction, logical reasoning, and spatial and geometric understanding. There are however several other aspects of reasoning and planning where the evaluation of the model is assessed based on the effectiveness of steps and plans that a model generates to solve a problem [108, 74, 99, 105, 130], and that we plan to include in future iterations.

Multimodal Evaluation: The overall consensus in the field regarding planning and reasoning is that multi-step planning in language and planning actions in the physical, multimodal world [87, 24] are still nascent. In particular for physical world planning and assistance, more work is needed for assessing reasoning skills on modalities that go beyond image and text to encompass video, audio, and inferences based on fusion across

multiple modalities. The fundamental tasks required for supporting true and physical multimodal interaction are numerous, including activity recognition, temporal reasoning, compositional reasoning, summarization and grounding, and event and state detection [93, 84, 41, 56, 69, 14]. Recent evaluations of systems that are designed to assist humans on physical tasks [15, 14] show how the fundamental capabilities that need to be evaluated to support these systems need to be extended with a rich set of subtasks that must be solved to support well-timed and formulated contributions by an AI system (e.g. Is now a good time or state to intervene?) or with capabilities to interpret subtle cues available via a single modality (e.g., face expression or voice tone which cannot be understood via text only).

Related to end-to-end system evaluation, model-based evaluations are only a small piece of the puzzle, as the larger systems that are designed to assist people in real-world, interactive settings are often composed of complex architectures that call models in the course of their operation. Evaluating the impact and effectiveness systems in human-AI collaborative settings requires rigorous and frequent studies with human subjects who engage with the systems in realistic scenarios. Evaluating models on benchmarks provides an initial indication about granular skills that are important for the larger applications. Beyond narrow measures of capability, the design and operation of these systems needs to consider the multiple dimensions of the user experience, including design of interfaces, workflow design [60, 39], and overreliance and trust of AI systems [83, 18].

8.2 Evaluation methodologies and frameworks

Model Evaluation Frameworks: As models improve in performance across wider range of capabilities, model evaluation has moved from evaluating on single task-specific test sets to broad benchmarks covering multiple tasks. Meta benchmarks like SuperGLUE [110], HELM [59], BigBench [101], Open LLM Leaderboard [40], HELM [53], MMBench [64] have aggregated evaluation sets from multiple tasks to enable broader study of model performance via common evaluations. A promising aspect of building meta evaluations is the standardization of inference and evaluation platform across tasks and models through frameworks like Eleuther Language Model Evaluation Harness [42], which provide transparency and reproducibility in the evaluation process. Such benchmarks have long been used to establish model performance and state-of-art claims in frontier model reports, but lack visibility into important experimental conditions and subcategories which often influence model selection for downstream applications [38, 100, 78, 11]. Hence, in this report we specifically focused on deeper analysis on specific, important capabilities rather than aggregating across benchmarks to identify overall model rankings. In parallel, platforms like LMSys Chatbot Arena [25] conduct large-scale pairwise preference evaluations on more open-ended questions with either humans or LLMs as judges producing real-world user aligned model rankings. Unfortunately, as preferences are binary signals aggregated to a single ELO Rating [37], preference ranking evaluations are unable to produce deeper insights needed for model selection and further development, and often encode subtle user or model biases like preference for assertive or longer outputs [46, 52, 113, 116].

Evaluation Methodology Advances: *Memorization* is a phenomenon that is closely associated with the saturation of benchmarks themselves. Several works have flagged the importance of considering and evaluating the impact of memorization in models. Understanding and evaluating the influence of memorization is a challenging endeavour given the continuing lack of transparency on training data details. Several research efforts have reported verbatim repetitions from test sets [16], drops in performance when the test dataset itself is recollected or expanded [127, 29, 118], or evidence that the test data is present in training set whenever available [30, 43]. While there has been some progress in building tools for distinguishing memorization from models genuinely improving in a capability [76], generalizable methods across modalities and datasets are not yet available. This means that, while we do focus on non-saturated benchmarks, we cannot guarantee that these benchmarks are not present in training sets. However, it is an indication that even if (in worst case) the test data was used for training, the model still is not able to perform well.

Nevertheless, many datasets in EUREKA-BENCH do have a *dynamic and procedural nature* and new samples of the data can be generated in the future. For example, all data in GeoMeter, Image Understanding, Vision Language Understanding, and Kitab can be re generated by using a different set of images, questions (for Vision Language Understanding), or authors (for Kitab). Toxigen instead was created by using an adaptive adversarial decoding technique with a classifier in the loop that continues to find new gaps on a model, however the resulting dataset sample was also manually curated by humans [44]. It will be interesting in the future to conduct side-by-side comparisons between current and future test data versions.

The idea of *procedurally and dynamically generating data* either in a controlled way or via a distribution has attracted interest with techniques like DyVal [134], AutoBench [58], S-Eval [122]. and self-evolving benchmarks [114]. An unanswered question is how to ensure that inferences for evaluation purposes are not

used as part of the training process itself accidentally or intentionally. For models served behind apis, ensuring non contamination in the long term remains at the discretion of model providers, as the api call itself reveals the test data (also when the test set is private to the evaluator and not public knowledge) even though it may not reveal the ground truth per se. In EUREKA-BENCH, we choose to prioritize transparency and reproducibility but it is also important to consider other forms of transparency that do not necessarily require full access to the whole test data. Recent work in evaluation methodology [65, 67] for example provides a process framework and guidance to adapting over time and across tasks, motivating the need for evaluation efforts to adapt and revise both test cases and methods.

Finally, as shown in sections dedicated to non-determinism and backward compatibility, LFM’s are sensitive to a myriad of parameters and conditions. *Prompt sensitivity* [49, 129] and *few-shot design* [77, 82] are amongst the top important ones and where there has been continuous evidence that the actual prompt or in-context examples being used have major impact on measurement. For example, Section 4.2 results on MMMU are an illustration of high prompt sensitivity across models. For other benchmarks, we rely on prompts that are well validated and vetted by the authors of such datasets or from pre investigations we did to make sure that model performance is not understated. While it is possible to run extensive experimentation through EUREKA by changing and reusing prompt templates, we also plan to devise and leverage techniques that can do this in a faster and cheaper manner [92, 49].

9 Conclusion

In conclusion, our work with the formulation of EUREKA and study of a set challenging of benchmarks highlights the critical need for more rigorous and nuanced evaluation of large foundation models (LFMs). Despite significant advancements in AI capabilities, we find that current models exhibit substantial weaknesses across various tasks. The complementary strengths of different models suggest that no single model currently excels across all capabilities, underscoring the importance of continued innovation and targeted improvements guided by detailed considerations of evaluations.

Moreover, our disaggregated approach to evaluation exposes granular failures that traditional, aggregate metrics often overlook. This level of detail is essential for identifying and addressing specific areas where models falter, thereby informing both future model development and the selection of benchmarks that remain relevant and challenging. Insights from our studies of backward compatibility and non-determinism raise important questions about the stability and consistency of AI models, especially as they evolve and are used repeatedly over time. These findings emphasize the need for ongoing, transparent evaluation practices that can adapt to the rapid pace of AI development and fielding.

We developed the EUREKA to facilitate deeper understandings of current LFM’s and to lay the groundwork for supporting more effective and targeted improvements. By making these tools and benchmarks available as open-source resources, we hope to foster a collaborative effort within the AI community to enhance the robustness, transparency, and reproducibility of model evaluations.

Acknowledgements

We would like to thank Ahmed Awadallah, Ece Kamar, Eric Horvitz, John Langford, Rafah Hosn, Saleema Amershi for valuable discussions and guidance throughout the whole timeline of the project. We would also like to thank several colleagues and collaborators that have worked and brainstormed with us on different evaluation efforts, and have informed design and scientific choices we have made in this work: Adam Fourney, Akshay Nambi, Alessandro Stolfo, Allie Del Giorno, Arindam Mitra, Clarisse Simoes, Dan Bohus, Dimitris Papailiopoulos, Forough Poursabzi, Gagan Bansal, Ida Momennejad, Jennifer Neville, Julia Kiseleva, Marah Abdin, Marco Rossi, Mazda Moayeri, Natasha Butt, Rahee Ghosh Peshawaria, Ronen Eldan, Saleema Amershi, Sean Andrist, Shital Shah, Shweti Mahajan, Siddharth Joshi, Suriya Gunasekar, Sunayana Sitaram, Tobias Schnabel, Victor Dibia, and Xin Wang.

References

- [1] Claude 3.5 sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>, 2024. Accessed: 2024-08-13.
- [2] Llava. https://huggingface.co/docs/transformers/en/model_doc/llava, 2024.
- [3] Mistral large 2. <https://mistral.ai/news/mistral-large-2407/>, 2024. Accessed: 2024-08-13.
- [4] Mmmu evaluation. <https://github.com/MMMU-Benchmark/MMMU>, 2024.
- [5] Openai evals. <https://github.com/openai/evals>, 2024.
- [6] M. I. Abdin, S. Gunasekar, V. Chandrasekaran, J. Li, M. Yüksekönül, R. G. Peshawaria, R. Naik, and B. Nushi. KITAB: evaluating llms on constraint satisfaction for information retrieval. In *International Conference on Learning Representations*, 2024.
- [7] K. Ahuja, H. Diddee, R. Hada, M. Ochieng, K. Ramesh, P. Jain, A. U. Nambi, T. Ganu, S. Segal, M. Ahmed, K. Bali, and S. Sitaram. MEGA: multilingual evaluation of generative AI. In H. Bouamor, J. Pino, and K. Bali, editors, *Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, 2023.
- [8] S. Ahuja, D. Aggarwal, V. Gumma, I. Watts, A. Sathe, M. Ochieng, R. Hada, P. Jain, M. Axmed, K. Bali, et al. Megaverse: Benchmarking large language models across languages, modalities, models and tasks. *arXiv preprint arXiv:2311.07463*, 2023.
- [9] S. Azad, Y. Jain, R. Garg, Y. Rawat, , and V. Vineet. Dh-bench: Probing depth and height perception of large visual-language models. *arXiv preprint arXiv:2408.11748*, 2024.
- [10] G. Bansal, B. Nushi, E. Kamar, D. S. Weld, W. S. Lasecki, and E. Horvitz. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *AAAI Conference on Artificial Intelligence*, 2019.
- [11] S. Barocas, A. Guo, E. Kamar, J. Krones, M. R. Morris, J. W. Vaughan, W. D. Wadsworth, and H. Wallach. Designing disaggregated evaluations of ai systems: Choices, considerations, and tradeoffs. In *AAAI/ACM Conference on AI, Ethics, and Society*, pages 368–378, 2021.
- [12] M. S. Bernstein, J. Teevan, S. Dumais, D. Liebling, and E. Horvitz. Direct answers for search queries in the long tail. In *SIGCHI conference on human factors in computing systems*, pages 237–246, 2012.
- [13] M. Bertran, N. Martinez, A. Oesterling, and G. Sapiro. Distributionally robust group backwards compatibility. *arXiv preprint arXiv:2112.10290*, 2021.
- [14] D. Bohus, S. Andrist, Y. Bao, E. Horvitz, and A. Paradiso. Is this it?: Towards ecologically valid benchmarks for situated collaboration. In *International Conference on Multimodal Interaction (ICMI Companion '24)*, 2024.
- [15] D. Bohus, S. Andrist, N. Saw, A. Paradiso, I. Chakraborty, and M. Rad. Sigma: An open-source interactive system for mixed-reality task assistance research—extended abstract. In *IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 889–890. IEEE, 2024.
- [16] S. Bordt, H. Nori, V. Rodrigues, B. Nushi, and R. Caruana. Elephants never forget: Memorization and learning of tabular data in large language models. In *Conference on Language Modeling*, 2024.
- [17] F. Brahman, S. Kumar, V. Balachandran, P. Dasigi, V. Pyatkin, A. Ravichander, S. Wiegrefe, N. Dziri, K. Chandu, J. Hessel, et al. The art of saying no: Contextual noncompliance in language models. *arXiv preprint arXiv:2407.12043*, 2024.
- [18] Z. Buçinca, M. B. Malaya, and K. Z. Gajos. To trust or to think: cognitive forcing functions can reduce overreliance on ai in ai-assisted decision-making. *ACM on Human-computer Interaction*, 5(CSCW1):1–21, 2021.

- [19] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.
- [20] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Driess, P. Florence, D. Sadigh, L. Guibas, and F. Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024.
- [21] L. Chen, M. Zaharia, and J. Zou. How is chatgpt’s behavior changing over time? *arXiv preprint arXiv:2307.09009*, 2023.
- [22] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation, 2017.
- [23] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- [24] Y. Chen, Y. Ge, Y. Ge, M. Ding, B. Li, R. Wang, R. Xu, Y. Shan, and X. Liu. Egoplan-bench: Benchmarking egocentric embodied planning with multimodal large language models. *arXiv preprint arXiv:2312.06722*, 2023.
- [25] W.-L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, H. Zhang, B. Zhu, M. Jordan, J. E. Gonzalez, and I. Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.
- [26] J. Cho, A. Zala, and M. Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. In *IEEE/CVF International Conference on Computer Vision*, pages 3043–3054, 2023.
- [27] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [28] J. Cunningham, S. L. Blodgett, M. Madaio, H. D. Iii, C. Harrington, and H. Wallach. Understanding the impacts of language technologies’ performance disparities on african american language speakers. In *Findings of the Association for Computational Linguistics ACL*, pages 12826–12833, 2024.
- [29] J. Dekoninck, M. N. Müller, and M. Vechev. Constat: Performance-based contamination detection in large language models. *arXiv preprint arXiv:2405.16281*, 2024.
- [30] C. Deng, Y. Zhao, X. Tang, M. Gerstein, and A. Cohan. Investigating data contamination in modern benchmarks for large language models. *arXiv preprint arXiv:2311.09783*, 2023.
- [31] Y. Deng, W. Zhang, S. J. Pan, and L. Bing. Multilingual jailbreak challenges in large language models. In *International Conference on Learning Representations*, 2024.
- [32] A. Diwan, L. Berry, E. Choi, D. Harwath, and K. Mahowald. Why is winoground hard? investigating failures in visuolinguistic compositionality. *arXiv preprint arXiv:2211.00768*, 2022.
- [33] D. Downey, S. Dumais, and E. Horvitz. Heads and tails: studies of web search with common and rare queries. In *ACM SIGIR conference on Research and development in information retrieval*, pages 847–848, 2007.
- [34] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In J. Burstein, C. Doran, and T. Solorio, editors, *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2368–2378, 2019.
- [35] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [36] J. Echterhoff, F. Faghri, R. Vemulapalli, T.-Y. Hu, C.-L. Li, O. Tuzel, and H. Pouransari. Muscle: A model update strategy for compatible llm evolution. *arXiv preprint arXiv:2407.09435*, 2024.

- [37] A. E. Elo. *The Rating of Chessplayers, Past and Present*. Arco Pub., New York, 1978.
- [38] S. Eyuboglu, M. Varma, K. K. Saab, J. Delbrouck, C. Lee-Messer, J. Dunnmon, J. Zou, and C. Ré. Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations*, 2022.
- [39] R. Fogliato, S. Chappidi, M. Lungren, P. Fisher, D. Wilson, M. Fitzke, M. Parkinson, E. Horvitz, K. Inkpen, and B. Nushi. Who goes first? influences of human-ai workflow on decision making in clinical imaging. In *ACM Conference on Fairness, Accountability, and Transparency*, pages 1362–1374, 2022.
- [40] C. Fourrier, N. Habib, A. Lozovskaya, K. Szafer, and T. Wolf. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.
- [41] C. Fu, Y. Dai, Y. Luo, L. Li, S. Ren, R. Zhang, Z. Wang, C. Zhou, Y. Shen, M. Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.
- [42] L. Gao, J. Tow, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, K. McDonell, N. Muenighoff, J. Phang, L. Reynolds, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou. A framework for few-shot language model evaluation, 2021.
- [43] S. Golchin and M. Surdeanu. Data contamination quiz: A tool to detect and estimate contamination in large language models. *arXiv preprint arXiv:2311.06233*, 2023.
- [44] T. Hartvigsen, S. Gabriel, H. Palangi, M. Sap, D. Ray, and E. Kamar. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Association for Computational Linguistics*, pages 3309–3326, 2022.
- [45] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.
- [46] T. Hosking, P. Blunsom, and M. Bartolo. Human feedback is not gold standard. In *International Conference on Learning Representations*, 2024.
- [47] E. Jahani, B. S. Manning, J. Zhang, H.-Y. TuYe, M. Alsobay, C. Nicolaidis, S. Suri, and D. Holtz. As generative models improve, people adapt their prompts. *arXiv preprint arXiv:2407.14333*, 2024.
- [48] B. Jayaraman, C. Guo, and K. Chaudhuri. D\`ej\`a vu memorization in vision-language models. *arXiv preprint arXiv:2402.02103*, 2024.
- [49] C. Jin, H. Peng, S. Zhao, Z. Wang, W. Xu, L. Han, J. Zhao, K. Zhong, S. Rajasekaran, and D. N. Metaxas. Apeer: Automatic prompt engineering enhances large language model reranking. *arXiv preprint arXiv:2406.14449*, 2024.
- [50] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.
- [51] A. Kembhavi, M. Salvato, E. Kolve, M. J. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *The European Conference on Computer Vision*, pages 235–251, 2016.
- [52] R. Koo, M. Lee, V. Raheja, J. I. Park, Z. M. Kim, and D. Kang. Benchmarking cognitive biases in large language models as evaluators. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 517–545, 2024.
- [53] T. Lee, M. Yasunaga, C. Meng, Y. Mai, J. S. Park, A. Gupta, Y. Zhang, D. Narayanan, H. Teufel, M. Belagente, M. Kang, T. Park, J. Leskovec, J.-Y. Zhu, F.-F. Li, J. Wu, S. Ermon, and P. S. Liang. Holistic evaluation of text-to-image models. In *Advances in Neural Information Processing Systems*, 2023.
- [54] M. Levy, A. Jacoby, and Y. Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models, 2024.

- [55] C. Li, M. Chen, J. Wang, S. Sitaram, and X. Xie. Culturellm: Incorporating cultural differences into large language models. *arXiv preprint arXiv:2402.10946*, 2024.
- [56] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, Y. Liu, Z. Wang, J. Xu, G. Chen, P. Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.
- [57] T. Li, G. Zhang, Q. D. Do, X. Yue, and W. Chen. Long-context llms struggle with long in-context learning, 2024.
- [58] X. L. Li, E. Z. Liu, P. Liang, and T. Hashimoto. Autobencher: Creating salient, novel, difficult datasets for language models. *arXiv preprint arXiv:2407.08351*, 2024.
- [59] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. A. Cosgrove, C. D. Manning, C. Re, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. WANG, K. Santhanam, L. Orr, L. Zheng, M. Yuksekogonul, M. Suzgun, N. Kim, N. Guha, N. S. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. A. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, and Y. Koreeda. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.
- [60] Q. V. Liao, M. Vorvoreanu, H. Subramonyam, and L. Wilcox. Ux matters: The critical role of ux in responsible ai. *Interactions*, 31(4):22–27, 2024.
- [61] S. Lin, J. Hilton, and O. Evans. Truthfulqa: Measuring how models mimic human falsehoods. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Association for Computational Linguistics*, pages 3214–3252, 2022.
- [62] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.
- [63] F. Liu, G. Emerson, and N. Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023.
- [64] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, K. Chen, and D. Lin. Mmbench: Is your multi-modal model an all-around player? *ECCV*, 2023.
- [65] Y. L. Liu, S. L. Blodgett, J. C. K. Cheung, Q. V. Liao, A. Olteanu, and Z. Xiao. Ecbd: Evidence-centered benchmark design for nlp. *arXiv preprint arXiv:2406.08723*, 2024.
- [66] S. Luccioni, C. Akiki, M. Mitchell, and Y. Jernite. Stable bias: Evaluating societal representations in diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [67] L. Lucy, S. L. Blodgett, M. Shokouhi, H. Wallach, and A. Olteanu. “one-size-fits-all”? examining expectations around what constitute “fair” or “good” nlg system behaviors. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1054–1089, 2024.
- [68] W. Ma, C. Yang, and C. Kästner. (why) is my prompt getting worse? rethinking regression testing for evolving llm apis. In *IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, pages 166–171, 2024.
- [69] K. Mangalam, R. Akshulakov, and J. Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, 2023.
- [70] A. Masry, D. X. Long, J. Q. Tan, S. R. Joty, and E. Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings of the Association for Computational Linguistics*, pages 2263–2279, 2022.
- [71] R. Matsuno and K. Sakuma. A robust backward compatibility metric for model retraining. In *ACM International Conference on Information and Knowledge Management*, pages 4190–4194, 2023.

- [72] K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.
- [73] M. Moayeri, E. Tabassi, and S. Feizi. Worldbench: Quantifying geographic disparities in llm factual recall. In *ACM Conference on Fairness, Accountability, and Transparency*, pages 1211–1228, 2024.
- [74] I. Momennejad, H. Hasanbeig, F. Vieira Frujeri, H. Sharma, N. Jovic, H. Palangi, R. Ness, and J. Larson. Evaluating cognitive maps and planning in large language models with cogeval. *Advances in Neural Information Processing Systems*, 36, 2024.
- [75] R. Naik and B. Nushi. Social biases through the text-to-image generation lens. In *AAAI/ACM Conference on AI, Ethics, and Society*, pages 786–808, 2023.
- [76] H. Nori, N. King, S. M. McKinney, D. Carignan, and E. Horvitz. Capabilities of gpt-4 on medical challenge problems, 2023.
- [77] H. Nori, Y. T. Lee, S. Zhang, D. Carignan, R. Edgar, N. Fusi, N. King, J. Larson, Y. Li, W. Liu, et al. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*, 2023.
- [78] B. Nushi, E. Kamar, and E. Horvitz. Towards accountable ai: Hybrid human-machine analyses for characterizing system failure. In *AAAI Conference on Human Computation and Crowdsourcing*, volume 6, pages 126–135, 2018.
- [79] B. Nushi, E. Kamar, E. Horvitz, and D. Kossmann. On human intellect and machine failures: Troubleshooting integrative machine learning systems. In *AAAI Conference on Artificial Intelligence*, 2017.
- [80] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.
- [81] Y. Oren, S. Sagawa, T. B. Hashimoto, and P. Liang. Distributionally robust language modeling. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, pages 4226–4236. Association for Computational Linguistics, 2019.
- [82] A. Parnami and M. Lee. Learning from few examples: A summary of approaches to few-shot learning. *arXiv preprint arXiv:2203.04291*, 2022.
- [83] S. Passi and M. Vorvoreanu. Overreliance on ai literature review. *Microsoft Research*, 2022.
- [84] V. Patraucean, L. Smaira, A. Gupta, A. Recasens, L. Markeeva, D. Banarse, S. Koppula, M. Malinowski, Y. Yang, C. Doersch, et al. Perception test: A diagnostic benchmark for multimodal video models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [85] Y. Perlitz, A. Gera, O. Arviv, A. Yehudai, E. Bandel, E. Shnarch, M. Shmueli-Scheuer, and L. Choshen. Benchmark agreement testing done right: A guide for llm benchmark evaluation. *arXiv preprint arXiv:2407.13696*, 2024.
- [86] K. Ramesh, S. Sitaram, and M. Choudhury. Fairness in language models beyond english: Gaps and challenges. In A. Vlachos and I. Augenstein, editors, *Association for Computational Linguistics*, pages 2061–2074, 2023.
- [87] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. D. Reid, and N. Sünderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In J. Tan, M. Toussaint, and K. Darvish, editors, *Conference on Robot Learning*, volume 229 of *Machine Learning Research*, pages 23–72, 2023.
- [88] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [89] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

- [90] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.
- [91] M. Research. Image understanding benchmark. <https://aka.ms/image-understanding-benchmark>, 2024.
- [92] T. Schnabel and J. Neville. Prompts as programs: A structure-aware approach to efficient compile-time prompt optimization. *arXiv preprint arXiv:2404.02319*, 2024.
- [93] P. Sermanet, T. Ding, J. Zhao, F. Xia, D. Dwibedi, K. Gopalakrishnan, C. Chan, G. Dulac-Arnold, S. Maddineni, N. J. Joshi, P. Florence, W. Han, R. Baruch, Y. Lu, S. Mirchandani, P. Xu, P. Sanketi, K. Hausman, I. Shafran, B. Ichter, and Y. Cao. Robovqa: Multimodal long-horizon reasoning for robotics. In *International Conference on Robotics and Automation*, pages 645–652. IEEE, 2024.
- [94] P. Seshadri, S. Singh, and Y. Elazar. The bias amplification paradox in text-to-image generation. *arXiv preprint arXiv:2308.00755*, 2023.
- [95] U. Shaham, M. Ivgi, A. Efrat, J. Berant, and O. Levy. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, 2023.
- [96] Y. Shen, Y. Xiong, W. Xia, and S. Soatto. Towards backward-compatible representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6368–6377, 2020.
- [97] F. Shi, M. Suzgun, M. Freitag, X. Wang, S. Srivats, S. Vosoughi, H. W. Chung, Y. Tay, S. Ruder, D. Zhou, D. Das, and J. Wei. Language models are multilingual chain-of-thought reasoners. In *International Conference on Learning Representations*, 2023.
- [98] F. Shi, M. Suzgun, M. Freitag, X. Wang, S. Srivats, S. Vosoughi, H. W. Chung, Y. Tay, S. Ruder, D. Zhou, D. Das, and J. Wei. Language models are multilingual chain-of-thought reasoners. In *International Conference on Learning Representations*, 2023.
- [99] T. Silver, V. Hariprasad, R. S. Shuttlesworth, N. Kumar, T. Lozano-Pérez, and L. P. Kaelbling. Pddl planning with pretrained large language models. In *NeurIPS 2022 foundation models for decision making workshop*, 2022.
- [100] S. Singla, B. Nushi, S. Shah, E. Kamar, and E. Horvitz. Understanding failures of deep networks via robust feature extraction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12853–12862, 2021.
- [101] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.
- [102] M. Srivastava, B. Nushi, E. Kamar, S. Shah, and E. Horvitz. An empirical analysis of backward compatibility in machine learning systems. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3272–3280, 2020.
- [103] S. Subramanian, W. Merrill, T. Darrell, M. Gardner, S. Singh, and A. Rohrbach. Reclip: A strong zero-shot baseline for referring expression comprehension, 2022.
- [104] T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela, and C. Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022.
- [105] Y. Tian, A. Ravichander, L. Qin, R. L. Bras, R. Marjeh, N. Peng, Y. Choi, T. L. Griffiths, and F. Brahman. Macgyver: Are large language models creative problem solvers? *arXiv preprint arXiv:2311.09682*, 2023.
- [106] F. Träuble, J. Von Kügelgen, M. Kleindessner, F. Locatello, B. Schölkopf, and P. Gehler. Backward-compatible prediction updates: A probabilistic approach. *Advances in Neural Information Processing Systems*, 34:116–128, 2021.

- [107] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.
- [108] K. Valmeekam, M. Marquez, A. O. Hernandez, S. Sreedharan, and S. Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, 2023.
- [109] B. Vidgen, A. Agrawal, A. M. Ahmed, V. Akinwande, N. Al-Nuaimi, N. Alfaraj, E. Alhajjar, L. Aroyo, T. Bavalatti, B. Blili-Hamelin, et al. Introducing v0. 5 of the ai safety benchmark from mlcommons. *arXiv preprint arXiv:2404.12241*, 2024.
- [110] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*, 2019.
- [111] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. In *Advances in Neural Information Processing Systems*, 2023.
- [112] J. Wang, Y. Ming, Z. Shi, V. Vineet, X. Wang, and N. Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision language models, 2024.
- [113] P. Wang, L. Li, L. Chen, Z. Cai, D. Zhu, B. Lin, Y. Cao, L. Kong, Q. Liu, T. Liu, and Z. Sui. Large language models are not fair evaluators. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2024.
- [114] S. Wang, Z. Long, Z. Fan, Z. Wei, and X. Huang. Benchmark self-evolving: A multi-agent framework for dynamic llm evaluation. *arXiv preprint arXiv:2402.11443*, 2024.
- [115] Y. Wang, H. Li, X. Han, P. Nakov, and T. Baldwin. Do-not-answer: A dataset for evaluating safeguards in llms. *arXiv preprint arXiv:2308.13387*, 2023.
- [116] M. Wu and A. F. Aji. Style over substance: Evaluation biases for large language models. *arXiv preprint arXiv:2307.03025*, 2023.
- [117] T. Xie, X. Qi, Y. Zeng, Y. Huang, U. M. Schwag, K. Huang, L. He, B. Wei, D. Li, Y. Sheng, et al. Sorry-bench: Systematically evaluating large language model safety refusal behaviors. *arXiv preprint arXiv:2406.14598*, 2024.
- [118] S. Yang, W.-L. Chiang, L. Zheng, J. E. Gonzalez, and I. Stoica. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*, 2023.
- [119] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380. Association for Computational Linguistics, 2018.
- [120] Y. Yao, A. Zhang, Z. Zhang, Z. Liu, T.-S. Chua, and M. Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models, 2022.
- [121] Z.-X. Yong, C. Menghini, and S. H. Bach. Low-resource languages jailbreak gpt-4. *arXiv preprint arXiv:2310.02446*, 2023.
- [122] X. Yuan, J. Li, D. Wang, Y. Chen, X. Mao, L. Huang, H. Xue, W. Wang, K. Ren, and J. Wang. S-eval: Automatic and adaptive test generation for benchmarking safety evaluation of large language models. *arXiv preprint arXiv:2405.14191*, 2024.
- [123] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

- [124] M. Yüksekönül, V. Chandrasekaran, E. Jones, S. Gunasekar, R. Naik, H. Palangi, E. Kamar, and B. Nushi. Attention satisfies: A constraint-satisfaction lens on factual errors of language models. In *International Conference on Learning Representations*, 2024.
- [125] A. K. Zhang, N. Perry, R. Dulepet, E. Jones, J. W. Lin, J. Ji, C. Menders, G. Hussein, S. Liu, D. Jasper, et al. Cybench: A framework for evaluating cybersecurity capabilities and risk of language models. *arXiv preprint arXiv:2408.08926*, 2024.
- [126] G. Zhang and M. Hardt. Inherent trade-offs between diversity and stability in multi-task benchmark. *arXiv preprint arXiv:2405.01719*, 2024.
- [127] H. Zhang, J. Da, D. Lee, V. Robinson, C. Wu, W. Song, T. Zhao, P. Raja, D. Slack, Q. Lyu, et al. A careful examination of large language model performance on grade school arithmetic. *arXiv preprint arXiv:2405.00332*, 2024.
- [128] W. Zhang, M. Aljunied, C. Gao, Y. K. Chia, and L. Bing. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. *Advances in Neural Information Processing Systems*, 36:5484–5505, 2023.
- [129] Y. Zhang, K. Zhou, and Z. Liu. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022.
- [130] H. S. Zheng, S. Mishra, H. Zhang, X. Chen, M. Chen, A. Nova, L. Hou, H.-T. Cheng, Q. V. Le, E. H. Chi, et al. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*, 2024.
- [131] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.
- [132] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [133] J. Zhou, T. Lu, S. Mishra, S. Brahma, S. Basu, Y. Luan, D. Zhou, and L. Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.
- [134] K. Zhu, J. Chen, J. Wang, N. Z. Gong, D. Yang, and X. Xie. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *International Conference on Learning Representations*, 2024.
- [135] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, 2015.
- [136] Z. Zong, G. Song, and Y. Liu. Detrs with collaborative hybrid assignments training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6725–6735. IEEE Computer Society, 2023.