

UC Santa Barbara

UC Santa Barbara Electronic Theses and Dissertations

Title

Advancing AI Understanding in Language & Vision

Permalink

<https://escholarship.org/uc/item/56h8g5zr>

Author

Sharma, Aditya

Publication Date

2024

Peer reviewed|Thesis/dissertation

University of California
Santa Barbara

Advancing AI Understanding in Language & Vision

A thesis submitted in partial satisfaction
of the requirements for the degree

Masters of Science
in
Computer Science

by

Aditya Sharma

Committee in charge:

Professor William Yang Wang, Co-Chair
Professor Tobias Höllerer, Co-Chair
Professor Xifeng Yan

June 2024

The Thesis of Aditya Sharma is approved.

Professor Xifeng Yan

Professor Tobias Höllerer, Committee Co-Chair

Professor William Yang Wang, Committee Co-Chair

June 2024

Advancing AI Understanding in Language & Vision

Copyright © 2024

by

Aditya Sharma

To my family

Acknowledgements

I would like to express my gratitude to all the individuals I have had the privilege of working with during the past four years at UC Santa Barbara. I had the opportunity to collaborate with a remarkable group of individuals, including **Justin Chang, Matthew Ho, Tobias Höllerer, Sharon Levy, Yujie Lu, Michael Saxon, William Wang, and Luke Yoffe** (in alphabetical order). Their contributions have greatly impacted the work presented in this thesis.

I fondly remember beginning my college journey in the College of Creative Studies Computing Program, where I was fortunate enough to receive invaluable mentorship from **Phillip Conrad and Richert Wang**, who guided me in course selection, research, and career planning. I was fortunate to meet some of the most hardworking, brightest, and like-minded peers in my Computing cohort, who encouraged me to take on challenging coursework while maintaining camaraderie. I would also like to thank **Diba Mirza and Chinmay Sonar** for advising the Early Research Scholars Program and instilling confidence in our team of undergraduate researchers who were passionate about starting their research journey.

I extend a special thanks to my advisor, **William Wang**, whose unwavering support and encouragement allowed me to explore my research interests. His thoughtful feedback and vision motivated me to work on projects that challenged the boundaries of research. His upper-division Machine Learning and Natural Language Processing courses supplemented my research and further established my research directions. I would also like to express my gratitude to my co-chair, **Tobias Höllerer**, for his guidance and insightful advice. His approachability and passion for education were consistently evident when I served as his Teaching Assistant and took his graduate class in Visual Computing and Extended Reality. Finally, I would like to thank my committee member, **Xifeng Yan**, for

everything I learned in his special topics graduate course in Conversational AI. Thanks to my committee members for their valuable suggestions to this thesis, it would not have been possible without their collaboration.

In particular, I cherish the wonderful memories with my Early Research Scholar teammates, **Matthew Ho and Justin Chang**. They were self-driven, motivated, and determined to make an impact in the field. I spent countless hours with them, bouncing off each other's ideas, and I was deeply impressed by their commitment. Similarly, I collaborated with **Luke Yoffe**, freely exchanging ideas and knowledge with the shared goal of continuous improvement. I would like to acknowledge my mentor, **Michael Saxon**, for his firm support in addressing research challenges and fostering my personal growth as a researcher.

I dedicate this thesis to my family — my parents and my sister — who have constantly supported me in my educational and research pursuits. Their influence has played such a meaningful role in shaping my journey. I owe everything I am today to them, and I am profoundly grateful.

Each chapter of this thesis includes additional acknowledgements, expressing my sincere appreciation to individuals who provided invaluable feedback, as well as to the funding that supported my research.

Curriculum Vitæ

Aditya Sharma

Phone: (408) 203-1217
Email: aditya_sharma@cs.ucsb.edu
Website: asharma381.github.io
LinkedIn: linkedin.com/in/adityas17
GitHub: github.com/asharma381

Education

- 03/2023–06/2024 **M.S. in Computer Science** – Natural Language Processing
Department of Computer Science, College of Engineering
University of California, Santa Barbara
B.S./M.S. Student of the Year
4.00 GPA
Thesis: “Advancing AI Understanding in Language & Vision”
Advisor: William Yang Wang
Committee Members: Tobias Höllerer, Xifeng Yan
- 09/2020–03/2023 **B.S. in Computer Science**
Computing Program, College of Creative Studies
University of California, Santa Barbara
Summa Cum Laude (Highest Honors)

Industry Experience

- 06/2023–09/2023 **Google** — Cloud Workspace, Gmail
Software Engineering Intern (Sunnyvale, CA)
- Led the Gmail iOS migration of native iOS components to adhere to Google Material 3 (GM3) design standards using Objective-C. Successfully launched the GenAI features to production, directly impacting billions of users worldwide.
 - Incoming Software Engineer at Google in Mountain View, CA
- 06/2022–09/2022 **Google** — Assistant, Text-to-Speech NLP Infrastructure
STEP Intern (Mountain View, CA)
- Developed end-to-end support for a new telephone verbalization style, **zero-as-zero**, using C++. Implemented changes in the Text-to-Speech (TTS) engine by creating a finite-state grammar and propagating it into SSML (Speech Synthesis Markup Language).

- 06/2021–09/2021 **Google** — Google Fi
STEP Intern (Remote, CA)
- Implemented a number porting tracker feature for the Google Fi iOS app using Objective-C, enabling new users to track the status of their phone number during carrier switches. Leveraged protocol buffers and RPC for efficient data transfer. Successfully deployed the feature to production.
- 06/2018–08/2018 **Authentic8, Inc**
Software IT Intern (Redwood City, CA)
- Managed IT assets by writing scripts using the Authentic8 Silo browser to help achieve FedRAMP Certification from the US Department of Homeland Security (DHS).

Research Experience

- 09/21 – 06/24 **UCSB Natural Language Processing (NLP) Group**
Graduate & Undergrad Student Researcher (Santa Barbara, CA)
- Advised by Professor William Wang as an early research scholar. Published ICLR 2023 Oral Paper: Top 5% out of all 4019 submissions.
- 03/23 – 03/24 **UCSB Four Eyes Lab**
Graduate Student Researcher (Santa Barbara, CA)
- Advised by Professor Tobias Höllerer. Published papers to IEEE AIXVR 2024 and IEEE ISMAR 2023.
- 06/19 – 09/20 **Santa Clara University** — Mobile Computing Laboratory
Researcher Assistant (Santa Clara, CA)
- Collaborated with Dr. Silvia Figueira. Published Pin&Post, an iOS mobile app targeted for residents to report electrical fire hazards by profiling unconstrained vegetation, to IEEE GHTC 2020.

Preprints

1. **Aditya Sharma***, Michael Saxon*, William Yang Wang. “Losing Visual Needles in Image Haystacks: Vision Language Models are Easily Distracted in Short and Long Contexts”. Preprint. June 2024. (Under review at EMNLP 2024).
2. Michael Saxon*, Fatima Jahara*, Mahsa Khoshnoodi*, Yujie Lu, **Aditya Sharma**, William Yang Wang. “Who Evaluates the Evaluations? Objectively Scoring Text-to-Image Prompt Coherence Metrics with T2IScoreScore (TS2)”. Preprint. April 2024. (Under review at NeurIPS 2024).

Publications

1. **Aditya Sharma***, Luke Yoffe* and Tobias Höllerer. “OCTO+: A Suite for Automatic Open-Vocabulary Object Placement in Mixed Reality”. In 2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR), Oral Paper: **Special Session XR for AI**, Los Angeles, CA, USA, 2024.
2. **Aditya Sharma***, Luke Yoffe*, and Tobias Höllerer. “OCTOPUS: Open-vocabulary Content Tracking and Object Placement Using Semantic Understanding in Mixed Reality”. In 2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Sydney, Australia, 2023.
3. **Aditya Sharma***, Matthew Ho*, Justin Chang*, Michael Saxon, Sharon Levy, Yujie Lu and William Yang Wang. “WikiWhy: Answering and Explaining Cause-and-Effect Questions”. In Proceedings of Eleventh International Conference on Learning Representations (**ICLR 2023**), Oral Paper: **Top 5% out of all 4019 submissions**, Kigali, Rwanda, May 1st to 5th.
4. **Aditya Sharma**, Ishan Goyal and Silvia Figueira. “PinPost - App-Based Reporting of Electrical Fire Hazards to Prevent Wildfires”. In 2020 IEEE Global Humanitarian Technology Conference (GHTC), Seattle, WA, USA, 2020.

Honors and Awards

- **B.S./M.S. Student of the Year**, The Caitlin Scarberry Memorial Award, UCSB Computer Science, 2024
- **UCSB Board of Trustees Speaker**, Presented with the College of Creative Studies Interim Dean, Tim Sherwood at the Board of Trustees Spring Meeting, UC Santa Barbara Foundation, 2023
- **Undergraduate Research Fellowship Award**, Traveling Undergraduate Research Fellowship (TURF), College of Creative Studies, UC Santa Barbara, 2023
- **UCSB Early Research Scholar**, ERSP Program, UCSB Computer Science, 2022
- **Top 2 Best in Show** at Google Cloud Demo Week, from all hackathon teams worldwide
- **Best AR/VR Hack**, Won Oculus Quest 2 VR Headset, Hack the Northeast, 2021
- **Google Cloud COVID-19 Research Grant & Fund Finalist**, Awarded \$5,000 research credits for open-source location based contract tracing app HotSpot.
- **Wolfram Award Winner**, Top 30 Hack of 1,200, Shell Hacks, 2020
- **ACM 2nd Place Student Award**, Association of Computing Machinery, 2020
- **IEEE Best Electro-Technology Prize**, Institute of Electrical and Electronics Engineers, 2020
- **Trimble Award Winner**, Trimble Inc, Fortune 750 Company

- **1st Award, Physical Science & Engineering**, Most Awarded Project, Synopsys Science Fair, Santa Clara Valley Science & Engineering Fair, 2020
- **Best COVID-19 Hack**, Top 30 Hack, MHacks, 2020

Coursework

- **Relevant Coursework:** Algorithms, Natural Language Processing, Conversational AI, Computer Vision, Machine Learning, Artificial Intelligence, Mixed and Augmented Reality, Trustworthy ML in Security, Neural Information Retrieval
- **Additional Coursework:** Data Structures, Compilers, Computer Networks, Virtual Machines, Operating Systems, Computer Architecture, Automata Theory
- **Teaching Assistant:** UCSB CS 130A, Data Structure & Algorithms, Spring 2024
- **Teaching Assistant:** UCSB CS 148, Computer Science Project, Winter 2024

Abstract

Advancing AI Understanding in Language & Vision

by

Aditya Sharma


Large Language Models (LLMs) have emerged as a powerful tool, demonstrating impressive capabilities in natural language generation. These pre-trained models consistently outperform benchmarks across a wide range of multi-modal tasks. However, this raises a crucial question: Do LLMs truly understand and reason about the information they process, or are they simply advanced pattern recognizers? This thesis investigates the reasoning and understanding capabilities of language models, aiming to develop more context-aware and intelligent AI systems. Firstly, we introduce WikiWhy, a benchmark designed to evaluate the reasoning capabilities of LLMs in answering and explaining cause-and-effect questions. Next, we present OCTO+, a state-of-the-art suite for automatic object placement in augmented reality, which leverages open-vocabulary Vision Language Models (VLMs) to integrate virtual content seamlessly. Finally, we propose the Visual Needle in a Haystack framework, which assesses the performance of VLMs in long-context reasoning and highlights their challenges with distractor images. By addressing the limitations in long-context reasoning and promoting interpretability, this thesis seeks to unlock the full potential of LLMs and VLMs, enabling them to truly understand and reason about the world.

Contents

Curriculum Vitae	vii
Abstract	xi
List of Figures	xiv
List of Tables	xvii
1 Introduction	1
2 Benchmarking LLM Reasoning	4
2.1 Introduction	4
2.2 Related Work	6
2.3 Background	9
2.4 Dataset	12
2.5 Experiments	16
2.6 Conclusion	25
2.7 Ethics Statement	26
2.8 Acknowledgements	26
3 Scene Understanding with LLMs	28
3.1 Introduction	28
3.2 Related Work	30
3.3 Method	32
3.4 Evaluation and Results	40
3.5 Discussion and Limitations	53
3.6 Application	54
3.7 Conclusion	56
3.8 Acknowledgements	56

4	Long-Context VLM Reasoning	58
4.1	Introduction	58
4.2	Related Work	60
4.3	LoCoVQA Generation Method	61
4.4	Experiments	66
4.5	Results	67
4.6	Ablation: Needle in a Haystack	71
4.7	Discussion	73
4.8	Conclusion	75
4.9	Limitations	76
4.10	Acknowledgements	76
A	Appendix	77
A.1	Appendix: Benchmark for LLM Reasoning	77
A.2	Appendix: Scene Understanding with LLMs	84
A.3	Appendix: Long-Context VLM Reasoning	86
	Bibliography	92

List of Figures







2.1	A simple example of an entry from WIKIWHY; a cause and effect sourced from a Wikipedia passage , a “why” question and its answer about this relation, and most importantly rationale that explains why cause leads to effect	5
2.2	Where and Why QA Pair Example	9
2.3	Explanation topologies in WIKIWHY mainly vary between a sequence of intermediate conclusions (chain-like) and a set of rationale that combine with the original cause to entail the final effect.	10
2.4	Dataset Collection and Validation Pipeline	12
2.5	Rationale Length Distribution	16
2.6	Alignment example for sentence-level metrics. Ordered evaluation uses the longest common subsequence as shown by alignment 1 and 2. The final alignment’s length is used to compute F-score metrics.	19
2.7	Panel Evaluation Criteria analyzing the Similarity & Correctness to assign a binary yes/no rating.	20
3.1	Overview of various methods we experimented with, including OCTO+  . To determine where a cupcake should be placed, we perform three stages: 1) image understanding : generate a list of all objects in the image (OCTO+ uses RAM++); 2) reasoning : select the most natural object with GPT-4; 3) locating : locate the 2D coordinate of the selected object in the image and ray cast the 2D coordinate to determine the 3D location in the AR scene.	32
3.2	Large Multimodal Model (LMM) prompt for Stage 1 – Image Understanding	35
3.3	Large Language Model (LLM) prompt for Stage 2 – Reasoning. The square brackets denote additional or customized information, omitted here for brevity and generality.	36
3.4	Large Multimodal Model (LMM) prompt for Stage 2 – Reasoning.	37
3.5	Large Multimodal Model (LMM) prompt for Stage 3 – Locating.	39
3.6	Comparative placements for one input image, and the prompt “Apple”.	41

3.7	PEARL -Score for one placement. The cases clearly define the importance of the individual terms in the PEARL -Score formula. The green box defines if the point is located in the mask. The yellow box minimizes over the points with respect to the mask. The red box calculates the euclidean distance.	48
3.8	Left: The 2D location in the image selected. Right: A screenshot of the 3D scene with a virtual cupcake placed on the plate. Both the 2D and 3D locations were found as described in the Locating section.	53
3.9	Comparison of PEARL Metrics alignment with Human preferences. PEARL -Score and IN MASK score are well aligned to human score, as observed by the strong positive correlation between automated metrics and human scores.	55
4.1	The impact of visual context on vision-language models (VLMs) in our modified, multi-image versions of the OK-VQA, MMStar, and MMBench evaluation benchmarks. <i>Distractor images</i> around the target image increase the <i>visual context length</i> needed to answer the questions. VLM performance exhibits an <i>exponential decay</i> as distractors increase, evident in both single composed (cmp) and multiple interleaved (int) image configurations.	59
4.2	Example of Image (X) corresponding to question-answer pair (Q , A) under increasing visual context lengths in the composed setting. <i>The green box is for illustration purposes; not included in model inputs.</i>	63
4.3	Each subfigure represents a variable number of MNIST digits (1, 4, 8) while maintaining a context length of 9 images.	66
4.4	VLM Performance on MMStar and OK-VQA. Note the clearly declining exponential fit trends for many of the models. The (model, task) pairs for which these trends do not hold by and large are <i>below the random baseline</i>	68
4.5	Analyzing GPT-4V performance across 8 multi-modal benchmarks with varied visual context lengths.	69
4.6	VLM Performance on the MNIST-Digits transcription task as a function of # of digits to transcribe. <i>These plots have a different x-axis than the plots in Figure 1 and Figure 4.4:</i> rather than the relationship between context size and performance, we are assessing the relationship between "task difficulty" and performance, at four context sizes.	70

4.7	Evaluation of the Visual Needle in a Haystack task using GPT-4V, best-performing VLM, conducted under both composed and interleaved haystack settings. Retrieval accuracy measures the frequency of correct answers produced by the VLM, in this case identifying the MNIST digit. In the composed setting, retrieval accuracy is measured by placing the needle at each cell. In the interleaved setting, needle depth signifies the position within the image sequence, with 0% representing the first image and 100% representing the last. Our evaluation highlights a consistent decline in retrieval accuracy as the visual context length increases.	72
A.1	GPT-3 Few-shot Exemplars	80
A.2	Amazon Mechanical Turk Interface for Stage 1	81
A.3	Amazon Mechanical Turk Interface for Stage 2	82
A.4	Amazon Mechanical Turk (MTurk) Interface for human evaluation	84
A.5	Collision Filtering Method for by prompting LLM with query Q to identify entities. Cell with ■ represents the entities for each image X_j . If there are no entities in common, there are no collisions so we mark cell with ■ indicating this is a valid construction of images for	88

List of Tables

2.1	A comparison of WIKIWHY with previous QA datasets relating to explanation.	8
2.2	Examples from 6 most frequent topics covered in WIKIWHY. c denotes cause, e effect, and s_i the i th rationale sentence.	14
2.3	position=top	15
2.4	position=top	16
2.5	Baseline Performance on Explanation Tasks (EO = Explanation-Only, A&E: Answer and Explanation). For Task 3, the Single Model setting has the generative model complete the end-to-end task in a single pass. The Pipeline setting allows each stage to be handled separately (QA is handled by BM25+FiD and explanation is done by GPT-3). Human evaluation was done with on a binary scale (correct/incorrect) and we report the proportion of correct evaluations.	24
2.6	Human evaluation. Overall correctness is marked on a binary scale— an explanation is complete and satisfying or not. Concision penalizes for repeated or unnecessary information, fluency evaluates grammar, and validity measures if the generated sequence makes logical sense regardless if it correctly explains the relation. W refers to Win, T refers to Tie, and L refers to Lose. For Win/Lose/Tie, annotators compared the generations against WIKIWHY’s gold references.	25
3.1	Properties of Closed-Vocabulary and Open-Vocabulary Models	29
3.2	Stage 1 Metrics. Best IN BOLD , second best <u>UNDERLINED</u>	43
3.3	Stage 2 Metrics. Best IN BOLD , second best <u>UNDERLINED</u>	45
3.4	Human Evaluation is performed in the following methods	50

3.5	Stage 3 Automated Metrics. The 3 baselines (natural, random, unnatural) denote the placements. InstructPix2Pix is abbreviated as InsPix2Pix for brevity. For locators column, (max) denotes selecting the pixel with the maximum intensity, (center) denotes the center of the bounding box, (bottom) denotes the bottom center of the bounding box, (pixel) denotes the pixel location was provided by GPT-4V directly. The best metric in each category is IN BOLD and the second best is <u>UNDERLINED</u> .  is the OCTO+ method and  is OCTOPUS.	51
3.6	Stage 3 Human Evaluation Metrics. The 3 baselines (natural, random, unnatural) denote the placements. For models, we select the 5 best performing methods from Table 3.5 results. The best metrics in each category is IN BOLD and the second best is <u>UNDERLINED</u> .  is OCTO+,  is OCTOPUS	52
4.1	Overview of  open-source and  proprietary vision-language models (VLMs).	67
A.1	Explanation Evaluation Results of WIKIWHY dataset according to the following metrics: SacreBLEU (S-BLEU) [1], Word-Mover’s distance (WMD) [2, 3], Sentence Mover’s Similarity Metrics (SMS) [4], BERT-f1 Score [5], ROUGE-1 (abbreviated as ROU.-1), ROUGE-2 (abbreviated as ROU.-2), and ROUGE-L (abbreviated as ROU.-L for brevity) (all ROUGE-f1 Scores [6] averaged). SMS is scaled by 1000 for readability.	78
A.2	GPT-3 explanation results with various input settings: Ideal- gold cause/answer, Well-Selected- provided cause/answer predicted by best-performing reader model (FiD), End-to-end- provided only question/effect (GPT-3 completes end-to-end task)	79
A.3	position=top	79
A.4	Answer Evaluation Results for WIKIWHY dataset. Stage 1: RoBERTa , BigBird , and FiD . FiD Gold is fine-tuned on 80% train split & evaluated on 10% dev split.	83
A.5	Explanation performance (unordered f1) over the most frequent topics. We GPT-2 under the greedy setting and GPT-3 under the same defaults as Table 2.5	83
A.6	Comparison in Runtime Analysis of OCTO+ with respect to other vision-language methods for content placement	85
A.7	Logarithmic curve fit r^2 values are reported for each open-weight model, followed by a symbol denoting the p -value for statistical significance. The symbols represent: * for $p \leq 0.05$, ** for $p \leq 0.01$, & *** for $p \leq 0.001$. Light pink cells represent correlations for sets of values that fall below chance performance.	91

A.8 Logarithmic curve fit r^2 values are reported for each closed-source model, followed by a symbol denoting the p -value for statistical significance. The symbols represent: * for $p \leq 0.05$, ** for $p \leq 0.01$, & *** for $p \leq 0.001$. Light pink cells represent correlations for sets of values that fall below chance performance. 91

List of Algorithms

1	Stage 1 Metrics Computation	42
---	---------------------------------------	----

Chapter 1

Introduction

Recent breakthroughs in language and vision research, particularly in multi-modal understanding, are paving the way for more robust Artificial Intelligence applications.

The origins of modern advancements in language and vision can be traced back to the introduction of the Transformer architecture. In 2017, a group of researchers at Google Brain published “Attention is All You Need,” presenting the Transformer [7], a sequence-to-sequence model capable of taking text as input and producing text as output. Unlike the predecessors bi-directional RNNs [8], LSTMs [9], GRUs [10], and ResNet [11], which suffer from vanishing gradients and were limited to sequential operations, Transformers rely solely on a self-attention mechanism using query, key, and value vectors making them highly parallelizable. The Transformer follows the encoder-decoder architecture. BERT [12] uses the encoder-part of the transformer to capture the underlying semantic and syntactic language information for natural language understanding. GPT [13] uses the decoder-part of the transformer to perform natural language generation.

Fueled by the success of transformers, researchers explored extending them beyond language to the vision domain. Vision Transformers (ViT) [14] achieve this by splitting an image into patches, flattening them, and feeding them into a Transformer. This

approach allows the Vision Transformer to capture complex relationships within the image, outperforming Convolutional Neural Network (CNN) [15] based architectures.

Following the advancements in Vision Transformers (ViTs) for image processing, researchers aimed to bridge the gap between language and vision. Contrastive Language-Image Pre-training (CLIP) [16] emerged as the method for aligning text and image representations. During training, CLIP learns from massive datasets of text-image pairs found on the internet. By using contrastive learning, it maximizes the probability between the image and text representations. This ability to learn relationships between text and images allows CLIP to perform zero-shot predictions, generalizing to unseen image and text combinations.

At the crossroads of language and vision research, we arrive at Vision-Language Models (VLMs). These impressive models are an extension to Large Language Models (LLMs) which can process inputs from both modalities (text and image) and generate natural language as output. Popular foundational VLMs include GPT-4o [17], GPT-4 Vision [18], Gemini 1.5 [19], Claude 3 Opus [20], PaliGemma [21], LLaVA [22], Qwen-VL [23], and Mantis [24]. In fact, VLMs are rapidly evolving, with new models emerging at an astonishing pace.

While large language models (LLMs) showcase impressive generation capabilities, a critical question remains: **Do these models truly understand the information they process?** Recent research suggests that LLMs may possess emergent abilities [25]. One such ability is the capability to explain its reasoning through Chain of Thought (CoT) prompting [26]. Chain of Thought prompting is a technique that involves providing the model with intermediate steps in the form of explanations or “think step-by-step” instructions. This approach, particularly effective in zero-shot setting, has led to improved performance across benchmarks.

In reality, however, LLMs are black-boxes, there is little insight into what input will

trigger the parameters inside these models to produce the answers it does. Moreover, LLMs store substantial amounts of knowledge implicitly in their parameters. In fact it is estimated that the text training data fed into LLMs alone would take 20,000 years for humans to read, this scale of data causes the models to generate output, but humans learn differently and researchers should consider brainstorming newer innovation in architectural designs instead of leaderboard climbing to score a few points higher than state-of-the-art. Focusing solely on leaderboard climbing and achieving marginal performance gains may be less productive than exploring new model architectures that prioritize interpretability and understanding.

In this thesis, we will explore the reasoning and understanding of LLMs in the the language and vision domain. In Chapter 2, we will benchmark LLM reasoning with WikiWhy. WikiWhy is a novel dataset of cause-and-effect questions curated to evaluate models on their ability to reason between cause and effect. In Chapter 3, we address the problem of content placement in augmented reality (AR) scenes. We introduce OCTO+, a state-of-the-art pipeline for automatic virtual content placement. This approach leverages the reasoning capabilities of LLMs and VLMs to understand the scene and determine suitable locations for placing objects. We conceptualize the placement problem into three stages. In Chapter 4, we discuss the challenges faced by VLMs in reasoning over long contexts. A crucial ability for effective long-context reasoning is the identification of relevant information within an extensive input. To assess VLM capabilities in this domain, we propose the “Visual Needle in a Haystack” task. This stress-test involves adding visual context distractor images to an existing multi-modal benchmark. We introduce a benchmark generating process specifically designed to test these capabilities in VLMs. Our findings reveal a clear trend: all popular VLMs exhibit a exponential decay in performance as the visual context length increases.

Chapter 2

Benchmarking LLM Reasoning

2.1 Introduction

Error analyses of practical NLP systems in recent history demonstrate that some of the mistakes made by state-of-the-art models would be avoided by basic human intuition [27], and some of the most challenging tasks for models are the same ones that might be trivial to human children. With modern systems’ impressive performance on tasks such as grammar correction showing that manipulating language is not the issue, LLMs seem to face a fundamental lack of common sense— an understanding of everyday phenomena and how they interact with each other and the world at large. As striking gains in subjective performance on summarization, creative text generation, and apparent language understanding continue to be called into question, the development of strong benchmarks to assess reasoning capabilities for these LLMs grows more important.

One popular approach to measuring reasoning capability is through performance on question answering (QA) benchmark tasks where direct queries for information act as a straightforward examination of a system’s “*understanding*.” Classic QA datasets, however, are primarily concerned with retrieving factoids to answer questions of “Who”,

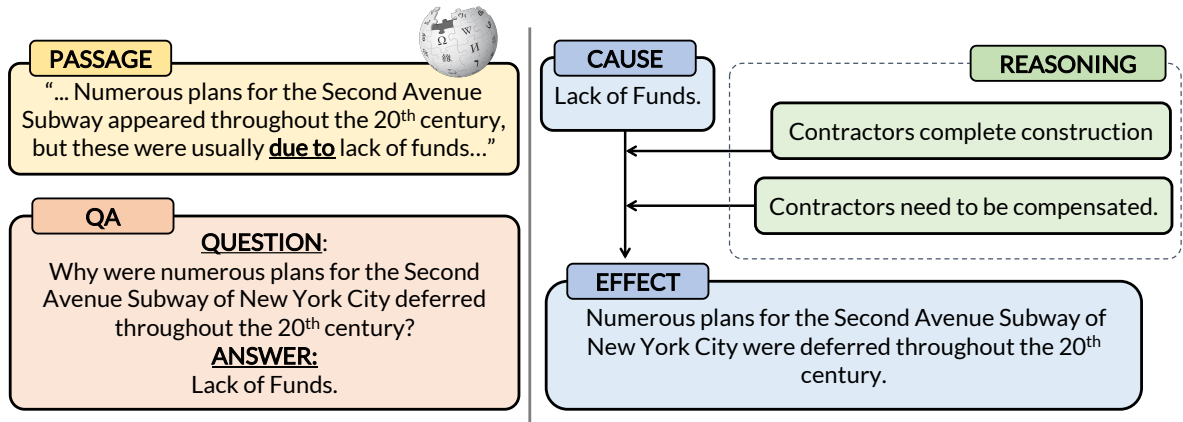


Figure 2.1: A simple example of an entry from WIKIWHY; a **cause** and **effect** sourced from a Wikipedia **passage**, a “why” **question** and its **answer** about this relation, and most importantly **rationale** that explains why **cause** leads to **effect**.

“What”, “When”, and “Where”. These questions have been shown to be answerable (with high accuracy) by simple pattern-matching approaches [28], thereby limiting their ability to measure the aforementioned reasoning capability. Looking to maintain the breadth of topics covered while increasing the difficulty of the QA task, researchers introduced multi-hop QA datasets like HotpotQA [29]. While challenging, the task’s extra complexity mostly leads to unnatural questions that can be addressed with iterated factoid retrieval and entity resolution, rather than a necessary understanding of how different entities interact. Noticeably absent in these prior datasets are “*why*” questions, which prompt for not factoids, but explanations– reasoning made explicit.

The task of explanation uses reasoning and produces explicit, interpretable *thought* processes. Capitalizing on these properties, this chapter introduces WIKIWHY, a novel dataset containing “why” question-answer pairs. Each WIKIWHY entry contains a rationale explaining the QA pair’s causal relation (Figure 2.1), summing to a total of 14,238 explanation elements. In the context of recent multimodal, self-supervised approaches aiming to capture intuitions unlearnable from text alone [30], WIKIWHY presents an

opportunity to investigate a specific kind of information absent in text: implicit commonsense assumptions. Compared to other QA datasets with rationales, WIKIWHY covers a significantly broader range of 11 topics which may prove valuable for developing the skill of applied reasoning on various specific situations.

Our experiments in explanation generation and human evaluation demonstrate that state-of-the-art generative models struggle with producing satisfying explanations for WIKIWHY cause-effect relations. Our experiments also demonstrate how our proposed task might be used to diagnose a lack of *understanding* in certain relations.

Our key contributions are thus:

- Explanation **within** cause-effect relations as a novel problem formulation for exploring LLM reasoning ability.
- WIKIWHY, the first question-answering dataset focusing on reasoning **within** causal relations, spanning **11 topics**.
- Experiments on state-of-the-art, generative models to investigate various settings and establish baseline results with sizable room for improvement.
- Introduce idea-level evaluation metrics for free-form text (explanation) generation and a human judgment correlation analysis, demonstrating that:
 - Reference similarity is strongly correlated with explanation correctness
 - Metrics introduced correlate with this proxy.

2.2 Related Work

Cause and Effect Causality has been a subject of rigorous work in various fields. In science philosophy, Pearl [31] has contributed seminal work relating to causal models,

Bayesian networks, and causal strength via interventions and counterfactuals. These ideas have even been incorporated into QA tasks through Knowledge Graph approaches, such as filtering spurious latent correlations (Sui et al., 2022) [32]. While our work emphasizes cause-and-effect, we are unconcerned with causal strength as we begin with Wikipedia-grounded relations and are interested in the information encoded into LLMs rather than augmented structures such as knowledge graphs.

Multi-hop Question Answering While datasets such as HotpotQA [29] and HybridQA [33] are instrumental in gauging models’ ability to handle multiple sources and modalities, they are focused on iterated factoid retrieval. Although chaining multiple facts into a multi-hop answer is useful for products, WIKIWHY focuses on *in-filling* rationales to demonstrate reasoning.

Visual Question Answering Vision and language tasks have also intersected with both QA and reasoning. The Visual Question Answering (VQA) dataset [34] prompts textual answers to questions about images. However, the caption-based generation leads to surface-level questions that require little reasoning ability, and the multiple-choice output format precludes explicit reasoning. The vision-based Sherlock dataset [35] is much closer to our work, focusing on abductive reasoning (working backward from a consequence). Setting aside modality differences, WIKIWHY requires deeper reasoning with its multi-hop explanations.

Explainable Question Answering One previous approach to building explanation resources collects direct answers to “why” questions. TellMeWhy [36] features question-answer pairs tied to short story narrative contexts. The dataset skips step-wise explanations, prioritizing reading comprehension instead. On the other hand, ELI5 [37] dives deep into reasoning with long-form, detailed explanations. However, the open-endedness

Dataset	Size	Answer Type	Explanation Type	Topics	Source
CoS-E ¹	9,500	MCQ	1-step	1	ConceptNet
eQASC ²	9,980	MCQ	2-step	1	WorldTree
CausalQA ³	24,000	Short	None	1	Yahoo Fin.
EntailmentBank ⁴	1,840	Short	Tree	1	WorldTree
WIKIWHY	9,406	Short	Set/Chain	11	Wikipedia

¹(Rajani et al., 2019) [38], ²(Jhamtani & Clark, 2020) [39], ³(Yang et al., 2022) [40], ⁴(Dalvi et al., 2021) [41]

Table 2.1: A comparison of WIKIWHY with previous QA datasets relating to explanation.

(compared to explaining a specific cause-effect relation) complicates evaluating candidate responses.

Another line of QA work emphasizes a rationale component as support for answer predictions. Datasets like CoS-E [38], eQASC [39], and ENTAILMENTBANK [41] focus on explanation and reasoning much like WIKIWHY, albeit with significant differences (Table 2.1). CoS-E’s explanations for CommonsenseQA [42] mark an important first step, but the commonsense explanations have limited depth, often requiring a single hop of reasoning. eQASC and ENTAILMENTBANK feature richer explanations with more complex structure, tightly focusing on grade school level science facts. Regarding structure, fixed-length rationale in CoS-E [38], eQASC [39], FEVER [43], and e-SNLI [44] capture less granularity, while entailment trees accept limitations in scale and naturalness in exchange for complete ordering information. Previous datasets tend towards retrieval tasks with eQASC’s corpus of all rationale sentences and EntailmentBank’s collection of root causes. Retrieval enables simple evaluation, at the cost of decreased difficulty, the possibility for exploiting spurious artifacts, and reduced debugging opportunity.

Where and Why QA Example	
Q: Where do the Tigris and Euphrates rivers meet?	A: The Persian Gulf.

Q: Why are precipitation levels falling in the Tigris and Euphrates river basin?	A: Climate Change.

Figure 2.2: Where and Why QA Pair Example

2.3 Background

2.3.1 Why focus on “Why” Questions?

“Why” questions are underrepresented in other QA datasets. Users tend to ask straightforward questions that use words like “who”, “what”, “when” or “where.” Questions of this more common form have simple answers that state standalone facts which may be elaborated but do not require explanation. Consider the pair, “Q: Where do the Tigris and Euphrates rivers meet? A: The Persian Gulf” (Figure 2.2). The answer is straightforward. In contrast, a “why” QA-pair encodes a cause-effect relation. Take, for example, “Q: Why are precipitation levels falling in the Tigris and Euphrates river basin? A: Climate Change.” This pair encodes the causal relation “Climate change is reducing the amount of precipitation in the Tigris and Euphrates river basin” (Figure 2.3). The answer to a “why”-question is an explanation itself (climate change explains reduced precipitation), but we can take it a step further and ask “why” *again* to request the understanding or intuition of this process. While there are some processes at the edge of human understanding or taken as axioms, we assert that there are valid explanations for most processes due to the layered nature of human understanding. This extra step is especially worth taking since it allows WIKIWHY to not only test if a model “knows” that “climate change causes reduced precipitation“ but also if it “understands” the underlying

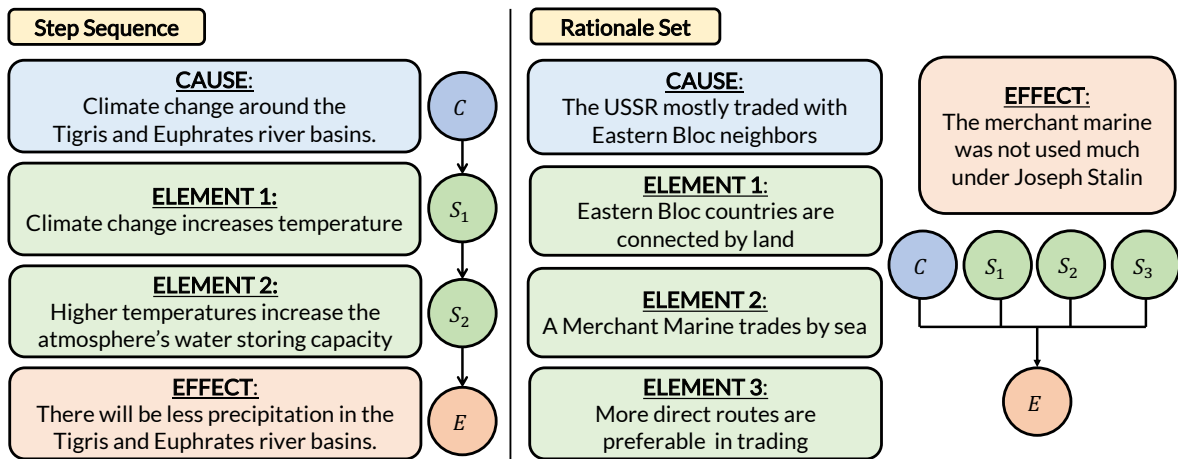


Figure 2.3: Explanation topologies in WIKIWHY mainly vary between a sequence of intermediate conclusions (chain-like) and a set of rationale that combine with the original cause to entail the final effect.

mechanics of why that is the case.

2.3.2 Task Formulation

Formally defined in §2.5, we propose a **generative** explanation task. Previous works have made strides in assessing reasoning through multiple choice [45], retrieval [46], and partial generation [41]. While these works are undoubtedly crucial towards the end goal of understanding and reasoning, their task formulations have some drawbacks. Referring back to education, studies on human students have shown that multiple choice questions “obscure nuance in student thinking” [47]. Likewise, a selection decision can be correct for retriever systems but for the wrong reasons. Augmenting multi-hop factoid questions with an additional task of selecting the relevant supporting facts from the context passage, $\mathcal{R}^4\mathcal{C}$ [48] emphasizes that interpretability is lost in the absence of explanation. Furthermore, text generation to combine existing ideas is arguably a different task than generating from scratch. The field of psychology defines recall (mental retrieval of information) as a distinct process from recognition (mental familiarity with the cue) [49]. Neural

nets’ biological inspiration suggests that there might be a similar difference between cue-aided retrieval and freeform generation. In the context of NLP, we are interested in the implicit understandings and assumptions embedded in LLMs and hypothesize that an entirely generative approach is most conducive to this study.

2.3.3 Explanation Structure

Explanations come in various structures, as seen in the typology defined by Ribeiro et al. (2022) [50]. Shown in Figure 2.3, our work focuses on a subset of said typology. WIKIWHY includes two structures that explain cause-and-effect relations:

1. Multi-hop step sequences $\mathcal{C} \rightarrow \mathcal{S}_1 \rightarrow \mathcal{S}_2 \rightarrow \dots \rightarrow \mathcal{S}_n \rightarrow \mathcal{E}$
2. Rationale sets $(\mathcal{C}, \mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_n) \rightarrow \mathcal{E}$

While the chain structure adds intermediate conclusions between cause and effect, rationale sets contain elements that support the relation from without. The rationale set topology acts as our general, catch-all case that other structures can be condensed to. Since our data collection procedure promotes a stepwise, ordered approach, we also consider the sequential topology to respect the structure exhibited in applicable explanations. We forego the unstructured approach as even limited structure helps bring freeform generated text evaluation within reach. Finally, we opt against pursuing the most complex entailment tree organization to maintain naturalness and facilitate crowdsourcing scalability.

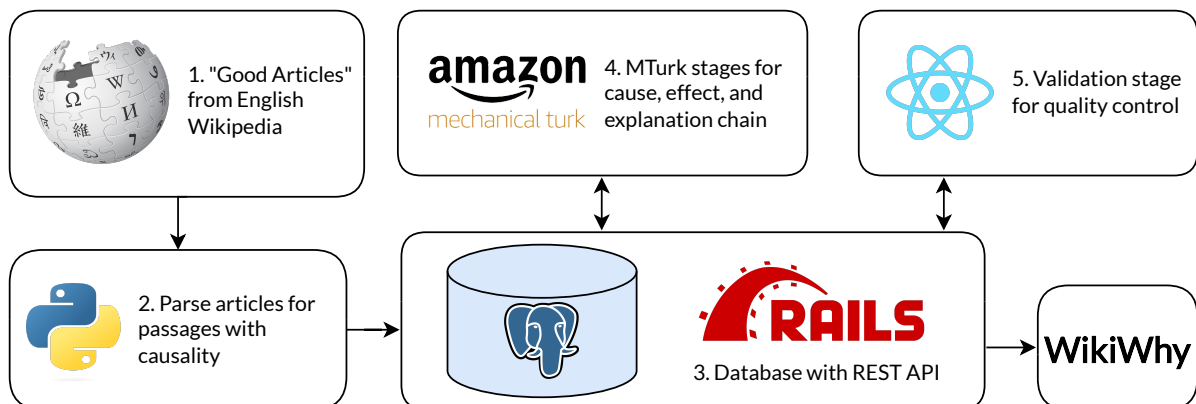


Figure 2.4: Dataset Collection and Validation Pipeline

2.4 Dataset

2.4.1 Data Collection

The objective of WIKIWHY is to present a high-quality, challenging dataset of QA pairs with corresponding causes, effects, and explanations. We developed an extensive data collection and validation pipeline around Amazon Mechanical Turk, depicted in (appendix). For each stage involving crowdsourced annotations, we perform rigorous worker-level quality control to ensure the dataset’s quality. The exact procedures are detailed in §subsection A.1.2 in the Appendix.

Preprocessing We begin with English Wikipedia’s corpus of “Good Articles,”¹ whose strict criteria of verifiability and neutrality (among others) ensure that WIKIWHY does not evaluate models on misinformation or opinionated views. From these articles, we extract passages containing causal relations using causal connectives. We selected a list of causal keywords (Appendix, §subsection A.1.1) from a more extensive set of causal connectives as their presence in a passage guarantees the existence of a cause and effect relation—some excluded connectives such as “since” or “as” are highly prevalent but

¹https://en.wikipedia.org/wiki/Wikipedia:Good_articles/all

are not necessarily causal. The presence of a causal word pattern on its own is a very simple heuristic—in the subsequent collection steps, we hired crowdworkers to ensure the quality of each sample.

QA Synthesis (Stage 1) Randomly sampled preprocessed Wikipedia passages containing potential causal statements were shown to qualified Amazon Mechanical Turk (MTurk) workers (see ethics statement for details), who were tasked with extracting the highlighted causal relation from the passage and re-framing it as a “why” question when possible. While automatic cause-effect relation extraction has seen recent progress [40], this human intelligence task (HIT) remains vital for two reasons. First, we find that quality in cause-effect is crucial for meaningful and valid explanations in the following stage. More importantly, we depend on human annotators to add sufficient context to the text of the cause, effect, and question to disambiguate them. This enables the question and cause-effect relation to be presented to models without the context we prepared (e.g., “Why was the river diverted?” is unanswerable without additional context). This feature is key to enabling WIKIWHY to assess the information and ideas within LLMs as opposed to whatever may be present in the context.

Explanation Synthesis (Stage 2) After verifying the quality of the examples, we prompt crowd workers to explain cause-effect pairs from **Stage 1** (§2.4.1). To encourage structured explanation, we supply an interface that allows sentences or ideas to be entered one at a time in separate fields. Though the input pairs should be context-independent, we provide the original passage as an aid for understanding the topic. Furthermore, we provide the link to the source article to encourage explanations leveraging topic-specific information in addition to commonsense knowledge.

WIKIWHY Examples

GENRE – Geography

- c* The geographic isolation of the Hupa homeland
 - s*₁ The Hupa’s homeland was separated by bodies of water or mountains
 - s*₂ Not many people could get to the Hupa’s homeland
 - e* The Hupa had few interactions with early European explorers up to the 19th century
-

GENRE – Literature

- c* Increased language contact in the globalizing world
 - s*₁ Increased contact between people requires increased communication
 - s*₂ Speaker of uncommon languages switch to more common languages
 - s*₃ Switching away from uncommon languages leads to them being forgotten
 - e* Many small languages are becoming endangered as their speakers shift to other languages
-

GENRE – Media

- c* Seeing the Castle of Cagliostro entrenched in Yamazaki that Japan can make high-quality films
 - s*₁ Viewing The Castle of Cagliostro inspired Takashi Yamazaki
 - s*₂ Out of national pride, Takashi Yamazaki followed a model that he believed would produce quality films
 - e* Director Takashi Yamazaki modeled his 2019 film Lupin III: The First after The Castle of Cagliostro
-

GENRE – Music

- c* The duration of Hotel California was longer than songs generally played by radio stations
 - s*₁ Most songs are only 3-4 minutes long
 - s*₂ Hotel California is over 6 minutes
 - s*₃ People would not want to listen to same song on radio for that long
 - e* Don Felder had doubts about the 1997 Eagles song Hotel California
-

GENRE – Natural Sciences

- c* The thermal stress at dawn and dusk
 - s*₁ The thermal temperatures change so drastically the rocks expand and contract
 - s*₂ This process weakens the structural integrity of the rocks
 - e* The boulders on Ceres are brittle and degrade rapidly
-

GENRE – Technology

- c* The use of coal power in Turkey
 - s*₁ Burning coal leads to air pollution
 - s*₂ Air pollution causes sickness and early death
 - s*₃ Sick and dead people cannot work
 - e* 1.4 million working days were lost across the population of Turkey in 2019
-

Table 2.2: Examples from 6 most frequent topics covered in WIKIWHY. *c* denotes cause, *e* effect, and *s*_{*i*} the *i*th rationale sentence.

Genres	Raw #	Freq.
AGRICULTURE	131	0.436
ARTS	577	0.396
ENGINEERING	952	0.336
GEOGRAPHY	754	0.624
HISTORY	1023	0.433
LITERATURE	455	0.340
MATHEMATICS	27	0.227
MEDIA	1773	0.399
MUSIC	1070	0.229
NATURAL SCIENCES	2952	0.768
PHILOSOPHY	302	0.465

Table 2.3: WIKIWHY dataset contains a diverse set of 11 genres. The raw counts of topic themes in articles is presented in the second column. The relative frequency is the percentage of articles in WIKIWHY sub-sampled from the *Good* Wikipedia articles list.

2.4.2 Dataset Description

Entry Contents In addition to the main fields of the question, answer, and explanation, each dataset entry contains the underlying relation’s cause and effect, the passage the question was extracted from, the article the passage is from, and Wikipedia’s topic categorization for that article.

Topic Diversity WIKIWHY improves upon other datasets due to its ability to examine reasoning proficiency across a broader range of concepts (Table 2.2 contains examples from the six most frequent topics). Overall, WIKIWHY contains a diverse set of 11 genres as shown in Table 2.3.

Rationale The statistics for the reasoning component are shown in Table 2.4. On average, each rationale contains **1.5137** elements. Figure 2.5 shows a histogram of rationale length by sentence count. WIKIWHY includes a range of rationale lengths, with more than one-third of examples (36%) containing two or more reasoning steps.

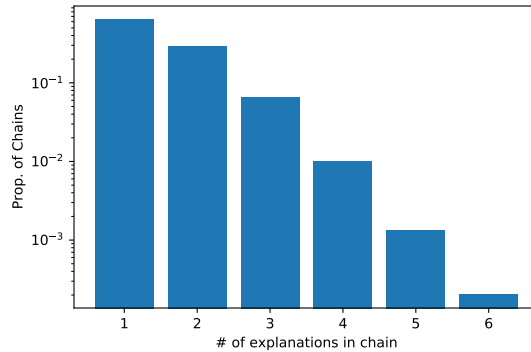


Figure 2.5: Rationale Length Distribution

WikiWhy Statistics	
# of Train	7,397
# of Dev	1,004
# of Test	1,005
# of Rationale	9,406
# of Rationale Elements	14,238
Avg. # Rationale Length	1.5137
Avg. # Tokens per Element	16.697

Table 2.4: WikiWhy Summary Statistics

2.5 Experiments

2.5.1 Experimental Settings and Models

Task Notation Let \mathcal{C} be a cause clause; \mathcal{E} be an effect clause corresponding to \mathcal{C} ; \mathcal{Q} be a question equivalent to “Why is it the case that \mathcal{E} ?”; \mathcal{A} be the answer to question \mathcal{Q} ²; \mathcal{X} be the explanation = $(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k)$ where \mathcal{S}_i is a sentence such that:

$$\mathcal{C} \wedge \mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \dots \wedge \mathcal{S}_k \vdash \mathcal{E}$$

²Note that \mathcal{Q} is a query that provides \mathcal{E} and is correctly answered by \mathcal{C} , $\mathcal{C} = \mathcal{A}$.

Task 1: Standard Question Answering (QA). **Input** = \mathcal{Q} , **Output** = \mathcal{A} . For thoroughness, we confirm high performance on **Task 1** (Standard QA) in the open-book setting. For this set of experiments, we use the classic approach of breaking the task into separate retrieval and reading comprehension phases. We experiment with BM25 [51] and Dense Passage Retriever (DPR) [52] as our document retriever, using their Pyserini implementations [53]. Using the Natural Questions [54] encoder, as in the original DPR paper, we build custom indices around segments from the subset of Wikipedia Articles shown to workers at collection time. For reading comprehension, we experimented with RoBERTa [55] and Big Bird [56] QA models. We also fine-tune a Fusion-in-Decoder (FiD) [57] model (80-10-10 split; default configurations), hypothesizing the decode-time combination of ideas could better model cause-effect relations.

The performance was unsurprisingly high, with BM25 achieving a high Top-1 Accuracy score of 0.810 in retrieval and FiD reaching a mean BERT-f1 of 0.78 (Table A.3 in Appendix). While retrieving the appropriate Wikipedia passage relating to some topic is straightforward, we found that producing an explanation of comparable quality to our gold rationales was difficult for the models we tested.

Task 2: Explanation Only (EO). **Input** = $(\mathcal{C}, \mathcal{E})$, **Output** = \mathcal{X} First, we examine Task 2: generating an explanation given an initial cause-effect pair. Given their stronger zero-shot generalization [58], we choose decoder-only models for our baselines. In this vein, we investigate the few-shot abilities of GPT-3 [59] with OpenAI’s most capable model, DaVinci-002³, at otherwise default settings. To better coax out the intermediates between cause and effect, we conduct prompt engineering over Wei et al. (2022)[26]’s Chain of Thought method. The exemplars are shown in Figure A.1.

We also make use of WIKIWHY’s scale for fine-tuning GPT-2 [60]. In this set of ex-

³<https://platform.openai.com/docs/models>

periments, we attempt to balance improving GPT-2’s understanding of the task’s structure while preserving the model’s “intuitions” for examination. We train GPT-2 for ten epochs using the training split ($\approx 80\%$ of the data) and Adam [61] optimizer with standard hyperparameters (learning rate $\alpha = .001$, $\beta_1 = .9$, $\beta_2 = .999$, $\epsilon = 1e-8$, decay = 0). For this tuned model we introduce special delimiter tokens `<cause>`, `<effect>`, and `<explanation>` in addition to the beginning and end tokens `<bos>` and `<eos>`. To support the delimiters and help the model distinguish the segments, we add token type embeddings (marking cause, effect, and explanation) as part of the preprocessing phase. At decoding time, we experiment with multiple temperatures.

Task 3: Answer and Explanation (A&E). **Input** = \mathcal{Q} , **Output** = $(\mathcal{A}, \mathcal{X})$ To investigate the performance of jointly predicting an answer and explanation given only a “why” question, we carry forward with our best performing baseline from the **EO** task (Task 2) — chain-of-thought prompted GPT-3. The first setting in this experiment set tasks a single model with the full end-to-end procedure. Once again, we utilize Chain-of-Thought (CoT) prompting, albeit with a modified prompt that also requests an answer to handle the different input format. Considering the impressive performance of existing (Information Retrieval) IR techniques on the **QA** task described above, we also study an additional setting incorporating the **QA** task. In the “*pipeline*” setting, the explainer model still lacks access to the ideal answer (the explanation’s starting point) but benefits from a reader model’s access to the original context. Here we combine our strongest performing approaches to the **QA** and **EO** tasks to make a 3-step pipeline of retrieval (BM25), reading (FiD), and explanation (GPT-3).

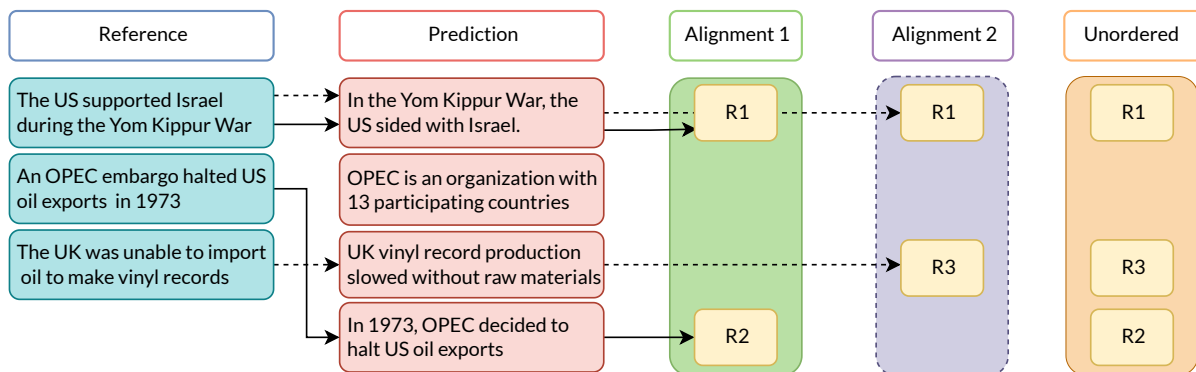


Figure 2.6: Alignment example for sentence-level metrics. Ordered evaluation uses the longest common subsequence as shown by alignment 1 and 2. The final alignment’s length is used to compute F-score metrics.

2.5.2 Automatic Evaluation Metrics

While the still developing area of text generation has measures and proxies for similarity that help with simple sequences, comparing reasoning sequences or rationale sets requires more involved measures. With the two topologies introduced in §2.3.3 in mind, we propose two related metrics, unordered and ordered, to handle sets and sequences, respectively.

Unordered Evaluation This first approach compares the ideas contained in the predictions and references. First, we split predicted and reference explanations into “ideas” or “steps” by sentence. We then compute a matrix of pairwise similarity scores before using a threshold to classify “matches”. Since a single prediction sentence may contain multiple reference ideas, we keep separate counts of precise prediction steps and covered reference steps. These counts are then micro-averaged for the test set’s overall precision, recall, and F1 scores.

Ordered Evaluation To respect the structure of multi-hop explanations, we penalize incorrectly ordered explanations. Here, we use the previously generated pairwise score

Panelist Evaluation Criteria	
1.	Similarity: Is the prediction similar to the reference?

2.	Correctness: Is the prediction a valid or correct explanation of the \mathcal{CE}^a pair?
<hr/>	
^a \mathcal{CE} refers to cause-effect	

Figure 2.7: Panel Evaluation Criteria analyzing the Similarity & Correctness to assign a binary yes/no rating.

matrix and its alignments to generate all possible assignments of prediction sequence elements to reference elements. As demonstrated in Figure 2.6, we compute the length of the longest common subsequence (LCS) between a prediction alignment against the reference labels for each candidate assignment. This length becomes the count of correctly incorporated structural elements— true positives. Note that the LCS alignment discounts repeated ideas in the prediction.

Metric Validity To understand the usefulness of our constructed metrics, we compare them against human judgements. A panel of 3 undergraduate students compared pairs of predictions and references on two binary scales, as shown in Figure 2.7. Each panelist answers the questions (**1. Similarity**) “Is the prediction similar to the reference?” and (**2. Correctness**) “Is the prediction a valid or correct explanation of the cause-effect pair?” Summing the panelist scores for each pair, we found a strong correlation ($r = 0.82$) between the similarity and correctness judgement. This validates comparison with WIKIWHY gold explanations as a useful proxy for explanation quality. Our proposed sentence-level processing incorporates the intuitions of checking for completeness with recall and penalizing over-explanation with precision.

Further, we use a single-explanation version of F-score to compare this proposed automatic metric with human judgement (the proposed F-score measures aggregate through

the whole dataset). With this variation, we find a modest correlation ($r = 0.35$) between ordered F1 and similarity, among other weaker correlations.

Besides supporting our proposed methods, this correlation analysis also enabled a data-driven approach to calibrating our similarity metric and match criteria. For each similarity metric, we selected a starting point through manual inspection of prediction-reference-similarity triples (which threshold value divides “genuine” from mistaken similarity) and used correlation for refinement. After trials with BLEURT [62] and BERTScore [5], different underlying models and different match thresholds, we selected BERTScore using a large DeBERTa [63] model (`microsoft/deberta-xlarge-mnli`) at a threshold of 0.64.

2.5.3 Human Evaluation

Recent studies by Goyal et al. (2022) [64] show that automatic metrics may not reliably evaluate results produced by models with few-shot capabilities like GPT-3. In light of this, we supplement our automatic evaluation with an additional human evaluation. We first evaluate each setting in each experiment using the binary correctness scale (see criteria definition below). Following this evaluation, we select the highest scoring explanations for each set of experiments for additional fine-grained evaluation. For each human evaluation task, we present a panel of three undergraduate students a random sample of 50 entries from each setting and the following binary (**True/False**) criteria guidelines:

- **Correctness:** Mark true if and only if the explanation is both complete and satisfying.
- **Concision:** Mark true if the explanation says everything it needs to say and nothing more. Mark false if extra information is included.

- **Fluency:** Is the explanation writing fluent? Mark false if there are any mechanical mistakes.
- **Validity:** Does the explanation make logical sense? Ignore whether or not the explanation successfully explains the cause/effect relation. Mark false if the explanation contains any illogical or untrue conclusions.
- **Win/Tie/Lose:** Compare the generated explanation against the provided reference (WIKIWHY gold explanation). Mark Win if you prefer the generated explanation, Tie if you have no preference, and Lose if you prefer the reference explanation. Equation 2.1 describes this where \mathcal{R} is the result when fed the WIKIWHY gold explanation, \mathcal{E} and the generated explanation, $\hat{\mathcal{E}}$.

$$\mathcal{R}(\mathcal{E}, \hat{\mathcal{E}}) = \begin{cases} \text{Win} & \text{if } \hat{\mathcal{E}} > \mathcal{E} \\ \text{Tie} & \text{if } \hat{\mathcal{E}} = \mathcal{E} \\ \text{Lose} & \text{if } \hat{\mathcal{E}} < \mathcal{E} \end{cases} \quad (2.1)$$

2.5.4 Results

Fine-Grained Human Evaluation With our human evaluation experiments, we find significant room for improvement across the board. Our strongest baseline, GPT-3 with greedy decoding, produced explanations judged to be satisfactory only 66% of the time in the most favourable setting of Task 2: EO (Table 3.4). Moreover, these explanations were judged to be worse than the gold reference 58% of the time. These results from our strongest baseline leave plenty of room to improve upon and motivate future work on this reasoning task.

Decoding Our experiments show increased performance with lower temperature sampling and best results with greedy decoding (Table 2.5). This aligns with existing notions of higher temperatures better suiting “creative,” open-ended tasks as opposed to more grounded ones. Explaining, as we hypothesize, relies more on the embedded assumptions in a generative model rather than the unlikely associations made more likely at higher temperatures.

Model Differences We find that GPT-3 significantly outperforms GPT-2. Comparing GPT-3’s output against its predecessor’s strongest setting shows increases in both ordered and unordered F1 scores by over 50%. Despite benefiting from fine-tuning and additional structural support from token type embeddings, GPT-2’s explanations are lacking compared to GPT-3’s few-shot explanations using only 4 exemplars. We find that GPT-2’s statements are often not only incomplete/unsatisfying for explaining the cause-effect relation at hand but also simply invalid. 94% of GPT-2’s statements were deemed worse than WIKIWHY’s gold references. The only area GPT-2 outperformed GPT-3 was in concision, however this is more a demerit of GPT-3 rather than a merit of GPT-2. We found that GPT-3 tended to occasionally add unnecessary detail to its explanations, often defining one of the entities in the prompt.

Answer & Explanation On the A&E task (Task 3), we find results that align cleanly with preconceived intuitions. Our baseline model is able to better handle explanations from points A to B when A is fixed and provided. Requiring the same procedures to generate more output creates more variance as incorrect or alternative starting points mislead the remaining generation. The “pipeline” setting strengthens this trend, as the better-informed answer generation allows for a higher quality explanation. This setting, simulating a three-step process with different models handling each step, demonstrates

Experiments	Unordered			Ordered			Human
	Prec. ¹	Rec. ²	BERT-f1	Prec. ¹	Rec. ²	BERT-f1	Correct
Task 2: EO							
GPT-2							
<i>Greedy</i>	0.249	0.196	0.220	0.239	0.179	0.204	0.100
<i>T = 0.50</i>	0.218	0.164	0.188	0.194	0.146	0.166	0.065
<i>T = 1.00</i>	0.072	0.056	0.063	0.071	0.054	0.062	0.064
GPT-3							
<i>Greedy</i>	0.347	0.388	0.366	0.307	0.355	0.329	0.660
<i>T = 1.00</i>	0.326	0.356	0.340	0.291	0.328	0.308	0.481
Task 3: A&E							
GPT-3							
<i>Single-Model</i>	0.092	0.095	0.094	0.082	0.092	0.087	0.140
<i>Pipeline</i>	0.229	0.233	0.231	0.211	0.220	0.215	0.387

¹Precision, ²Recall

Table 2.5: Baseline Performance on Explanation Tasks (EO = Explanation-Only, A&E: Answer and Explanation). For Task 3, the Single Model setting has the generative model complete the end-to-end task in a single pass. The Pipeline setting allows each stage to be handled separately (QA is handled by BM25+FiD and explanation is done by GPT-3). Human evaluation was done with on a binary scale (correct/incorrect) and we report the proportion of correct evaluations.

an intermediate performance between having the oracle-provided answer and requiring the explainer to manage the entire process. Under these settings, where the model’s input excludes the correct answer (the cause), the “validity” criteria of our human evaluation is especially interesting. While the majority of the end-to-end setting’s explanations were marked incorrect or unsatisfying, a notable proportion was still marked as having a valid chain of reasoning. This suggests that a significant portion of this setting’s difficulty lies in the generation of the initial correct answer.

Explanation Failure A typical error observed in GPT-3’s predictions is repeating the cause-effect relation. To explain why [A] leads to [B], GPT-3 might only write “[B] because [A]” or another semantically equivalent formulation. This pattern may be ex-

Setting	Fine Grained Human Evaluation						
	Correctness	Concision	Fluency	Validity	W (↑)	T	L (↓)
GPT-2: EO	0.100	0.880	0.860	0.520	0.040	0.040	0.920
GPT-3: EO	0.660	0.680	1.00	0.960	0.080	0.360	0.580
GPT-3: A&E	0.140	0.680	0.900	0.720	0.080	0.100	0.820

Table 2.6: Human evaluation. Overall correctness is marked on a binary scale– an explanation is complete and satisfying or not. Concision penalizes for repeated or unnecessary information, fluency evaluates grammar, and validity measures if the generated sequence makes logical sense regardless if it correctly explains the relation. W refers to Win, T refers to Tie, and L refers to Lose. For Win/Lose/Tie, annotators compared the generations against WIKIWHY’s gold references.

plainable with a fine-tuned baseline where annotation errors of the same kind might have slipped into the training set, but GPT-3 was prompted with hand-picked exemplars with no such mistakes. Furthermore, we observe successful explanations on some inputs we expect to be more difficult alongside errors on relatively less challenging inputs. These observations, together with the consistently high fluency scores showing syntactic competence, seem to indicate a reasoning failure as opposed to a systematic “misunderstanding” of the task at hand. Per the original goal of better understanding what and how LLMs “understand” the world, this might indicate a gap in commonsense: that GPT simply memorized the fact that [A] leads to [B].

2.6 Conclusion

In this chapter, we release WIKIWHY, a Question-Answering dataset enabling the analysis and improvement of LLMs’ reasoning capability. We propose explanation **between** grounded cause-effect pairs to distinguish memorization of the relation from a genuine understanding of the underlying mechanics. Compared to related works on explainable QA, our explanation format finds a natural middle ground that balances **com-**

plexity and **depth**, allowing our crowdsourcing methods to produce thought-provoking examples while being highly scalable. We exploit this scalability to cover topics previously overlooked by other explanation datasets and demonstrate our proposed task to be difficult with strong baselines (our experiments feature models failing to produce satisfying explanations even under ideal conditions). Finally, we motivate the development of new automatic metrics that are better able to handle the complexities of generated reasoning.

2.7 Ethics Statement

For data collection, our listing required workers to have a high HIT approval rating ($\geq 96\%$) and be located in English speaking regions (Australia, Canada, New Zealand, the United Kingdom, and the United States). The average hourly pay is 12.00 dollars, which exceeds the income requirements proposed in the human subjects research protocols. The project is classified as exempt status for IRB. Our interfaces include notices that we are collecting information for dataset creation, consent forms, and a link for inquiries and concerns. Our MTurk interfaces are displayed in the Appendix A. Due to the experimental nature, limited production applicability, and relatively small dataset scale, we believe the potential for misuse or harm is negligible.

2.8 Acknowledgements

The contents of this chapter is a result of a collaboration with Matthew Ho, Justin Chang, Michael Saxon, and Sharon Levy. This collaboration was a result of the Early Research Scholars Program (ERSP) advised by William Wang and coordinated by Diba

Mirza and Chinmay Sonar. The WIKIWHY dataset and codebase, released publically⁴, contains the model tuning procedures, settings, few-shot prompts, and evaluation script. This work has previously appeared in 2023 Proceedings of the Eleventh International Conference on Learning Representations.

⁴<https://github.com/matt-seb-ho/WikiWhy>

Chapter 3

Scene Understanding with LLMs

3.1 Introduction

Augmented reality (AR) holds the potential to seamlessly integrate digital content into the physical world, necessitating the placement of virtual elements in natural locations. The population of 3D environments with 3D virtual content often requires developers to specify a target location, such as a “wall”, for each 3D virtual object, such as a painting. Then, while the application is running, it will recognize and track the specified location in the current scene, and anchor the virtual object on it. However, there are many cases where adding new virtual objects, including those not considered by the developers, into 3D scenes is desirable. For example, this need arises when applying custom themes for entertainment or other applications that present modified realities, such as simulation and training. Adapting AR content to different physical environments often involves placing many virtual objects, making doing this manually in every new environment cumbersome. Some automated placement techniques exist; however, their applicability is limited as they can not accommodate arbitrary objects and scenes due to the closed-vocabulary nature of the underlying machine-learning models. This constraint

	Closed-Vocabulary	Open-Vocabulary
Properties	1. Trained on a fixed set of words/categories 2. Only handles these predefined values at inference time	1. Trained on a large dataset, giving it an <i>understanding</i> of the language 2. Can handle inputs not seen during training
Object Detector	A model trained to recognize cat and dog will not recognize tree	Model accepts any text (e.g. cat or the person wearing a red shirt)

Table 3.1: Properties of Closed-Vocabulary and Open-Vocabulary Models

implies that these models can only process a predefined set of words. On the other hand, “open-vocabulary” models can adapt to words not seen during training. Table 3.1 analyzes the properties of closed-vocabulary and open-vocabulary models. In general, we see that closed-vocabulary models can only accept a **fixed** vocabulary set, thereby, only having the ability to recognize those fixed categories at inference time. Open-Vocabulary models are trained on a **large** dataset allowing it to generalize over domains, hence, giving a better *understanding* of the language.

We combine several models to create OCTO+, a state-of-the-art pipeline and evaluation methodology for virtual content placement in the Mixed Reality (MR) setting. We build on OCTOPUS [65], a recently introduced 8-stage approach to the placement problem. OCTO+ accepts as input an image of a scene and a text description of a virtual object to be placed in the scene and determines the most *natural* location in the scene for the object to be placed.

In summary, this chapter presents the following contributions of this work:

- State-of-the-art pipeline OCTO+, outperforming GPT-4V and the predecessor OCTOPUS method on virtual content placement in augmented reality scenes.

- Extensive experimentation conducted with the state-of-the-art large multimodal models (LMMs), large language models (LLMs), image-editing models, and methods employing a series of models, leading to an overall 3-stage conceptualization for the automatic placement problem.
- Introduce PEARL, a benchmark for **P**lacement **E**valuation of **A**ugmented **R**eality **E**lements – **PEARL**.

3.2 Related Work

3.2.1 Virtual Content Placement

In the context of augmented reality, virtual objects must satisfy physical and semantic constraints in order to appear *natural*.

In general, virtual content should be aligned with the natural world and follow the laws of physics. For example, furniture should either be resting on the floor or against a wall [66], and objects should not float above the ground [67]. Evaluating such constraints automatically is a difficult task because a unique ground truth does not exist, and even if an authoritative ground truth placement were available, simply taking the distance between a proposed location and a ground truth location is not a reliable metric. The quality of a placement location also depends on many other factors, including viewing angle, viewing distance, and object size [68]. To address this, Rafi et al. (2022) [68] introduced a framework that predicts how humans would rate the placement of a virtual object. The framework makes these predictions based on the “placement gap”, which is the distance between the bottom of the object and the plane it is supposed to be placed on, and other factors such as viewing angle, so it is meant to measure how physically realistic the placement of a virtual object looks. Our benchmark, on the other hand,

focuses on how semantically realistic the placement of a virtual object is.

Previous work has also investigated how to place objects. To position virtual interface elements seamlessly when transitioning between physical locations, Cheng et al. [69] introduced an approach that discovers a semantically similar spot in the new scene corresponding to where the virtual interface elements were placed in the previous scene. To put virtual agents in AR, Lang et al.[70] introduced a method involving reconstructing the 3D scene, identifying key objects, and optimizing a cost function based on the detected objects, among other things. Existing work focuses on placing specific objects or operates with closed-vocabulary choices. In contrast, we aim to create a single pipeline to identify *any* object without special training.

3.2.2 Vision-Language Models

Many of the models we use to automate the virtual content placement task are able to handle both image and text, but historically, machine learning models were typically either used on images or text. However, both types of models can have similar architectures, such as the Transformer architecture [7]. For a model to process images and text, they must be encoded into the same semantic embedding space, where similar text and images are close and unrelated text and images are distant.

One way to ‘align’ text and images into the same embedding space is to use Contrastive Language-Image Pretraining (CLIP) [16]. Given \mathcal{N} pairs of text-image- (t_k, i_k) , a text encoder \mathcal{E}_t and an image encoder \mathcal{E}_i are used to create text embeddings $\mathcal{T}_e = \{t_1 \dots t_{\mathcal{N}}\}$ and image embeddings $\mathcal{I}_e = \{i_1 \dots i_{\mathcal{N}}\}$. The encoders are trained using the contrastive learning objective. In contrastive learning, we maximize the cosine similarity between embeddings (t_a, i_a) that originated from the same text-image pair and to minimize the cosine similarity between embeddings (t_a, i_b) where $a \neq b$ that came from

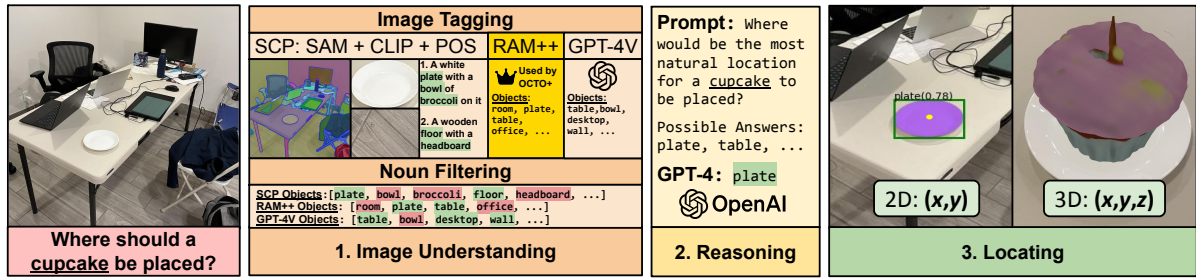



Figure 3.1: Overview of various methods we experimented with, including **OCTO+** . To determine where a cupcake should be placed, we perform three stages: 1) **image understanding**: generate a list of all objects in the image (**OCTO+** uses RAM++); 2) **reasoning**: select the most natural object with GPT-4; 3) **locating**: locate the 2D coordinate of the selected object in the image and ray cast the 2D coordinate to determine the 3D location in the AR scene.

different text-image pairs. After this training, the encoders will map text and images into the same semantic embedding space, which allows models to take either images or text as input.

3.3 Method

Various approaches to address the broader challenge of mixed reality object placement were investigated. This task involves inputting a camera frame, denoted as **I**, along with a natural language description of the object, denoted as **T**, and generating the optimal 2D coordinate for placing the object. The problem can be further decomposed by moving from 3D coordinates to 2D image space, as complete 3D scene models are not always available, and if they are, a 3D coordinate can be easily obtained by ray casting. While such object placement in a 2D camera frame may be a simple task for humans, it requires understanding what items are in the image, reasoning about which item in the image the object would most naturally be placed on or nearby, and finally, locating the item in the image on which the object should be placed. These subtasks are outlined in Figure 3.1, and the next section will describe each in detail.

3.3.1 Stage 1. Image Understanding

Image understanding is the process of interpreting the content in images similarly to how humans do it. In the context of object placement in augmented reality, the image understanding stage focuses on identifying all the surfaces in the image that virtual objects could be placed on. Image tagging, or recognizing all objects in an image, can be approached in different ways, including by using object-level tagging models, image-level tagging models, large language models (LLMs), and large multimodal models (LMMs).

Object-level Tagging

Object-level tagging entails dividing the input image into regions of interest before generating any tags. We previously introduced the OCTOPUS [65] pipeline, using three state-of-the-art vision and language models to accomplish this task. First, it uses the Segment Anything Model (SAM) [71] to divide the image into regions that may contain objects. Next, OCTOPUS uses clip-text-decoder [72] to generate captions for each region. Lastly, it used English Part-of-Speech tagging in Flair [73] to extract nouns from each caption. We will refer to the chaining of these three models as SCP—Segment, Clip, Parts of Speech tagging.

Tag Filtering

Some nouns found may have been misidentified and must be filtered out before passing to the next stage. This is because the next stage assumes that all nouns given are represented in the image and are valid locations for an object to be placed. We experiment with multiple strategies to accomplish this.

- **ViLT** [74], a Vision Transformer-based model, is used in OCTOPUS for visual question answering. For every noun provided by SCP, the model is presented with

the image and the question: “Is there a *noun* in the image?” Only the nouns that result in ViLT outputting ‘yes’ are retained.

- **CLIPSeg** [75] is a model that takes an image and text query as input and generates a heatmap illustrating the correlation between each image pixel and the text query. CLIPSeg is run on the input image and each noun, and the intensity of the brightest pixel in the resulting heatmap is recorded. This intensity is then used to rank the nouns, and the top- k nouns are kept, where k is a predetermined parameter.
- **Grounding DINO** [76] is a model that accepts as input an image and a text query (in our case, the SCP nouns separated by commas) and outputs bounding boxes for every object it found in the image related to the query. A threshold t can be adjusted to exclude boxes that are not sufficiently similar to any word in the query. We exclude any bounding boxes that cover over 90% of the image area, as these are typically generic words such as ‘room’ and do not refer to specific objects in the image.

All three of these methods effectively remove nouns incorrectly identified by SCP; however, they also risk excluding too many nouns. There is a trade-off between filtering out nouns not in the image and keeping valid candidate placement targets in the image. In the case of CLIPSeg and Grounding DINO, the values of k and t can be adjusted to fine-tune this balance.

Image-level Tagging

It is also possible to pass the entire image into an image tagging model without dividing it into regions first. One advantage to this approach is that the model has access to the entire image, rather than just a tiny patch, and can use that to identify objects better.

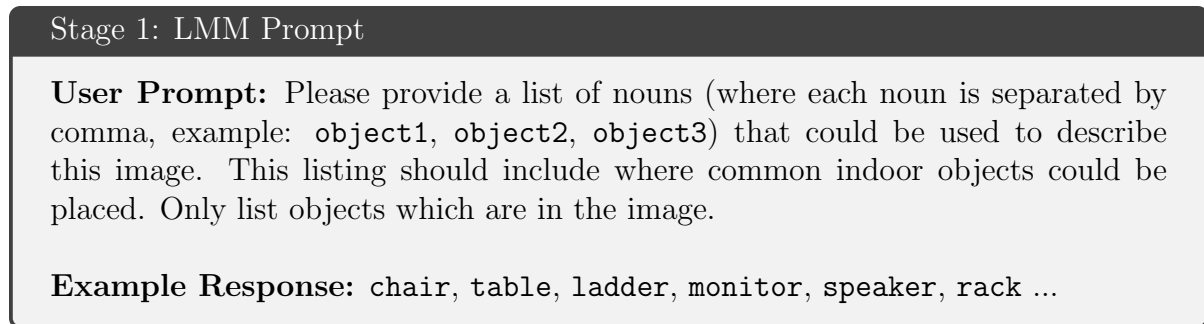


Figure 3.2: Large Multimodal Model (LMM) prompt for Stage 1 – Image Understanding

The image tagging model we experimented with is the state-of-the-art Recognize Anything Plus Model (RAM++) [77], which takes an image and a list of text labels (this list can contain any words, including those not seen during training), and determines which of the labels are in the image. By default, the model is run on 4,585 labels, and more can be added if needed (we found these default labels to be more than sufficient). We experimented with different thresholds for a tag to be classified as being in the image. We also tried using Grounding DINO as a tag filter since Grounding DINO was the best out of the three filter options for SCP (Table 3.2). RAM++ and Grounding DINO comprise the first stage of OCTO+.

Large Multimodal Models (LMMs)

Multimodal large language models can accept images and text as input, reason about them, and generate a text response. We used two such models, GPT4-V [78] and LLaVa-1.5 [79], to create a list of nouns by prompting them with the image and instructions (Figure 3.2).

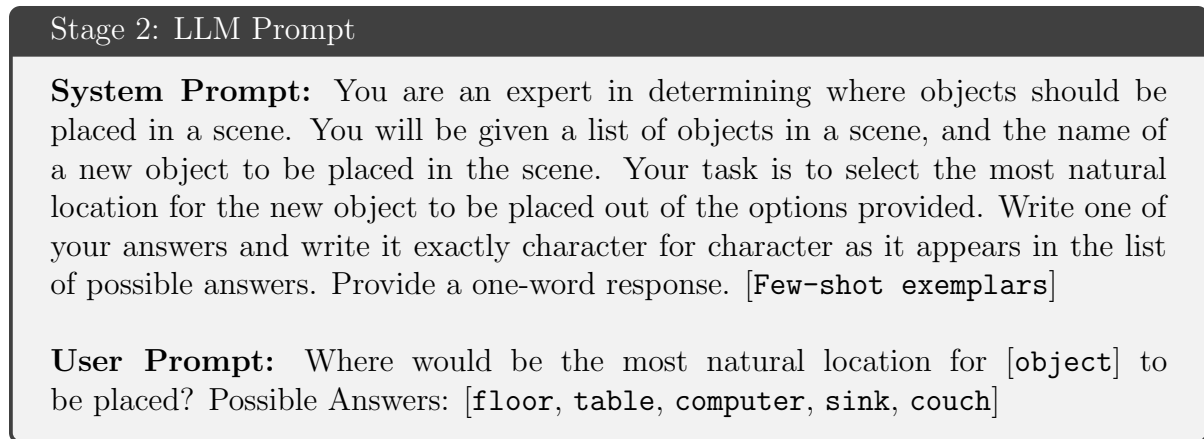


Figure 3.3: Large Language Model (LLM) prompt for Stage 2 – Reasoning. The square brackets denote additional or customized information, omitted here for brevity and generality.

3.3.2 Stage 2. Reasoning

The overall task for Stage 2 is to select the object from the list produced in Stage 1, which is the most natural location for the virtual object to be placed on. This requires complex reasoning abilities. Large language models (LLMs) have recently showcased exceptional reasoning capabilities for natural language generation. We use LLMs and LMMs to make this selection and take advantage of these abilities.

Large Language Model

Both OCTOPUS and OCTO+ use OpenAI’s currently most capable model, GPT-4 [78], to select the target object on which the virtual object should be placed. Chain of Thought (CoT) prompting [26] and in-context learning [80] are two notable techniques that increase the reasoning abilities of LLMs. As a result, we use 3-shot prompting, which means we provide 3 example questions and responses before the actual question to better guide the LLM. We provide the prompt to GPT-4 (Figure 3.3) using `temperature=0.2`.

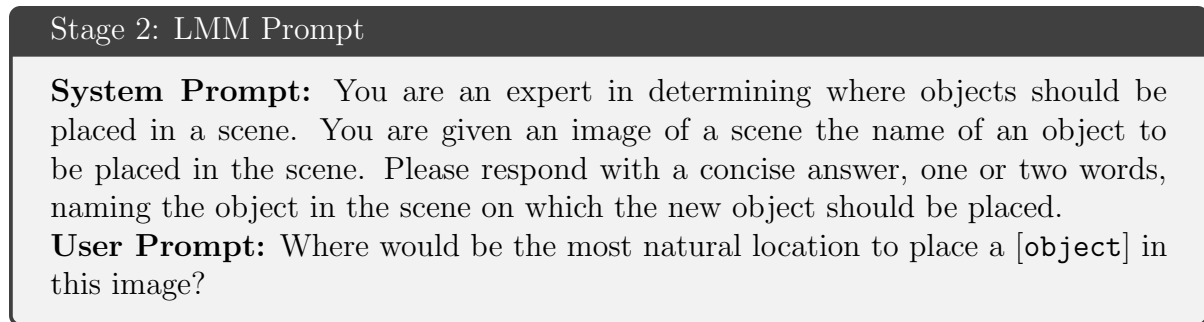


Figure 3.4: Large Multimodal Model (LMM) prompt for Stage 2 – Reasoning.

Multimodal Large Language Model

The original GPT-4 model was not a multimodal LLM, so it cannot view images. Therefore, it has to rely on the list of nouns from Stage 1 to understand the content of images. By contrast, multimodal LLMs, such as GPT-4V, can take images as input and answer questions about them. As a result, MLLMs can consolidate Stages 1 and 2 into a single step.

The OCTOPUS [65] paper experimented with visual question-answering models, such as ViLT. These models were tasked with processing an image and a text prompt that inquired about the optimal placement of an object within the image. At that time, the models frequently produced unsatisfactory responses. However, significant progress has since occurred in the field, so our experimentation now includes state-of-the-art closed-source and open-source models, specifically GPT-4V (via the API) [78] and 13B LLaVA-1.5 [79] with a `temperature=0.01`. We directly provide the models with the image and object to be placed, prompting them to name where in the image the object should be placed. Figure 3.4 showcases the prompt used for GPT-4V.

3.3.3 Stage 3: Locating

Once a suitable surface for the virtual object’s placement has been found, the next task is to select a 2D coordinate for the object to be placed. We explore two distinct strategies: **CLIPSeg** and **Grounded-Segment-Anything**.

1. CLIPSeg

As discussed in the Tag Filtering section, CLIPSeg generates a heatmap indicating the similarity between each pixel in the image and the provided text query. We present CLIPSeg with our image and surface selected in Stage 2 to determine the object’s placement. We choose the location (x, y) of the pixel with the highest activation in the heatmap. This is the model used in OCTOPUS.

2. Grounded-Segment-Anything (G-SAM)

G-SAM [81] is another approach that integrates Grounding DINO and SAM. As input, we provide the image and text specifying the surface chosen in Stage 2. First, Grounding DINO identifies bounding box(es) corresponding to the text. These boxes are input to SAM, which generates a precise segmentation mask around the object. The masks are consolidated into one, and we select the point in the combined mask that is farthest away from any edge of the mask. This ensures that we do not select a point right at the edge of a surface, which would not be very natural. This is the model used by OCTO+.

Large Multimodal Models

We also examine performing Stages 1-3 in one step using GPT-4V. We ask GPT-4V to determine the (x, y) pixel location directly with the following prompt (Figure 3.5).

Stage 3: LMM Prompt

System Prompt: You are an expert in determining where objects should be placed in a scene. You are given an image of a scene and the name of an object to be placed in the scene. Please respond with the pixel coordinates locating where in the scene would be the most natural location to place the object. The image is 561 pixels wide and 427 pixels long, and the top left corner is the origin.

Chat Prompt: Where would be the most natural location to place [object] in this image? Please briefly explain your selection and enter the x and y coordinate locations in the format (x, y) at the end.

Example Response: The banana should be placed on the table. (173,294).

Figure 3.5: Large Multimodal Model (LMM) prompt for Stage 3 – Locating.

Image-Editing Methods

One final approach to obtain the 2D coordinate for object placement in an image starts with InstructPix2Pix [82]. InstructPix2Pix is an instruction-based image editing model that takes both an image and a text prompt specifying the desired edit. Using Stable Diffusion, InstructPix2Pix then generates a new image incorporating the edit. In our case, where we seek to identify the optimal placement of an object, we prompt InstructPix2Pix to “add [object]” and provide the input image. Leveraging its image understanding and reasoning capabilities, InstructPix2Pix generates a new image with the object seamlessly integrated. We then use Grounding-Segment-Anything to detect the object and segment the region. To determine the 2D placement coordinate, we select the bottom-most pixel in the mask, considering it to be the location of the surface beneath the object.

3.3.4 3D Location in AR Scene

Once the 2D (x, y) location in the image is identified, the final step is to calculate the corresponding 3D (x, y, z) position in the scene, which is where the virtual object

will be positioned in augmented reality (AR). To accomplish this, we employ raycasting into the scene (executed by ARKit and ARCore, supported natively in iOS and Android devices).

3.4 Evaluation and Results

To determine which models performed the best in each stage, we designed a series of experiments, one for each stage, to compare the different methods.

Experiment Setup A representative set of objects and images is needed for meaningful and fair evaluation. We limited our assessment to indoor scenes. By consulting LLMs and optimizing for object diversity, we compiled a list of 15 common indoor objects: (apple, cake, cup, plate, vase, stool, painting, lamp, book, bag, computer, pencil, shoes, cushion, cat). This step was decoupled entirely from the scene selection, for which we randomly sampled 100 indoor scene images from the NYU Depth Dataset [83] and Sun3D Dataset [84].

Annotation We had two team members annotate each of the 100 images with a *natural* and *unnatural* location to place each of the 15 objects. For example, it would be *natural* to place a cupcake on a plate, but *unnatural* to place a cupcake on the floor. Team members also wrote a list of valid locations for each object to be placed in each image.

Figure 3.6 describes the different placements. Any objects that were deemed unsuitable or irrelevant for a specific image were excluded from further analysis throughout the experiment for that particular image (this happened in 573 of the 1,500 image-object combinations). We also generated placement coordinates using random point selection. Ultimately, we arrived at 927 object location-image pairs for each of the three baseline placement methods. These images and lists of valid locations for each object in each

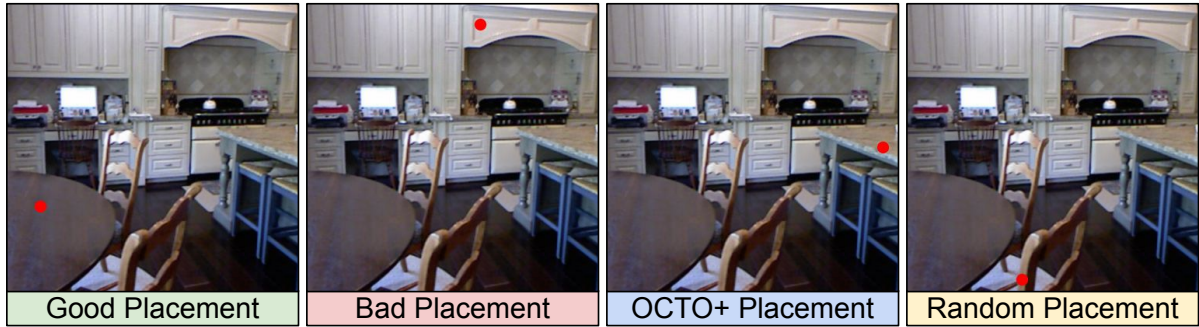


Figure 3.6: Comparative placements for one input image, and the prompt “Apple”.

image annotation are a part of the PEARL benchmark and are used to evaluate all three stages.

3.4.1 Stage 1: Image Understanding Evaluation

Stage 1 must produce an accurate list of objects in the image for the later stages to succeed. This means that all the important objects in the image should be included in the list, and objects not in the image should not. We consider objects important if they are surfaces on which other objects could be placed, such as tables, chairs, kitchen counters, or the floor. In other words, the necessary objects are the valid locations annotated as described above. Therefore, we compare the list of objects generated in Stage 1 with the valid locations in two ways: Exact Match & Sentence-BERT similarity.

- **Exact Match (EM):** For each object (e.g., “cat”) in each image, we check if any of its valid locations (e.g., “couch,” “floor”) is in the list of tags.
- **Sentence-BERT [85] (sBERT) Similarity:** For each object (e.g., “cat”) in each image, we find the maximum similarity between any of its valid locations (e.g., “couch,” “floor”) and any of the tags. We do this using sBERT, a modified version of BERT [12] that encodes text into a semantic embedding space, such that the cosine similarity between the embedding for words that have similar meanings (e.g.,

sBERT(“couch”, “sofa”) = 0.856) will be higher than it would be for words that have different meanings (e.g., sBERT(“couch”, “table”) = 0.334). Using sBERT, we can reward methods that generate tags with the correct semantic meaning.

Algorithm 1 Stage 1 Metrics Computation

Require: \mathcal{I} : 100 images

Require: \mathcal{T} : model-generated tags generated for each image

Require: \mathcal{O} : names of up to 15 objects placed in each image

Require: \mathcal{L} : valid locations for each object in each image

```

 $\mathcal{E} \leftarrow 0$  ▷ Number of exact matches EMs
 $\mathcal{S} \leftarrow 0$  ▷ Total sBERT score
 $\mathcal{N} \leftarrow \text{length}(\mathcal{I})$  ▷ Number of Tags n
for  $i \in 1$  to  $\mathcal{N}$  do
   $e \leftarrow 0$ 
   $s \leftarrow 0$ 
   $\mathcal{M} \leftarrow \text{length}(\mathcal{O}_i)$ 
  for  $j \in 1$  to  $\mathcal{M}$  do
    if  $\mathcal{T}_i \cap \mathcal{L}_{i,j} \neq \emptyset$  then
       $e \leftarrow e + 1$ 
    end if
     $s \leftarrow s + \max_{(t,l) \in \mathcal{T}_i \times \mathcal{L}_{i,j}} \text{sBERT}(t, l)$ 
  end for
   $\mathcal{E} \leftarrow \mathcal{E} + \frac{e}{\mathcal{M}}$ 
   $\mathcal{S} \leftarrow \mathcal{S} + \frac{s}{\mathcal{M}}$ 
end for
 $\mathcal{E} \leftarrow \frac{\mathcal{E}}{\mathcal{N}}$ 
 $\mathcal{S} \leftarrow \frac{\mathcal{S}}{\mathcal{N}}$ 

```

We used these techniques to compute overall metrics for the tagging stage using the algorithm shown in Algorithm 1, which computes the average score for each image, and then averages each image’s score to produce a final score. In addition to the average number of tags generated per image, these metrics are shown for each method in Table 3.2.

Starting with the object-level tagging method, we found that the unfiltered list of tags was very long (60.67 on average), with many of the nouns being irrelevant or not in the image. Out of the filtration methods we tried, we felt that Grounding DINO with

Models	Exact Match \uparrow	sBERT \uparrow	# Tags \downarrow
Object-level Tagging			
SCP (SAM + CLIP + POS)	0.734	0.892	60.67
+ ViLT	0.658	0.850	25.96
+ CLIPSeg (k=10)	0.439	0.698	<u>10.00</u>
+ CLIPSeg (k=15)	0.535	0.755	15.00
+ CLIPSeg (k=20)	0.634	0.807	20.00
+ CLIPSeg (k=30)	0.684	0.841	29.95
+ G-DINO ($\tau=.25$)	<u>0.712</u>	<u>0.852</u>	16.04
+ G-DINO ($\tau=.35$)	0.532	0.745	7.87
Image-level Tagging			
RAM++ ($\tau=0.8$)	0.909	0.976	61.66
+ G-DINO ($\tau=.25$)	<u>0.865</u>	<u>0.942</u>	18.51
RAM++ ($\tau=0.9$)	0.851	0.937	31.63
+ G-DINO ($\tau=.25$)	0.812	0.923	<u>15.67</u>
RAM++ ($\tau=1.0$)	0.619	0.854	16.55
+ G-DINO ($\tau=.25$)	0.610	0.834	10.84
Multimodal Large Language Models			
GPT-4V(ision)	0.399	0.782	11.69
LLaVA-v1.5-13B	0.605	0.805	29.77

Table 3.2: Stage 1 Metrics. Best **IN BOLD**, second best UNDERLINED.

a threshold of 0.25 had the best trade-off between keeping relevant nouns (the number of **EMs** and **sBERT** score only dropped by a few percent compared to unfiltered), while the average number of tags per image dropped by nearly 75%. Taking the top-20 (k=20) tags from CLIPSeg also performed well, so we experimented with both of these in the later stages. A trade-off exists where higher thresholds result in removing more undesired words and excluding more target words. We generally leaned towards less strict thresholds, as excluding a target word is usually more harmful than including an extra word.

On to the image-level tagging results, we found that RAM++ with 0.8 times the default threshold, combined with the best filtering strategy from before (Grounding DINO with a threshold of 0.25) had the best performance of any Stage 1 method. Compared to

RAM++ with the default threshold and no filter, this method had much better metrics and only a slightly longer average number of tags.

Lastly, we considered the multimodal large language models, which were prompted to generate a list of nouns in the image. We frequently omitted crucial nouns in both models, so we did not experiment further with their noun lists. Additionally, LLaVA often repeats sequences of words repeatedly, making it impractically slow. This was not an issue with GPT4-V, however.

3.4.2 Stage 2: Reasoning Evaluation

We measure the performance of this stage on the **PEARL** benchmark by comparing the target tag $\hat{\mathcal{T}}$ produced a method with the set of expert annotated tags \mathcal{T} for each object in each image. To measure the performance, we use two metrics:

- **Exact Match:** For each object in each image, we check if the predicted surface $\hat{\mathcal{T}}$ matches any of the expert annotated tags in \mathcal{T}
- **sBERT:** To reduce how much we penalize methods that select a word with the correct meaning but not an exact match (e.g., “couch” vs. “sofa”), we record the maximum similarity between $\hat{\mathcal{T}}$ and any of the annotated tags in \mathcal{T} .

We compute the average exact match and **sBERT** score over each object in each image and then average over all images to get a final score (there are 927 image-object pairs). We report the scores for the five methods we tried in Table 3.3. We find that the SCP tags filtered with G-DINO perform better than the SCP filtered by CLIPSeg with **EM** scores of 0.630 and 0.621, respectively. Still, RAM++ outperforms SCP with an **EM** of 0.645 and an **sBERT** score of 0.821. In Multimodal LLMs, we find that the open-source LLaVA-v1.5-13B model significantly outperforms GPT-4V. Although

Models	Exact Match \uparrow	sBERT \uparrow
LLM: Tags + Object as Input		
SCP (Filter: CLIPSeg) + GPT-4	0.621	0.786
SCP (Filter: G-DINO) + GPT-4	<u>0.630</u>	<u>0.791</u>
RAM++ (Filter: G-DINO) + GPT-4	0.645	0.821
Multimodal LLMs: Image + Object as Input		
GPT-4V(ision)	<u>0.479</u>	<u>0.754</u>
LLaVA-v1.5-13B	0.711	0.844

Table 3.3: Stage 2 Metrics. Best **IN BOLD**, second best UNDERLINED

this is the case, we find that LLaVA is substantially slower (when run on an NVIDIA TITAN X Pascal GPU) than GPT-4V, making it impractical to be used in a real-time AR application. Furthermore, in Table 3.5, it is evident that GPT-4V outperforms LLaVA with a higher score, suggesting that Stage 2 metrics may not fully represent the overall task.

3.4.3 Stage 3: Target Locating Evaluation

Measuring the performance of Stage 3 is the most important as it is the only way to evaluate the system end-to-end. Even if the first two stages correctly select the target noun, determining a *natural* location for placement is still required. Therefore, we establish a robust evaluation procedure incorporating automated metrics and human evaluation.

Automated Metrics

The 98 indoor scene images used in the **PEARL** benchmark came from the NYU Depth Dataset [83] and Sun3D dataset [84], which included segmentation masks for many indoor objects. Some of the images did not come with any segmentation masks labeled with any of **PEARL**'s annotated ground truth locations for particular objects (e.g.,

“desk” and “table” segmentation masks were not provided in an image for the indoor classroom, so we were not able to evaluate how well an apple could be placed in that image). As a result, we had to remove 152 of the image-object pairs from the original 927, leaving us with 775 image-object pairs for which we can compute metrics. Each pixel in the 561×427 segmentation mask was annotated with a value between 1 and 895, where each value was mapped to a unique label.

To evaluate how naturally an object is placed in an image, **PEARL** combines all of the segmentation masks whose label matches one of the ground truth locations for that object in that image. For instance, a “cat” could be naturally placed on a chair, floor, or couch in a given image. To obtain a new segmentation mask, assigning a value of 1 in all valid locations for the cat and 0 elsewhere, we consolidate the segmentation masks for that image corresponding to the labels chair, floor, or couch. In some cases, the segmentation masks had more specific labels than **PEARL** or used synonyms, so we had to change some of the labels in the segmentation dataset manually. For example, we changed “sofa” to “couch” and merged “coffee table” and “desk” into “table.” In total, we modified 43 segmentation mask labels.

$$V(x, y) = \begin{cases} 1 & \text{if } (x, y) \text{ is inside mask} \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

In the following descriptions of our metrics, we will use $(\hat{x}_{ij}, \hat{y}_{ij})$ to denote the 2D location that a model output for object j in image i . When we refer to a “mask” for an object and image, we refer to the set of valid locations for that object to be placed in that image.

The first automated metric we define is the percentage of the predicted 2D coordinate $(\hat{x}_{ij}, \hat{y}_{ij})$ was in the mask. For a given placement $(\hat{x}_{ij}, \hat{y}_{ij})$ of object j in image i , we assign

a score of 1 if $(\hat{x}_{ij}, \hat{y}_{ij})$ is in the mask, and 0 otherwise. More formally, $V(x, y)$, is used to define the value of a given 2D placement (x, y) in mask, as described by Equation 3.1. We repeat this for every object in an image and find the proportion of times the predicted location was in the mask. We then take the average of all images’ scores to compute the final IN MASK score in Table 3.5.

The second automated metric we define is **PEARL**-Score. We observed that the IN MASK score does not consider the distance between the point and the mask. In other words, if the predicted location is not in the mask, it is treated the same as if the point was on the other side of the image. In addition, the score does not consider where the object is to be placed in the mask. To humans, it is more *natural* to place objects away from the edge of a surface. For example, one would not place a “cup” on the corner of a table; rather, they would place it where the “cup” is more centered and not at risk of falling. Therefore, we designed **PEARL**-Score to be negative when the placement location is outside the mask, and positive when it is inside the mask. We give more points if the placement is well within the mask, and subtract more points if the placement is far outside the mask. We divide this into three cases:

1. **Case 1:** If the 2D Coordinate $(\hat{x}_{ij}, \hat{y}_{ij})$ is in the mask, we find the closest point (x, y) outside the mask. Then, the **PEARL**-Score is the Euclidean distance between the two points $\mathcal{L}_2(x, y, \hat{x}, \hat{y})$ as defined in Equation 3.4. The farther away $(\hat{x}_{ij}, \hat{y}_{ij})$ is from the edge of the mask, the higher the **PEARL**-Score. A point near the edge of the mask would have a lower **PEARL**-Score.
2. **Case 2:** If the 2D Coordinate $(\hat{x}_{ij}, \hat{y}_{ij})$ is outside the mask, we find the closest point (x, y) inside the mask. Then, the **PEARL**-Score would be $-\mathcal{L}_2(x, y, \hat{x}, \hat{y})$. The score is negative because the placement should be penalized for being outside the mask.

(\hat{x}, \hat{y}) is the proposed placement location

$$\mathcal{V}(x, y) = \begin{cases} 1 & \text{if } (x, y) \text{ is inside mask} \\ 0 & \text{otherwise} \end{cases}$$

\mathcal{S} is the PEARL-Score

$$\mathcal{S} = (-1)^{1-\mathcal{V}(\hat{x}, \hat{y})} \min_{x, y: \mathcal{V}(x, y) = 1-\mathcal{V}(\hat{x}, \hat{y})} \sqrt{(\hat{x} - x)^2 + (\hat{y} - y)^2}$$

Case 1: (\hat{x}, \hat{y}) in mask	$\mathcal{V}(\hat{x}, \hat{y}) = 1$ $-1^0 = 1$	Min over points (x, y) outside mask	Euclidean distance between (\hat{x}, \hat{y}) and (x, y)
Case 2: (\hat{x}, \hat{y}) not in mask	$\mathcal{V}(\hat{x}, \hat{y}) = 0$ $-1^1 = -1$	Min over points (x, y) inside mask	Euclidean distance between (\hat{x}, \hat{y}) and (x, y)

Figure 3.7: **PEARL**-Score for one placement. The cases clearly define the importance of the individual terms in the **PEARL**-Score formula. The green box defines if the point is located in the mask. The yellow box minimizes over the points with respect to the mask. The red box calculates the euclidean distance.

3. **Case 3:** If the 2D Coordinate $(\hat{x}_{ij}, \hat{y}_{ij})$ is one the edge of the mask, then the **PEARL**-Score would be 0.

The **PEARL**-Score can be summarized in Equation 3.2, where:

- \mathcal{S} is the overall **PEARL**-Score on a set of \mathcal{N} images.
- \mathcal{M}_i is the number of objects being placed in image i .
- $\mathcal{V}_{ij}(x, y)$ is the value of the mask of valid locations for object j in image i at point (x, y) . Its value is 0 outside the mask and 1 inside the mask.
- $(\hat{x}_{ij}, \hat{y}_{ij})$ is the predicted placement location for object j in image i .

$$\mathcal{S} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \frac{1}{\mathcal{M}_i} \sum_{j=1}^{\mathcal{M}_i} (-1)^{1-\mathcal{V}_{ij}(\hat{x}_{ij}, \hat{y}_{ij})} \cdot \mathcal{D}(\hat{x}_{ij}, \hat{y}_{ij}) \quad (3.2)$$

$$\mathcal{D}(\hat{x}_{ij}, \hat{y}_{ij}) = \min_{\substack{x, y: V_{ij}(x, y) = \\ 1 - V_{ij}(\hat{x}_{ij}, \hat{y}_{ij})}} \mathcal{L}_2(x, y, \hat{x}_{ij}, \hat{y}_{ij}) \quad (3.3)$$

$$\mathcal{L}_2(x, y, \hat{x}_{ij}, \hat{y}_{ij}) = \sqrt{(\hat{x}_{ij} - x)^2 + (\hat{y}_{ij} - y)^2} \quad (3.4)$$

$\mathcal{D}(\hat{x}_{ij}, \hat{y}_{ij})$, defined in Equation 3.3, is the minimum distance between the predicted placement location and a point on the edge of the mask. The score for an image is calculated by averaging the scores of all object placements within the image. Finally, the overall **PEARL**-Score \mathcal{S} is computed by taking the average score across all images. Figure 3.7 highlights the individual components of the **PEARL**-Score.

Human Evaluation

To verify the agreement of our automated metrics with human preferences, we supplement our results with a human evaluation. To gather opinions from a general audience, we conducted an Amazon Mechanical Turk (MTurk) study to evaluate OCTO+, OCTO-PUS, and three of the other placement techniques we implemented that performed the best on our automated metrics (Table 3.4).

To evaluate each placement method, we used them to generate 2D placements for 100 randomly selected image-object pairs. We divided the 100 images into Human Intelligence Tasks (HITs), each containing five images, for a total of 20 HITs for each of the evaluations listed above. For the baselines (expert *unnatural* and random), we only ran the assessment on 50 images, or 10 HITs.

In each HIT, evaluators were told what object was to be placed and were shown two images side-by-side. Both images were annotated with a red circle indicating a proposed placement location. One of the proposed placement locations came from the evaluated method, while the other was a *natural* placement annotated by an expert. The evaluators then selected which placement location was superior or declared a tie if both locations



Method	Tagger	Filter	Selector	Locator
OCTO+ 	RAM++	G-DINO	GPT-4	G-DINO
	RAM++	G-DINO	GPT-4	CLIPSeg
OCTOPUS 	SCP	ViLT	GPT-4	CLIPSeg
	GPT-4V	—	—	G-Dino

Table 3.4: Human Evaluation is performed in the following methods

were deemed equally appropriate for the object. The evaluators did not know which method produced each placement location.

To ensure our data was of high quality, we specified the following criteria workers must meet to work on the task:

- Only allow workers with 95%+ HIT Approval Rate
- Only allow workers with 50+ HITs approved
- Only allow workers from regions in US & UK
- Added a sixth side-by-side comparison of an obvious *good* vs *bad* placement as an ‘attention check’.

In Table 3.6, the results of the Mechanical Turk study are shown in the column labeled MTURK. Suppose a placement is tied with a *natural* placement. In that case, it must be natural, so to compute the metrics shown in the table, we added the proportion of the time that the method in question won or tied against the *natural* placement.

The results showed that GPT-4V with CLIPSeg as the locator performed best, tying or winning against the expert *natural* placements 58.2% of the time. GPT-4V with G-SAM was second place, followed by the RAM++ variants (which include OCTO+), with OCTOPUS (SCP + ViLT + GPT-4) performing the worst out of the methods that we ran an evaluation for.













Tagger 	Filter 	Pipeline 		Automated Metrics 	
		Selector 	Locator 	In Mask 	Score 
Baselines					
_____	_____	_____	Natural	0.907	17.987
_____	_____	_____	Random	<u>0.161</u>	<u>-106.113</u>
_____	_____	_____	Unnatural	0.010	-176.375
Selected Tag + Image as Input					
SCP	CLIPSeg	GPT-4	CLIPSeg (Max)	0.572	-20.730
 SCP	ViLT	GPT-4	CLIPSeg (Max)	0.588	-15.300
SCP	G-DINO	GPT-4	CLIPSeg (Max)	0.596	-13.005
SCP	CLIPSeg	GPT-4	G-SAM (Center)	0.613	-10.783
SCP	G-DINO	GPT-4	G-SAM (Center)	0.615	-6.464
_____	_____	LLaVA-1.5	CLIPSeg (Max)	0.649	-13.17
RAM++	G-DINO	GPT-4	CLIPSeg (Max)	0.671	-4.185
_____	_____	GPT-4V	G-SAM (Center)	0.686	<u>4.317</u>
_____	_____	GPT-4V	CLIPSeg (Max)	<u>0.692</u>	-4.492
 RAM++	G-DINO	GPT-4	G-SAM (Center)	0.702	7.634
Object + Image as Input					
_____	_____	InsPix2Pix	G-SAM (Bottom)	<u>0.283</u>	<u>-60.852</u>
_____	_____	_____	GPT-4V (Pixel)	0.321	-34.282

Table 3.5: Stage 3 Automated Metrics. The 3 baselines (natural, random, unnatural) denote the placements. InstructPix2Pix is abbreviated as InsPix2Pix for brevity. For locators column, (max) denotes selecting the pixel with the maximum intensity, (center) denotes the center of the bounding box, (bottom) denotes the bottom center of the bounding box, (pixel) denotes the pixel location was provided by GPT-4V directly. The best metric in each category is **IN BOLD** and the second best is UNDERLINED.  is the OCTO+ method and  is OCTOPUS.

Tagger 🗡️	Filter 🏹	Pipeline 📡		Human Evaluation 👤	
		Selector ✓	Locator 📍	MTurk ⬆️	Expert ⬆️
Baselines					
		Natural Placement		1.000	1.000
		Random Placement		<u>0.467</u>	<u>0.040</u>
		Unnatural Placement		0.167	0.020
Selected Tag + Image as Input					
🌟 SCP	ViLT	GPT-4	CLIPSeg (Max)	0.514	0.570
RAM++	G-DINO	GPT-4	CLIPSeg (Max)	0.514	————
————	————	GPT-4V	G-SAM (Center)	<u>0.580</u>	————
————	————	GPT-4V	CLIPSeg (Max)	0.582	<u>0.620</u>
👑 RAM++	G-DINO	GPT-4	G-SAM (Center)	0.527	0.690

Table 3.6: Stage 3 Human Evaluation Metrics. The 3 baselines (natural, random, unnatural) denote the placements. For models, we select the 5 best performing methods from Table 3.5 results. The best metrics in each category is **IN BOLD** and the second best is UNDERLINED. 👑 is OCTO+, 🌟 is OCTOPUS

Looking at our baselines, the expert unnatural placements won or tied with the expert natural placements 16.7% of the time, which is higher than expected and could indicate data noise. We also performed an expert-level human evaluation to mitigate noise in the data. Two of our team members evaluated OCTOPUS, OCTO+, the best performing GPT-4V method, and the baselines (random and *unnatural*). The format mirrored the MTurk study, displaying evaluators two side-by-side images: one generated using the evaluated method and the other annotated by experts with *natural* placement. This process was performed on 50 randomly selected object-image pairs for baselines and 100 for other comparisons.

The results show in EXPERT column of Table 3.6, reveal that in the judgment of the two evaluators, 69% of the time, OCTO+ selected a location at least as natural as the human expert selecting a *natural* location. Comparing this with GPT4-V, only 62% of the time and SCP only 57% of the time did the human experts select a natural location. The experts’ natural locations won over the random and unnatural locations 96% and

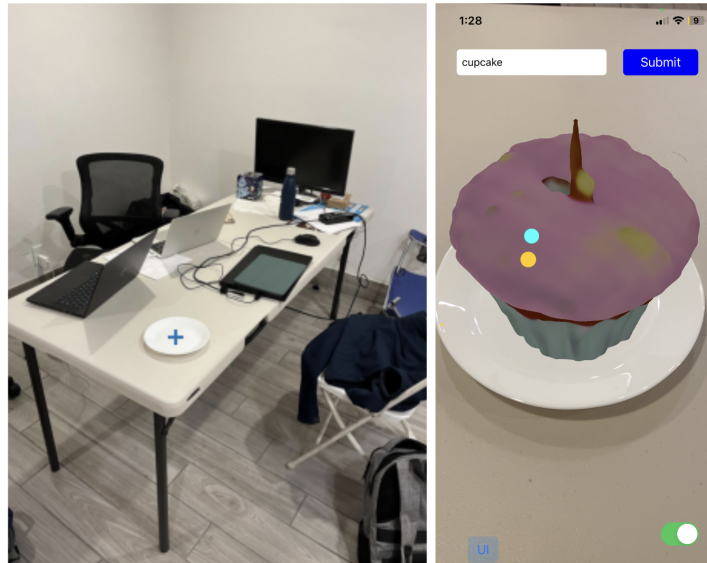


Figure 3.8: **Left:** The 2D location in the image selected. **Right:** A screenshot of the 3D scene with a virtual cupcake placed on the plate. Both the 2D and 3D locations were found as described in the Locating section.

98% of the time, respectively, which confirms that they were indeed appropriate locations the vast majority of the time, demonstrating that it is tailored to human preferences and not far off from the expert’s *natural* placements.

In summary, as documented by Table 3.5 and Table 3.6, OCTO+ performs the best in three of the four evaluation studies evaluated and remains competitive in the fourth evaluation study.

3.5 Discussion and Limitations

While our method generally places objects naturally, it has limitations. First, it takes up to 10 seconds to generate a single placement location on an NVIDIA RTX A4000, which means it is not yet truly practical in real-world applications, in particular when making live queries with AR cameras. We specifically tested it in such a use case (see §3.6 below), and the latency may pose an inconvenience to users. Additionally, while

our method excels at selecting the optimal entity for virtual object placement and aims for a centrally located point on the surface, the outcome is not always the most *natural*. For example, it is conventional for people to hang paintings at eye level, a consideration our pipeline currently lacks. Our pipeline can also struggle when the surface selected is a complex shape. For example, if our pipeline determines that a cat should be placed on a couch, it will not consider that the cat would most naturally be placed on the seat, and may select a location on the backrest or side instead. This could potentially be addressed with complete consideration of the 3D model of the scene, which would enable us to restrict placement locations to horizontal or vertical planes, depending on the object being placed. Enhancing 3D reasoning with vision-and-language models will eventually yield even better results.

3.6 Application

To demonstrate the practical purposes of our model, we created an iOS AR application that uses OCTO+, as shown in the right image of Figure 3.8. The app takes a text prompt as input to convert this text prompt into a 3D model and then places the model in the scene at a natural location. For example, if the text input was “a cupcake” and the real-world scene contained a pair of shoes, a backpack, and a table with a plate on it, then our app would generate a 3D model of a red cake and place it on the plate. We use ARKit [86] to track the device and identify planes in the image on which rays can be cast and SceneKit [87] to render 3D models. We also have a backend where the two crucial steps, **Object Generation** and **Object Placement** are performed offline.

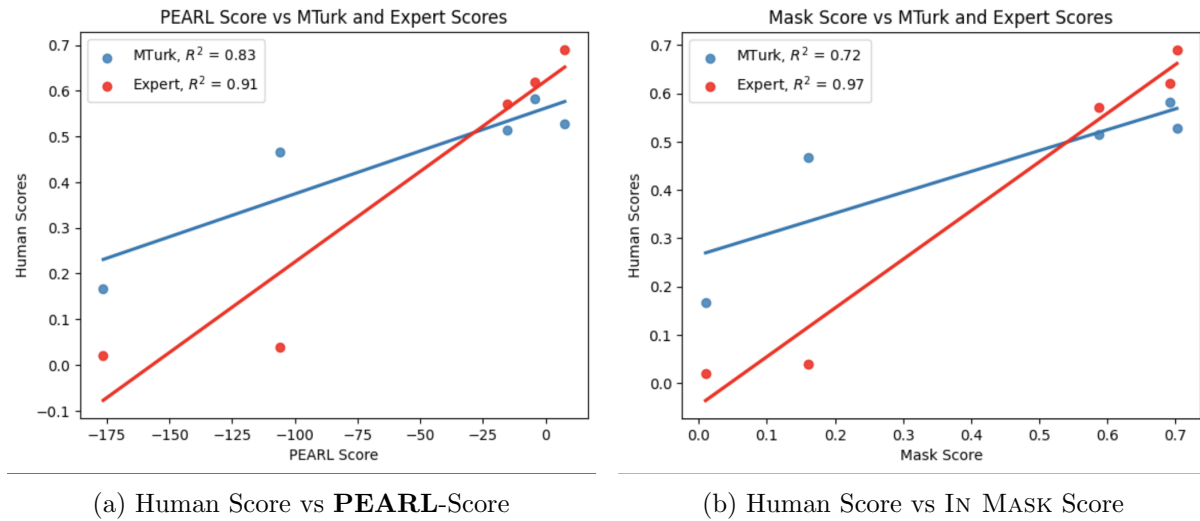


Figure 3.9: Comparison of **PEARL** Metrics alignment with Human preferences. **PEARL**-Score and IN MASK score are well aligned to human score, as observed by the **strong positive** correlation between automated metrics and human scores.

3.6.1 Object Generation

To generate a 3D model of the text prompt, we use OpenAI’s Shap-E [88], a text-to-3D diffusion model. Shap-E can take in any text as input, so it is not restricted to a specific list of objects like a library of 3D assets would be. Among text-to-3D models, we choose Shap-E because it is fast (20 seconds on an NVIDIA RTX A4000) and generates acceptable-quality models.

3.6.2 Object Placement

Our pipeline takes the camera frame from when the user submits their input text prompt and returns the 3D world coordinates to place the object. Once the object is placed in the scene, the user can see it and look around it. Multiple objects can be generated to give a room a theme. For example, to create a Halloween-themed room, the user could create pumpkins, skeletons, and candy. This AR experience can enable users to design their own themed virtual environments.

3.7 Conclusion

In this chapter, we present OCTO+, a state-of-the-art method for placing virtual content in augmented reality. OCTO+ outperforms its successor OCTOPUS and the state-of-the-art multimodal model GPT-4V in three of the four metrics observed in this chapter. The OCTO+ pipeline is built using RAM++ as the image-tagging model, G-DINO as the filter, GPT-4 LLM as the reasoner to select the best object in the image, and G-SAM as the locator to choose the more natural 2D location. The entire OCTO+ pipeline is open-vocabulary, meaning it can be used to place *any* object in *any* scene out of the box without *any* fine-tuning. We also present **PEARL**-Score, an automated metric aligned to human preferences Figure 3.9. **PEARL** introduces a challenging benchmark in virtual content placement in augmented reality.

In future work, we would like to accelerate further the placement determination (which currently takes upwards of 10 seconds) to enable truly interactive Mixed Reality experiences. Also, we are exploring relaxations of the placement phrasing, which presently only considers placing objects *on* elements visible in the picture. Other prepositions (e.g., “above” and “near”) could be considered. Object orientation could also be specifically specified and addressed (e.g. “facing the camera/facing the window”). LLMs can adeptly manage complex, vague, and multi-object spatial directives (e.g., “add paintings and poster to this room” or “add Christmas decorations”). Future research can extend their capabilities in handling such directives.

3.8 Acknowledgements

The contents of this chapter is a result of a collaboration with Luke Yoffe. This project was advised by Tobias Höllerer. This work has previously appeared in 2024 IEEE

International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR).

Chapter 4

Long-Context VLM Reasoning

4.1 Introduction

Long-context vision-language models (VLMs) which accept multiple image inputs are pushing the boundaries of multimodal reasoning applications, but evaluation methods have not kept up. As open-weight long-context language models (LMs) have been around for the last few years [89], researchers have developed robust evaluations to compare their capabilities against those of proprietary LMs [19, 90]. However, until very recently [91], the small set of long-context VLMs supporting multi-image input have been inaccessible to the public [?], so evaluation practices for long-context VLMs [92, 93, 94] are in their infancy [95]. Most reasoning-based VLM evals have analogues in text-domain LM tasks. For example, image captioning [96, 97, 98] can be likened to text summarization, and visual question answering [99, 100, 101] to text QA.

We aim to bridge gaps in multi-image VLM evaluation by drawing analogies to established long-context LM tasks. Long-document QA [102] is an easy-to-evaluate test that directly showcases model performance on a useful application, and hints at its potential in retrieval-augmented generation (RAG) more broadly. Central to these evaluations is

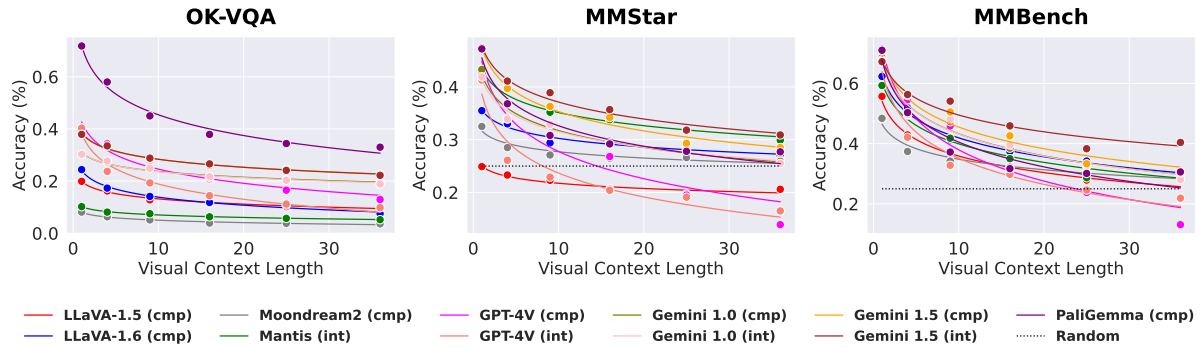


Figure 4.1: The impact of visual context on vision-language models (VLMs) in our modified, multi-image versions of the OK-VQA, MMStar, and MMBench evaluation benchmarks. *Distractor images* around the target image increase the *visual context length* needed to answer the questions. VLM performance exhibits an *exponential decay* as distractors increase, evident in both single composed (cmp) and multiple interleaved (int) image configurations.

the model’s ability **to recognize which elements in a lengthy input are necessary for answering a query and which are not**, a skill we term *extractive reasoning*. Do long-context VLMs also exhibit these extractive reasoning capabilities? While current VLM benchmarks align with long-document summarization tasks, such as whole-video question answering [103] or summarization [104], no existing benchmarks effectively capture a VLM’s ability to **filter out irrelevant images in a long context to reason or answer a query**—essentially, perform extractive reasoning over images.

Measuring extractive reasoning over image sequences is crucial. Just as extractive textual reasoning facilitates text-domain RAG, multi-modal RAG demands that VLMs efficiently reason over and extract information from documents featuring multiple images. Similarly, video QA necessitates that models focus solely on frames containing relevant information, much like long-document QA.

We introduce LoCoVQA, a benchmark generator for **Long Context Visual Question Answering (VQA) with distractors**. LoCoVQA enables the creation of long-context image understanding evaluations from any image comprehension dataset by presenting

a sequence with the question-relevant image alongside a configurable set of visual distractors. This allows us to accurately assess how effectively VLMs extract **only the pertinent information to a query within a cluttered context**.

We find that even top proprietary VLMs struggle with this capability, even over short visual contexts, likely due to fundamental deficiencies in their training objectives. By unveiling this LoCoVQA identifies a crucial area for performance improvement in future VLMs. In summary, we:

- Evaluate long-context extractive reasoning across a wide array of VLMs using LoCoVQA.
- Find that all existing VLMs exhibit significant fragility with increasing distractor context sizes.
- Use LoCoVQA to create a *visual needle in a haystack* test, revealing substantial positional biases in SOTA VLMs in extractive reasoning.

4.2 Related Work

Text-based long-context tasks such as long-document question-answering (QA) [105], summarization [106], and retrieval-augmented generation (RAG) [107, 108] offer analogues to long-context reasoning in VLMs. Many of these tasks (e.g., QA, RAG) require language extractive reasoning, as discussed above.

Few VLM long-context evals measure extractive reasoning. Existing VQA benchmarks involving *distractor information* (thereby extraction) do not focus on long contexts, and long-context VQA benchmarks do not involve distractors. MultipanelVQA [109] includes distractor information, but only within single input images. Some VLM evaluations focused on hallucinations indirectly capture a notion of distraction [110, 111]

but do not explicitly measure it alone. MILEBench [112] evaluates VQA in long contexts, but only on non-distractor tasks such as video summarization or difference detection, where all inputs are relevant.

Needle-in-a-haystack evaluation tasks (asking models to recover hidden passphrases at various positions) do require long-context extraction, but reasoning is not involved. Gemini 1.5 [19] extended this task to video comprehension by hiding a simple passphrase in many positions within a video. Wang et al. (2024) [113] present a static benchmark concurrently to ours centered on multimodal needle-in-a-haystack.

4.3 LoCoVQA Generation Method

Samples synthesized through LoCoVQA contain one or more *content* images X corresponding to a question and answer pair (Q, A) . Content images can be sampled from various image comprehension benchmark, such as OK-VQA [114], MMStar [115], MNIST [116], and others. Alongside the content image(s), each sample includes up to 35 *distractor images*, which are either sampled in-distribution from the content image set (ensuring no content image collisions to prevent ambiguity, as described in §4.3.1) or out-of-distribution from other image sets such as MS COCO [117].

Samples can be arranged as **interleaved** image sequences for VLMs that accept multi-image inputs or as **composite** images arranged in a grid, as depicted in Figure 4.2. For all models capable of ingesting interleaved sequences, we evaluate both interleaved and composite examples.

Visual context refers to the visual elements within an image or sequence of images that are relevant to answering a question. This includes the *content* image and distractor images. The challenge is to identify and focus on the pertinent details while ignoring irrelevant information. We can sample images both in-distribution to create challenging

visual contexts or out-of-distribution to isolate extraction capabilities in the simplest setting.

Although this method can be applied to any vision-language dataset, we will describe the three specific tasks for which we generated long-context visual distractor benchmarks.

4.3.1 Single-image Reasoning Tasks

First, we discuss the visual reasoning benchmarks which we expand into long-context samples containing one content image per sample. For most question answering and visual reasoning tasks, this is the only LoCoVQA expansion that makes sense: few VQA samples can be combined such that a plausible new QA pair requiring information from multiple images. Since most interleaved models we evaluate support 36 or fewer images as sequential inputs without modification, we do not evaluate any models with context lengths beyond 36. However, LoCoVQA scales to any size. For the single-context reasoning tasks, we exclusively sample the distractors in-distribution.

OK-VQA

OK-VQA [114] is a single-image visual question answering dataset containing 5,072 question-answer-image triples. It requires external knowledge to reason beyond the image. LoCoVQA generates in-distribution long-context OK-VQA samples, ensuring that no content images have concept collisions that may complicate evaluation. For instance, the question about a character on top of a cake, as shown in Figure 4.2, is sampled from OK-VQA.

Since OK-VQA is an open-domain, free-form answer dataset, we score the samples using three metrics: exact match (full score if the model’s response contains any ground truth answer as a substring), and continuous text generation metrics BERTScore [5] and

Image (X) – Randomly chosen from dataset \mathcal{D} .
 Question (Q) – What cartoon character is on this cake?
 Answer (A) – Winnie the Pooh



Figure 4.2: Example of Image (X) corresponding to question-answer pair (Q, A) under increasing visual context lengths in the composed setting. *The green box is for illustration purposes; not included in model inputs.*

ROUGE_L [6] between candidates and references.

MMStar

MMStar [115] is a multi-domain visual question answering dataset combining examples from various existing datasets: 424 questions from SEEDBench [118], 366 questions from MMBench [119], 100 questions from MMMU [120], 345 questions from MathVista [121], 69 examples from ScienceQA [45], and 196 examples from AI2D [122]. MMStar contains 1,500 high-quality multiple-choice questions that *require visual information from the images* to answer, a filtering step not initially performed on the source datasets. For example, over 50% of ScienceQA questions can be solved by a text-only LLM [115]. Similar to OK-VQA, we generate LoCoVQA samples for MMStar using the collision filtering technique to produce pseudo-documents composed of multiple example images.

As a multiple choice dataset, scoring MMStar is more straightforward. Full details on how we faithfully extract multiple choice answers from the models is provided in §A.3.4.

Filtering Collisions in LoCoVQA

To address the problem of *content-distractor collisions*—where multiple similar in-distribution images in the visual context make the QA pair ambiguous—we implement a robust LM-based filtering method. For each visual context image, we prompt GPT-4 to list the top five entities; if there is overlap, we consider the question potentially ambiguous. Detailed implementations and examples of our filtering method are provided in §A.3.3. To validate this approach, we manually assessed a subset of LoCoVQA generations and found it to be consistent, with no such collisions.

4.3.2 Multi-image Reasoning Tasks

In §4.3.1, we explored tests designed to evaluate whether VLMs can extract a single relevant image from a sequence to answer a query, thereby probing their long-context reasoning capabilities. Extending this to test how well VLMs can extract information from a multi-image sequence is a natural progression. However, VQA examples cannot be easily combined in a way that requires multiple images to answer a single question. Therefore, we turn to constructing “sequential VQA” sets using synthetic tasks. Optical character recognition (OCR) is a straightforward task to convert into multi-image question answering by including multiple OCR examples as interleaved images and asking the VLM to list all the text.

MNIST

We use MNIST [116] as it is a canonical dataset for OCR. For a desired visual context length, we sample between 1 and 8 randomly-colored digits from the MNIST training set of around 60K images, resizing them to between 1/6 and 1/2 of the maximum height of other context images. The remaining distractor images are randomly sampled from a subset of 5K high-quality MS COCO [117] validation images. The VLM is then prompted to *list all handwritten digits present in the sequence*.

By varying the number of digits in the sequence, we can dynamically adjust the difficulty level of the multi-image distractor OCR task. Figure 4.3 illustrates examples with 1, 4, and 8 digits in a 9-image context. An output is considered correct only if the stored string of generated digits exactly matches the ground truth, with no partial credit.

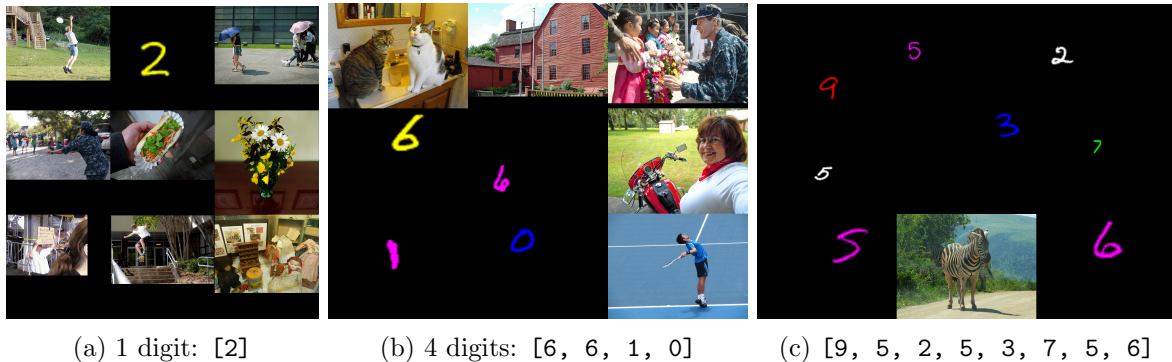


Figure 4.3: Each subfigure represents a variable number of MNIST digits (1, 4, 8) while maintaining a context length of 9 images.

4.4 Experiments

We evaluate the performance of nine current vision-language models on our LO-CoVQA-generated benchmarks. The open-weight models tested are Moondream2 [123], LLaVA-1.5 [79], LLaVA-1.6 [124], PaliGemma-3b [125], and two Mantis variants [91]. Additional details are provided in §A.3.1. Both Mantis variants are VLMs further tuned on interleaved multi-image instruction following. The three proprietary models we evaluate are GPT-4V [78, 18], Gemini 1.0 Pro Vision [126], and Gemini 1.5 Flash [19]. Table 4.1 showcases all the models along with the model context sizes and image downsampling.

Interleaved image support. All closed-source models and Mantis support multi-image interleaved inputs, while the others only support single images. For multi-image models, we evaluate both the **composed (cmp)** & **interleaved (int)** settings. For single-image models, we only test the **composed (cmp)** setting.

Downsampling images. Some models accept images of arbitrary resolution, while others automatically downscale inputs. This difference is crucial, especially in the **cmp** setting, as increasing the visual context can lead to information loss. For the downsampling models that support both **cmp** and **int** settings, we assess both. Any performance

Vision-Language Model	Context	Base LM	↓sample
<i>Single-Image Input VLMs</i>			
Moondream2-1.6B	2K	Phi-1.5	✓
LLaVA-1.5-7B	4K	Vicuna	✓
LLaVA-1.6-7B	8K	Mistral	✓
PaliGemma-3B	8K	Gemma	
<i>Multi-Image Input VLMs</i>			
Mantis-Baklava-7B	8K	Mistral	✓
Mantis-Idefics2-8B	8K	Mistral	✓
Gemini 1.0 Pro Vision	32K	Gemini	
GPT-4 Vision	128K	GPT-4	
Gemini 1.5 Flash	1M	Gemini	

Table 4.1: Overview of open-source and proprietary vision-language models (VLMs).

differences between these settings would highlight the impact of downsampling on the **cmp** setting.

4.5 Results

Figure 4.4 illustrates how model performance (across 10 experiments, in both composed and interleaved settings) changes with increasing visual context lengths on single-image LoCoVQA tasks. The first two rows displays results from MMStar dataset, which consists of a subset of six other datasets, with titles highlighted in and random guess thresholds indicated by dotted black lines. The bottom row presents OK-VQA scores using three scoring metrics, with titles highlighted in .

Across all models on OK-VQA and the majority on MMStar subsets for SeedBench, MMBench, ScienceQA, and AI2D, we observe *striking exponential decay trends* in model performance with increasing visual context length. Correlation coefficients, r^2 , and p -values for each trendline are reported in Table A.8, §A.3.4. In general, the closed-source models outperform their open-weight counterparts, especially on multiple-choice tasks. Among the open-weight models, PaliGemma performs the best, likely due to its substan-

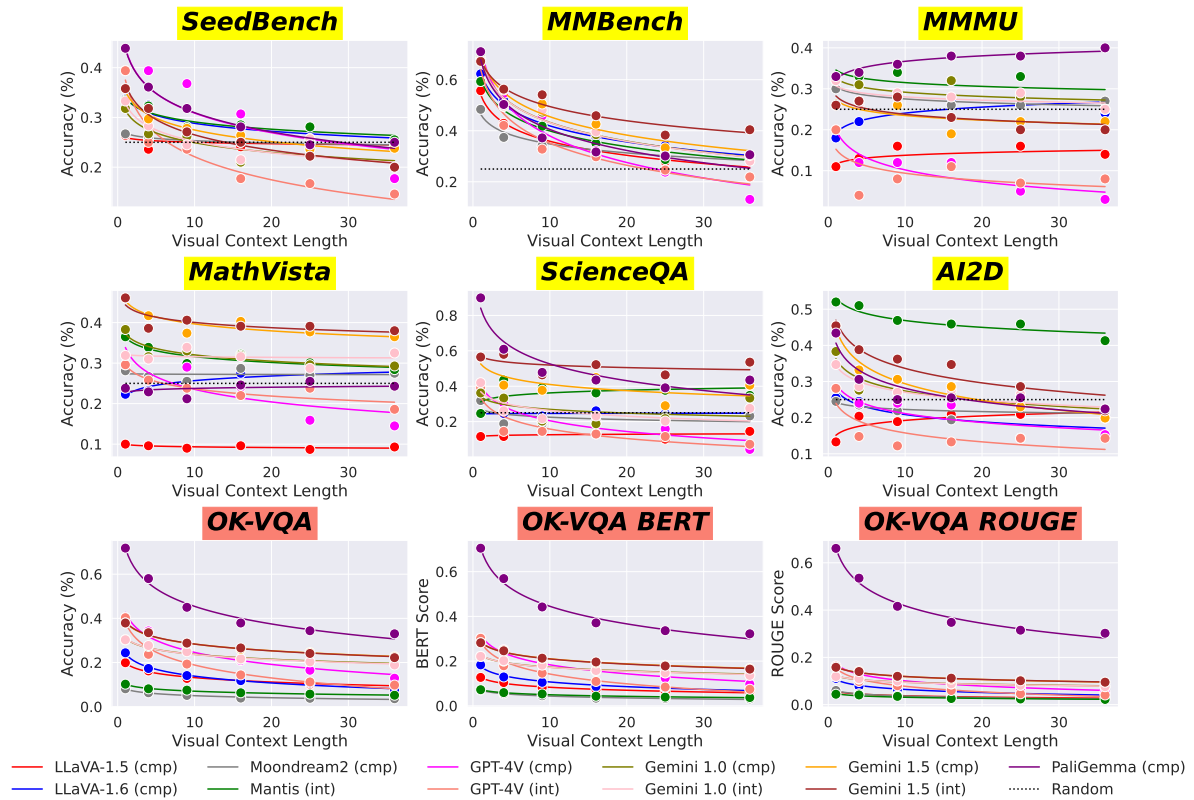


Figure 4.4: VLM Performance on MMStar and OK-VQA. Note the clearly declining exponential fit trends for many of the models. The (model, task) pairs for which these trends do not hold by and large are *below the random baseline*.

tially more extensive training on vision-language tasks compared to its counterparts like LLaVA or Mantis.

On several multiple-choice tasks, Mantis (the interleaved open-weight model) outperforms the other open-weight models that rely solely on composite inputs. This is likely due to the image subsampling required by several of the other open-weight models negatively impacting performance when using composite inputs. However, on the OK-VQA task, Mantis and Moondream are the least performant, even at low context lengths. These models were likely not trained on visual question-answering tasks to the same extent as LLaVA variants during their instruction tuning steps.

The most noteworthy point is this: **the logarithmic decay trend holds equally**

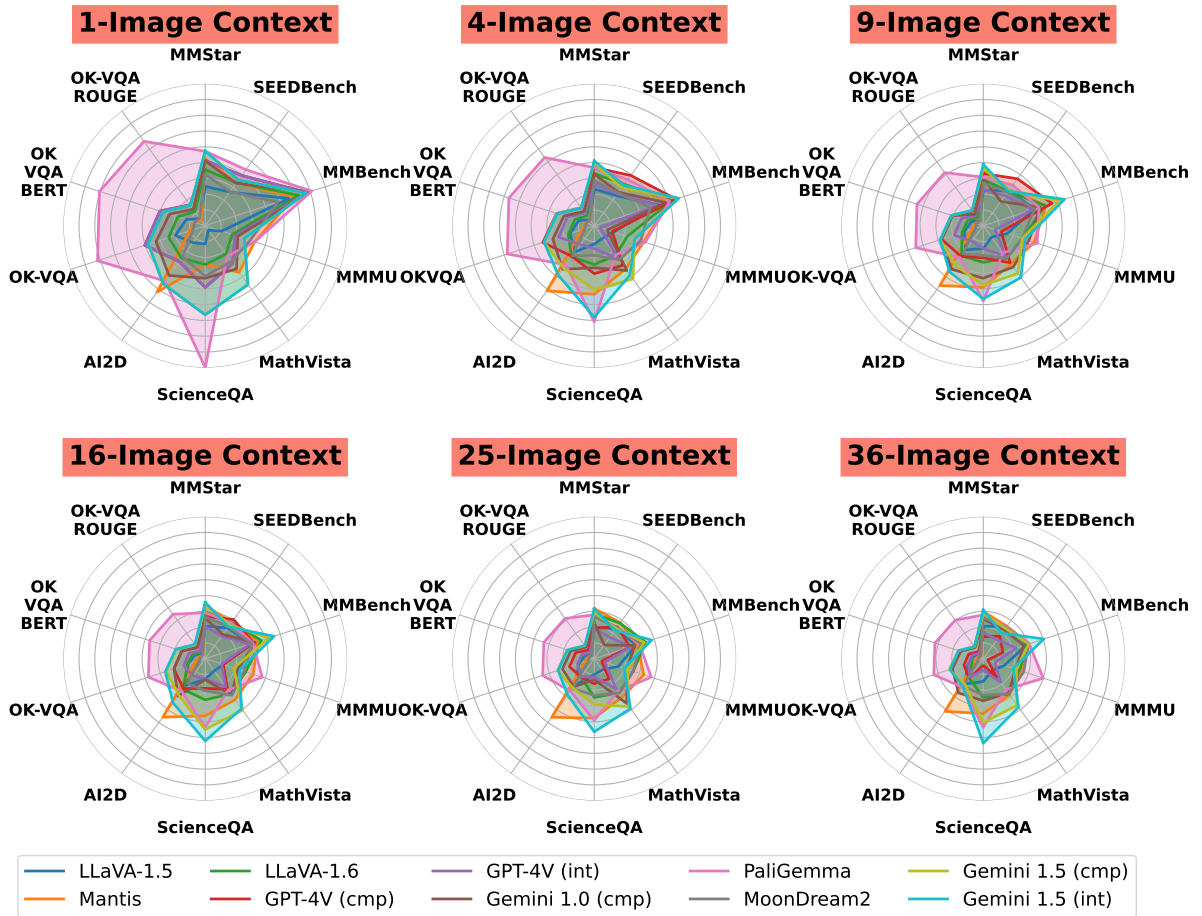


Figure 4.5: Analyzing GPT-4V performance across 8 multi-modal benchmarks with varied visual context lengths.

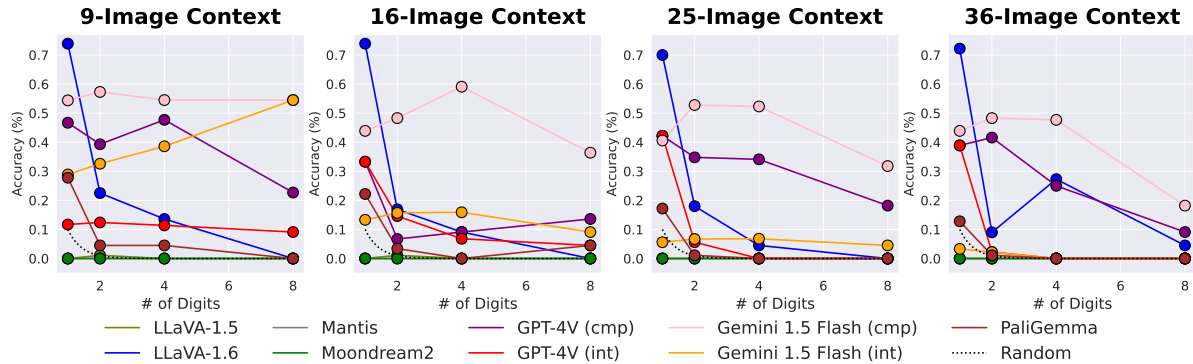


Figure 4.6: VLM Performance on the MNIST-Digits transcription task as a function of # of digits to transcribe. *These plots have a different x-axis than the plots in Figure 1 and Figure 4.4: rather than the relationship between context size and performance, we are assessing the relationship between "task difficulty" and performance, at four context sizes.*

well in interleaved and composed settings. This indicates that the performance-visual context length trend is fundamental and **cannot be attributed solely to down-sampling effects in the composite setting.** Furthermore, for the models tested in both conditions (GPT-4V and Gemini), the same trends are observed in both interleaved and composite settings. Sub-chance performance on the multiple-choice question-answering datasets, particularly for the closed-weight models, occur due to high refusal rates. This may illustrate how "alignment hampering" can hinder performance.

Taskwise Performance by context length. Figure 4.5 illustrates the models' performance at context lengths of 1, 9, and 25 for each task, to compare the relative advantages different models have on various tasks. For example, PaliGemma excels on OK-VQA compared to the other models, while Mantis performs well on AI2D. These differences are likely due to variations in training tasks.

4.5.1 Performance on Multi-image Tasks

Figure 4.6 presents model performance on the MNIST-Digits transcription task based on the number of digits to transcribe. While there is a trend of decreasing performance with an increasing number of images, it does not follow a simple pattern like the context-length trend. For example, Gemini 1.5 Flash experiences minimal performance degradation even as the target digit length increases.

Characterizing difficulty as a function of digit length is complex—as digit counts increase in a fixed context window, *the the output label search space grows*, while the *ratio of relevant images to irrelevant images increases*. This may explain why some models have *consistent* or even *increasing* performance as # digits increases. Analyzing why different models handle these axes of difficulty differently is an interesting future direction.

However, akin to the single-image tasks, increasing the overall *visual context length* (seen in same-color, x value points across plots), makes the task more difficult, albeit without as clear a correlation.

4.6 Ablation: Needle in a Haystack

As performance within context windows decreases, a natural question arises: **Are performance failures equally distributed across the visual context range?** To investigate this, we adapt our MNIST-based visual context OCR task into a **visual needle in a haystack** task. Needle in a haystack tests are a common minimal test of long-context capabilities in LMs [127], involving hiding a passphrase, such as the word “needle,” in various positions within a long document and asking the model to retrieve it. To adapt this concept, we modified our single-digit MNIST task by sampling a set of 10 colored MNIST digits (one for each number, e.g. blue 3, green 7, etc.) and hiding them

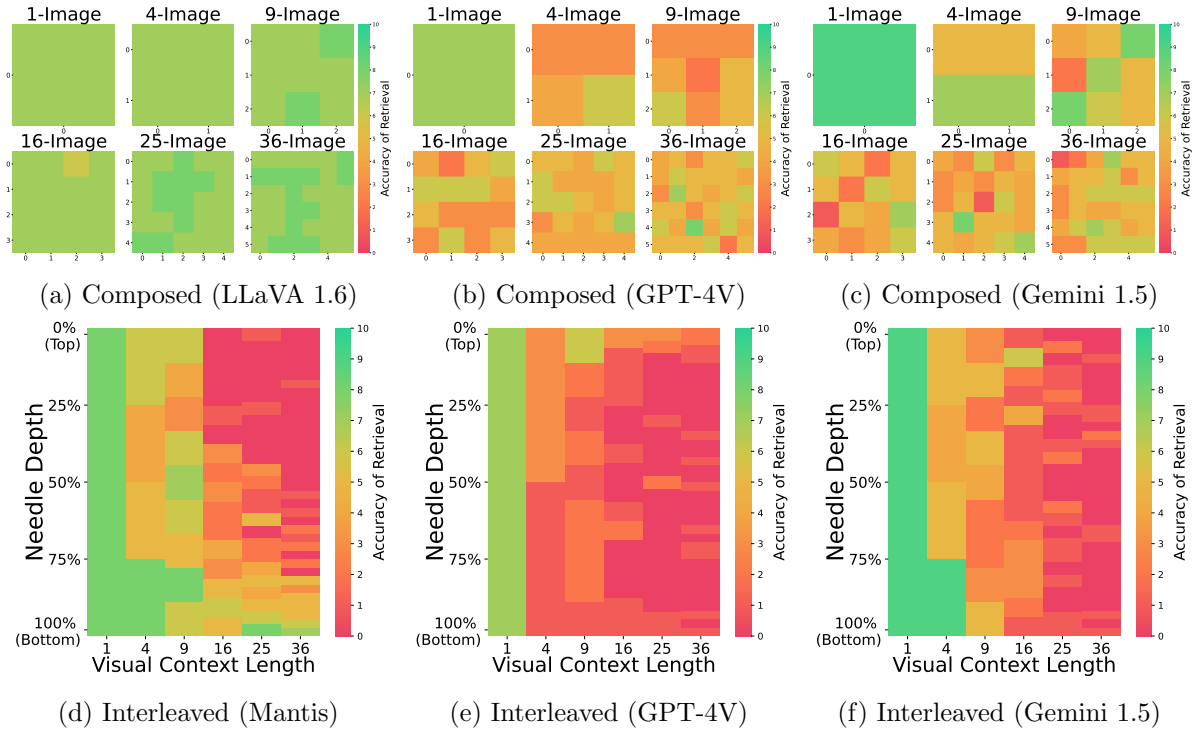


Figure 4.7: Evaluation of the Visual Needle in a Haystack task using GPT-4V, best-performing VLM, conducted under both composed and interleaved haystack settings. Retrieval accuracy measures the frequency of correct answers produced by the VLM, in this case identifying the MNIST digit. In the composed setting, retrieval accuracy is measured by placing the needle at each cell. In the interleaved setting, needle depth signifies the position within the image sequence, with 0% representing the first image and 100% representing the last. Our evaluation highlights a consistent decline in retrieval accuracy as the visual context length increases.

in each possible position within both interleaved and composed visual context sequences.

By assessing a model’s single-digit recovery rate at each position, we can identify any systematic bias in extraction capabilities by position. Figure 4.7 presents the performance of a subset of the tested VLMs on the composed and interleaved MNIST visual needle in a haystack tasks. We test GPT-4V and Gemini 1.5 Flash in both settings, and treat LLaVA 1.6 and Mantis as analogous methods for the interleaved and composed settings.

Across the three composed tests (subfigures a, b, and c), we do not find a systematic

bias toward any particular position on the single-digit recognition task. Surprisingly, LLaVA 1.6 performance the best, probably due to more MNIST-like OCR data in its training mix compared to the others.

However, *we find strong systematic biases with respect to position in the interleaved setting for all models* (subfigures d, e, and f). Mantis shows a preference for late positions in sequences of any length, while GPT-4V and Gemini exhibit weak biases towards early positions. Given that the biases are evident even in the relatively simple MNIST-base OCR task, it is likely that this effect plays a significant role in the performance penalty these models experience with longer contexts.

These results are supported by similar findings in from the image-needle test from Wang et al. (2024) [113]. They also found that—contrary to Google’s claims to significant needle-in-a-haystack performance for Gemini Reid et al. (2024) [19]—for non-trivial hidden visual information tasks current SOTA VLMs are woefully non-performant.

4.7 Discussion

Why do we observe such striking logarithmic decay in performance with increasing visual context length? The fact that the trend is consistent in both interleaved and composite settings that downsampling effects in the latter are not the primary cause of performance decay. There may be an information-theoretic interpretation of this behavior: an increasing signal-to-noise ratio. As the visual context length increases, the amount of signal (relevant information) remains constant, while the amount of noise (distractor information) increases.

However, this signal-to-noise ratio problem also exists in text-only tasks, yet long-context LLMs perform well on needle in a haystack and long-context QA and summarization tasks, underlying their performance in retrieval-augmented generation. When it

comes to visual contextual tasks that involve distractor information, VLMs struggle to perform even on the easiest long-context tasks, such as transcribing MNIST characters. They are even less capable on more complex, real-world tasks like distractor VQA. The core issue appears to be that both short-context and long-context VLMs are not trained on tasks requiring attention over multiple context images to retrieve information. In contrast, for long passages of text, information from throughout the passage remains relevant—referring back to a specific sentence early in a document is intrinsic to the LM training objective.

However, the same may not necessarily be true for the sequential image language modeling task. In the existing interleaved vision-language training corpora, images are likely to be followed immediately by their relevant text. As a result, attending to much earlier images in the documents may not be crucial for achieving low LM loss. Additionally, no open VLM is trained on image *generation* tasks conditioned on text while updating the core transformer weights. Overlooking this objective may prevent VLMs from robustly modeling the relationship between images and text, which could be crucial for performing well on these tasks.

Interesting examples of poor performance Our analysis unveiled a few particularly surprising results. For example, contrary to our expectation that models capable of handling interleaved input would perform better with it than with composite input, we found that GPT-4V actually favors composite input on many subtasks. Additionally, to our surprise LLaVA 1.6 performed much better than GPT-4V on the composite haystack task at all sizes. This outcome was driven by multiple factors: GPT-4V tends to refuse some tasks rather than guessing, while LLaVA always provides a guess. Furthermore, GPT-4V often “hallucinates” multiple numbers instead of just one.

Possibility of memorization Some of the surprising performance results are driven by the training mix. For example, Mantis’s significant lead on AI2D, and LLaVA’s strong performance on the MNIST task may result from having more relevant data or those specific training sets included in their data. However, even considering that some models were trained on these tasks, the pronounced drops in generalization performance as the context length increases are even more striking. This illustrates that current VLMs fundamentally struggle to attend image sequences as well as they do with text.

Upper bound for video QA A more construct-valid test for real-world visual extractive reasoning would be QA over a long video where only a sub-element is required. E.g., putting the entire film *Star Wars* as input for QA pair (Q: “which character shoots a green alien in the Mos Eisley cantina?”, A: Han Solo). Information from a single scene must be extracted to perform this task, thereby requiring extractive reasoning. However, it is plausible that sequences of related images, unlike sequences of unrelated images, are more in-distribution for long-context VLMs and extractive tasks involving them could be easier for them to solve. By providing completely unrelated images, our task may be harder than video-based VQA, and may represent an upper-bound for visual extractive reasoning.

4.8 Conclusion

Vision-language models struggle with long-context visual extractive reasoning. Many models fail to extract necessary information from even short contexts of 36, 25, or 16 images, resulting in near- or sub-baseline performance across tasks. LOCoVQA presents a simple benchmark-generating process applicable to any VQA or retrieval evaluation dataset, making it easy to assess extractive reasoning in VLMs. Our findings suggest that

training tasks requiring attention across multiple context images to extract information—rather than simple single-image tasks—should be included in VLM training. By measuring this capacity it offers an appealing direction for future work.

4.9 Limitations

Although LoCoVQA is a generalized process for producing any VLM benchmark, we only evaluated it on three tasks. While the strong exponential decay trends between visual context length and performance observed across all three are compelling, a future direction is to expand LoCoVQA to additional tasks.

While LoCoVQA samples distractor images from the same datasets for open-domain and multiple-choice VQA, and our process appears to accurately filter out collisions, it is likely that a small number of collisions still occur, as it is inherently difficult to ensure no failures in an automated generating process [128]. This may lead to a natural ceiling on VLM performance at each visual context length.

Other important long-context capabilities likely exist that are not captured by LoCoVQA or prior work such as MILEBench [112]. Augmenting these evaluations with tests that capture additional orthogonal VLM long-context capabilities is an important direction for future work.

4.10 Acknowledgements

The contents of this chapter is a result of a collaboration. This work was supported in part by the National Science Foundation Graduate Research Fellowship under Grant No. 1650114, and CAREER Award under Grant No. 2048122. This project was mentored by Michael Saxon and advised by William Wang.

Appendix A

Appendix

A.1 Appendix: Benchmark for LLM Reasoning

A.1.1 Data Collection

Our corpus consists of the entirety of the English Wikipedia, snapshotted on 23 May 2022. Wikipedia presents a list of curated “Good Article”¹, which are articles that are nominated and reviewed to fit the “Good Article Criteria”². Articles from this category are guaranteed to have correct spelling and grammar, as well as clear and concise diction. Our final keyword list includes: “because”, “due to”, “therefore”, “consequently”, “resulted in”, “resulting in”, and “as a result”.

A.1.2 Data Collection Validation

Each stage in our data collection process is followed by two additional validation layers. For Stage 1, workers are prohibited from submitting more than 20 entries until their annotations have been manually validated. The annotation result passes through

¹https://en.wikipedia.org/wiki/Wikipedia:Good_articles/all

²https://en.wikipedia.org/wiki/Wikipedia:Good_article_criteria

Model	Fine-tuned GPT-2 vs. Few-shot GPT-3						
	S-BLEU	WMD	SMS	BERT-f1	ROU.-1	ROU.-2	ROU.-L
GPT-2							
<i>Greedy</i>	0.042	0.541	15.81	0.773	0.212	0.057	0.184
<i>Temp 0.5</i>	0.037	0.540	15.30	0.770	0.198	0.047	0.169
<i>Temp 1.0</i>	0.022	0.536	13.25	0.760	0.161	0.022	0.134
GPT-3							
<i>Temp 1.0</i>	0.055	0.555	14.93	0.792	0.240	0.057	0.199

Table A.1: Explanation Evaluation Results of WIKIWHY dataset according to the following metrics: SacreBLEU (S-BLEU) [1], Word-Mover’s distance (WMD) [2, 3], Sentence Mover’s Similarity Metrics (SMS) [4], BERT-f1 Score [5], ROUGE-1 (abbreviated as ROU.-1), ROUGE-2 (abbreviated as ROU.-2), and ROUGE-L (abbreviated as ROU.-L for brevity) (all ROUGE-f1 Scores [6] averaged). SMS is scaled by 1000 for readability.

another phase of manual validation to ensure that the quality is kept up after workers’ initial submissions are accepted by quality control. For Stage 2, we track a separate list of qualified workers for explanation quality.

Similar to Stage 1, Stage 2’s initial submit limit (the “speed bump”) is 10. Undergraduate students manually reviewed the examples from stage-2-qualified workers. These panelists were instructed and shown demonstrations of marking explanations as satisfying or not and correcting minor errors for slight quality improvements. While manually approved workers write each WIKIWHY explanation, these hand-reviewed samples ultimately comprise the test and development sets. The continuous flow between stages is enabled by a backend system we implemented to maintain a database of submissions. This system serves inputs to both MTurk interfaces, as well as the front-end validation interfaces provided to the undergraduate panelists.

A.1.3 Additional Results

We include additional evaluations of our generated explanations using simple metrics. Table A.1 shows performance on the **EO** task, and Table A.2 show performance on the

Model	GPT-3 Prompt Input Experiments						
	S-BLEU	WMD	SMS	BERT-f1	ROU.-1	ROU.-2	ROU.-L
Input Setting							
<i>Ideal</i>	0.055	0.555	14.93	0.792	0.240	0.057	0.199
<i>Well-Selected</i>	0.030	0.546	13.27	0.776	0.203	0.049	0.149
<i>End-to-end</i>	0.023	0.542	13.22	0.768	0.200	0.038	0.144

Table A.2: **GPT-3** explanation results with various input settings: *Ideal*- gold cause/answer, *Well-Selected*- provided cause/answer predicted by best-performing reader model (**FiD**), *End-to-end*- provided only question/effect (**GPT-3** completes end-to-end task)

MODEL	WikiWhy	
	Top-1 Acc	MRR
BM25	0.810	0.858
DPR	0.340	0.448

Table A.3: Document Retrieval for WIKIWHY. **BM25** consistently outperforms **DPR**.

A&E task. We also include results from the QA task in Table A.3 and Table A.4. Automatic evaluation on individual topics categories are included in Table A.5.

GPT-3 Few-shot Exemplars

Cause (\mathcal{C}): There were time constraints to writing “Boruto: Naruto the Movie”

Effect (\mathcal{E}): Hiroyuki Yamashita felt pressured writing “Boruto: Naruto the Movie”

Explanation (\mathcal{X}): Creativity is difficult when put on a strict timetable. There was a need to both produce a good movie and do so on a strict time budget. These two demands put stress on Hiroyuki Yamashita while he worked.

Cause (\mathcal{C}): Homer P. Rainey had liberal views.

Effect (\mathcal{E}): Homer P. Rainey was fired by the University of Texas in 1944.

Explanation (\mathcal{X}): If the University of Texas is conservative, they wouldn’t want people working there who have liberal views.

Cause (\mathcal{C}): The large size and reddish tint of red maple buds

Effect (\mathcal{E}): Red maple buds which form in fall and winter are often visible from a distance.

Explanation (\mathcal{X}): The color red stands out from a distance, so if the buds are red in the fall and winter, you’d be able to see them from a distance.

Cause (\mathcal{C}): There were advances in technology, lower energy prices, a favorable exchange rate of the United States dollar, and lower alumina prices.

Effect (\mathcal{E}): Productions costs of aluminum changed in the late 20th century.

Explanation (\mathcal{X}): With advances in technology, prices of manufacturing change usually because they are now easier and cheaper to make. In this case it is aluminum that the price changed on because the technology improved the process.

Figure A.1: GPT-3 Few-shot Exemplars

Find a **Cause-Effect** Relation & Turn it into a **Why Question**.

Your question must be **specific enough** to make sense **ON ITS OWN** (without the passage)

Example 1: Use Full Names of People! ▼

Example 2: Specify When and Where so the situation is totally clear! ▼

Your Passage (click link to find details)

https://en.wikipedia.org/wiki/Chrissie_Watts (link opens in new tab)

The aftermath dominated EastEnders in 2005 and helped to revive the fortunes of the show. According to the former head of BBC Drama Serials, Mal Young, this was dependent on the character of Chrissie, who was responsible for "anchoring the success of the anniversary storyline". A similar sentiment was expressed by Ian Hyland in the Sunday Mirror, who although critical of the convoluted plot felt EastEnders was improving "mainly **because** Chrissie is doing her best to rescue the fallout from the storyline dirty bomb Den's murder has become", and described the character as performing a "rescue act" on the show. However, Jim Shelley of the Daily Mirror was highly critical of Chrissie, calling her "the ludicrous Lady MacBeth wannabe", and felt her departure was enabling EastEnders to move forward. In contrast, the TV editor of The Daily Telegraph hailed Chrissie as "helping revive the show's fortunes that had been lagging somewhat in recent years".

click me only if the current passage lacks a cause-effect relation

Step 1: Find Cause & Effect from the Passage

- To make the next step easier, write your effect as a full sentence with details so it's clear what the exact situation is
- Please do NOT repeat the cause in the effect box

Cause: write your specific cause here

Effect: write your specific effect here

Step 2: Turn the Cause-Effect into a Why Question & its Answer

- **The Question should ask about Effect, and the Answer should be Cause**
 - Example: Cause=drug overdose, Effect=Heath Ledger died.
 - Question: Why did Heath Ledger die?
- **Your Question must make sense on its own (without the passage)**
 - Add details (who, what, when, where) so it's clear what the exact situation is
 - Use full names for people, groups, and places
- **Someone seeing only your QUESTION should NOT need to ask any clarifying questions**
 - They SHOULD NOT need to ask "which ___ are you talking about?" since your question should already be explicit about which ___ it's talking about

Why Question: write why question about effect here

Answer (Cause): [write cause & effect first!]

Please Note: Failing to follow instructions will result in your WorkerId being blocked from future tasks published by our group. By submitting, you agree to the terms of this [consent form](#).

Submit

Figure A.2: Amazon Mechanical Turk Interface for Stage 1

1. Choose a Question to Answer (each involve **cause & effect**)

- Why did "The similarity between The Fault in Our Stars and Perks of Being a Wallflower." lead to "Stephen Chbosky turned down the opportunity to direct The Fault in Our Stars."?
- Why did "Stephen Chbosky turned down the opportunity to direct The Fault in Our Stars." result from "The similarity between The Fault in Our Stars and Perks of Being a Wallflower."?
- Why does "The similarity between The Fault in Our Stars and Perks of Being a Wallflower." cause "Stephen Chbosky turned down the opportunity to direct The Fault in Our Stars."?
- Why is "Stephen Chbosky turned down the opportunity to direct The Fault in Our Stars." a consequence of "The similarity between The Fault in Our Stars and Perks of Being a Wallflower."?

Additional Context for the Questions

On January 31, 2012, it was announced that Fox 2000, a division of 20th Century Fox, had optioned the rights to adapt John Green's novel *The Fault in Our Stars* for a feature film. Wyck Godfrey and Marty Bowen were due to produce the film with their production company, Temple Hill Entertainment. Stephen Chbosky, who directed *The Perks of Being a Wallflower* (also filmed in Pittsburgh), was in talks to direct the film but turned it down because of its similarity to *Perks*. On February 19, 2013, Josh Boone was hired as director; Scott Neustadter and Michael H. Weber were hired to adapt the novel into a screenplay their second adaptation for Fox, following *Rosaline*.

2. Answer the Question

Example 1: Most explanations are fairly short (1 or 2 entries) ∨

Example 2: Most explanations only require basic logic ∨

Example 3: You may need to search online if you are unsure of the answer ∨

Requirements

- Add a new entry for each step/sentence
- Use complete sentences with good spelling and grammar
- **Carefully read the question you chose and actually respond to it**
- **DO NOT** only write or rephrase "cause leads to effect"
 - We already know this! We want you to explain WHY that is the case

Your Question

Why did "The similarity between *The Fault in Our Stars* and *Perks of Being a Wallflower*." lead to "Stephen Chbosky turned down the opportunity to direct *The Fault in Our Stars*."?

If he had directed both, it could have endangered his nomination for the Academy Award for "Perks of being a Wallflower"

Filming two similar movies would make him look like a one-trick pony. x

Add Explanation Step

Please Note: Submitting work with egregious grammar errors, inappropriately copied text, or nonsense answers (or otherwise failing to follow instructions) will result in your WorkerId being blocked from future tasks published by our group.

By submitting, you agree to the terms of this [consent form](#).

Submit

Figure A.3: Amazon Mechanical Turk Interface for Stage 2

MODEL	WikiWhy		
	S-BLEU	BERT-f1	WMD
RoBERTa			
<i>Gold</i>	0.246	0.860	0.637
<i>BM25</i>	0.214	0.832	0.620
BigBird			
<i>Gold</i>	0.258	0.825	0.615
<i>BM25</i>	0.223	0.802	0.602
FiD			
<i>Gold</i>	0.373	0.863	0.658
<i>BM25</i>	0.259	0.827	0.617

Table A.4: Answer Evaluation Results for WIKIWHY dataset. Stage 1: **RoBERTa**, **BigBird**, and **FiD**. **FiD Gold** is fine-tuned on 80% train split & evaluated on 10% dev split.

	Most Frequent Genres						
	ARTS	GEOG	HISTORY	MEDIA	MUSIC	SCIENCE	TECH
Models							
GPT-2	0.256	0.221	0.202	0.161	0.239	0.252	0.236
GPT-3	0.412	0.372	0.341	0.335	0.301	0.371	0.333

Table A.5: Explanation performance (unordered f1) over the most frequent topics. We use GPT-2 under the greedy setting and GPT-3 under the same defaults as Table 2.5

A.1.4 Crowd Worker Interface

Figure A.2 and Figure A.3 display the interfaces for the first and second stages respectively. In addition to the list of requirements, we provide examples and tips to further clarify our expectations. The passage is displayed with a link to the full article so workers can view the complete context if needed. Every passage contains a highlighted causal connective, allowing workers to quickly scan and skip irrelevant portions. Each passage is retrieved from our custom database through our API. If the passage is too difficult for the worker to understand or lacks a cause-effect relation, the worker can click the button below for another random passage.

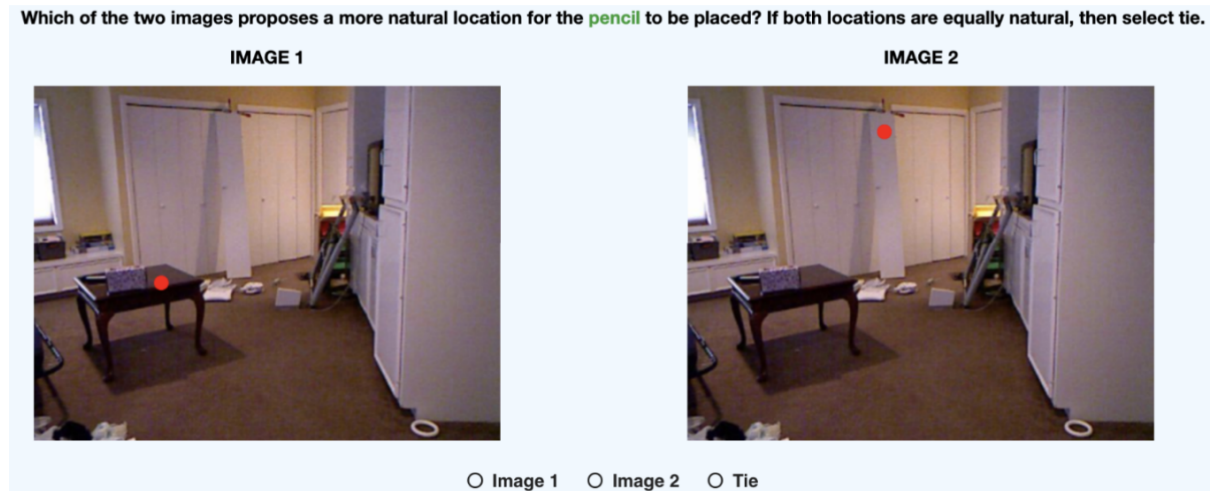


Figure A.4: Amazon Mechanical Turk (MTurk) Interface for human evaluation

A.2 Appendix: Scene Understanding with LLMs

A.2.1 Website with results, videos, and benchmark

The project website at <https://octo-pearl.github.io/> includes a demo video of the **OCTO+** method, leaderboard scores of various methods on the **PEARL** benchmark. The website also links to the official open-sourced code base <https://github.com/octo-pearl/octo-pearl>. We also created a HuggingFace Spaces demo <https://huggingface.co/spaces/adityas/OCTO> and HuggingFace PEARL dataset <https://huggingface.co/datasets/adityas/PEARL>.

A.2.2 Human Evaluation

We conducted human evaluation in two steps: MTurk study and expert evaluation. The Amazon Mechanical Turk (MTurk) study consisted of selecting which of the two images Image 1 and Image 2 proposes are more natural location for an object to be placed? A concrete example of the interface is shown in Figure A.4.

The human evaluation score is determined through the calculation of wins and ties

Method	Stage 1		Stage 2		Stage 3		Total Time (seconds)
	Models	Time (sec)	Models	Time (sec)	Models	Time (sec)	
OCTOPUS	SCP + ViLT	19.98	GPT-4	0.811	CLIPSeg	0.053	20.844
OCTO+	RAM++, G-DINO	0.539	GPT-4	0.811	G-SAM	2.428	3.778
OCTO+ w/CLIPSeg	RAM++, G-DINO	0.539	GPT-4	0.811	CLIPSeg	0.053	1.403
Best GPT-4V	—	—	GPT-4V	3.380	CLIPSeg	0.053	<u>3.433</u>

Table A.6: Comparison in Runtime Analysis of **OCTO+** with respect to other vision-language methods for content placement

(Equation A.1). This is based on the premise that if our method achieves a tie with the natural placement, it signifies the method’s ability to produce placements that are sufficiently natural.

$$\mathcal{S} = \frac{\text{wins} + \text{ties}}{\text{wins} + \text{ties} + \text{losses}} \quad (\text{A.1})$$

A.2.3 Runtime Analysis

We performed a runtime analysis to measure how other methods compare with **OCTO+**. To summarize the findings from Table A.6, we observe that **OCTO+** is 5.5 times faster than **OCTOPUS**. As expected, GPT-4 is 4.2 times faster than GPT-4V. We think this is due to the fact that the vision endpoint takes in the image embeddings as well as the text embeddings. These times were calculated by running on an NVIDIA T4 GPU on Google Colab session.

A.2.4 Additional Insights

Given the rapid advancement of large language models, we envisage the potential for the refinement of Stage 2 within the OCTO+ pipeline to accommodate an open-source LLM, as opposed to the proprietary GPT-4 model. Notably, models such as Mixtral [90], Vicuna [129], and Gemma [21] demonstrate commendable performance levels comparable to that of GPT-4, and may even exhibit superior capabilities in due course.

Furthermore, an avenue for enhancing precision in describing placement locations by LLMs could involve the utilization of prepositions (e.g., “above” or “left of”) instead of the conventional approach of merely situation objects “on” designated tags. Similarly, there exists potential for future methodologies to incorporate considerations of the scene’s 3D geometry, as opposed to solely selecting a central location.

A.3 Appendix: Long-Context VLM Reasoning

A.3.1 Supplementary Information on Evaluated Models

The open-weight models that we evaluated are:

- (i) Moondream2-1.6b [123] based on Phi-1.5 [130],
- (ii) LLaVA-1.5 [79] with Vicuna-7b [129] as the LLM backbone,
- (iii) LLaVA-1.6 (LLaVA-Next) [124], an improvement over LLaVA-1.5 with higher image resolution and better visual reasoning, uses Mistral-7b [131],
- (iv) PaliGemma-3b [125], based on open components from the SigLip [132] image encoder and the Gemma [133] language model,
- (v) Mantis-bakllava-7b [91] fine-tuned from BakLLaVA [134] and derived from LLaVA but using Mistral-7b [131],

- (vi) Mantis-Idefics2-8b [91], the current state-of-the-art Mantis variant, based on Idefics2-8b [135].

The closed-source models that we evaluated are:

- (i) OpenAI’s GPT-4 Vision [18] API (December 2023)
- (ii) Gemini 1.0 Pro Vision [126] API (December 2023)
- (iii) Gemini 1.5 Flash [19] API with a 1.0M+ context window (May 2024)

A.3.2 Release Information

Full source code for generating distract datasets is available at locovqa.github.io. is released under the Apache v2.0 license.

A.3.3 Filtering Collisions in LoCoVQA

Figure A.5 shows an example of methodology used to filter for collisions in 4-Image Context Length input. The LLM Query (Q) was used to prompt GPT-4 LLM. The second figure shows an example of a collision occurring when creating a 4-image context. In this case, the two images contain oranges, so we resample one of the images to avoid collision. The OK-VQA questions in this case are “What type of plant do these fruits grow from?” for the content image and “In which US states are these fruits commonly grown?” for the distractor image. Both require visual reasoning step that the fruit described in this image is an orange.

A.3.4 Model-level Scoring Details

In this section we discuss the model-level scoring details for OK-VQA, MMStar, and MNIST experiments.



Figure A.5: Collision Filtering Method for image construction by prompting LLM with query Q to identify entities. Cell with represents the entities for each image X_i . If there are no entities in common, there are no collisions so we mark cell with indicating this is a valid construction of images for S .

For the OK-VQA free-form ground truth answers, if any of the ground truth candidate answers is a substring of the model-generated answer, we award full points for the exact-matching setting. The other two metrics we used were BERTScore and ROUGE scores. BERTScore [5] is a robust text comparison metric which matches candidate and reference based on the cosine similarities of the embeddings. Using the Sentence Transformer package `all-MiniLM-L6-v2` model, we calculated the maximum BERTScore between all the ground truth answers against the model answer. For the ROUGE score [6], we compared the ground truth answer and model answer using the default settings with ROUGE-L, which measure the longest common subsequence at the sentence-level

For multiple-choice answer form in MMStar, when prompting the VLM, we ask it to produce the answer in the following format: “Please provide the answer in format `<answer> </answer>` where the answer is between the tags. For example if the result is apple, your answer should be `<answer>apple</answer>`.” This format ensure some grammatical structure and requires the answer to be enclosed within the `<answer>` tags. GPT4V, Gemini 1.0, and Gemini 1.5 under both composed and interleaved settings produced answers following this format. However, we did not observe consistent behavior from LLaVA-based and Mantis-based variants. Moondream and Gemma consistently responded with a single-choice answer, making the evaluation of these two models the easiest. Due to the variance in VLM responses, we adopted a robust evaluation procedure. First, we checked if the answer was between the `<answer>` tags and, if so, extracted the MCQ choice directly from the tag. If not, we noted that outputs often followed the format, “Answer: *choice*,” where the choice follows directly after. We also checked edge cases, including instances where the first letter in the string is a multiple-choice followed by a colon, choices provided in parentheses, only the answer text without the corresponding letter, and a single letter provided.

For MNIST evaluation, we ensure that the model response contains a list of digits

separated by commas. If the output is separated by spaces, we parse it into an array to provide a completely fair evaluation. The response must contain exactly the same number of digits as those in the MNIST digits. We sort both the candidate and reference lists and compare them to check for equality.

A.3.5 Supplementary Results

Table A.7 for open-weight models and Table A.8 for closed-source models display the r^2 values for how well the curves from Figure 4.4 fit. We also provide the p -val to denote the statistical significance of the overall fit. The red highlight signifies subchance performance (highlighted if more than half of the data points were below random choice). A down arrow indicates a negative correlation (as visual context length increase, performance increases). This phenomenon is observed in very few samples, likely due to the visual information not being necessary to answer the question or a smaller sample size causing noise in the data, thereby resulting in similar performance or even performance gains in higher contexts.

Methods	LLaVA (c)	LLaVA-1.6 (c)	Moondream (c)	PaliGemma(c)	Mantis (i)
MMStar	.924***	.957***	.903***	.947***	.967***
SEEDBench	.119	.825***	.825***	.988***	.885***
MMBench	.959***	.992***	.937***	.946***	.972***
MMMU	.367↓**	.691↓***	.586**	.923↓	.282***
MathVista	.513*	.907↓***	.004	.132↓	.848***
ScienceQA	.080↓	.020	.279*	.907***	.209↓*
AI2D	.658↓**	.604**	.375	.882***	.857***
OK-VQA	.985***	.989***	.979***	.968***	.991***
BERT	.986***	.973***	.981***	.968***	.986***
ROUGE	.988***	.973***	.985***	.968***	.918***
Haystack-9	.067	.850	.000	.895*	.000
Haystack-16	.000	.790	.000	.535	.000
Haystack-25	.000	.804*	.000	.651	.000
Haystack-36	.000	.596	.000	.668	.000

Table A.7: Logarithmic curve fit r^2 values are reported for each open-weight model, followed by a symbol denoting the p -value for statistical significance. The symbols represent: * for $p \leq 0.05$, ** for $p \leq 0.01$, & *** for $p \leq 0.001$. Light pink cells represent correlations for sets of values that fall below chance performance.

Methods	GPT-4V (c)	GPT-4V (i)	Gemini 1.0 (c)	Gemini 1.0 (i)	Gemini 1.5 (c)	Gemini 1.5 (i)
MMStar	.880***	.937***	.972***	.972***	.985***	.981***
SEEDBench	.737***	.956***	.765***	.765***	.977***	.980***
MMBench	.951***	.959***	.977***	.977***	.964***	.948***
MMMU	.861***	.372***	.493**	.493**	.487**	.533**
MathVista	.676***	.858***	.832***	.832***	.840***	.690***
ScienceQA	.932***	.820***	.268*	.268*	.546**	.309*
AI2D	.743***	.635**	.880***	.880***	.969***	.882***
OK-VQA	.963***	.975***	.962***	.962***	.991***	.989***
BERT	.969***	.977***	.955***	.955***	.989***	.993***
ROUGE	.958***	.979***	.955***	.955***	.981***	.989***
Haystack-9	.505	.632	-	-	.052	.895*
Haystack-16	.366	.866*	-	-	.025	.256
Haystack-25	.863*	.704	-	-	.118	.146
Haystack-36	.841*	.645	-	-	.489	.896

Table A.8: Logarithmic curve fit r^2 values are reported for each closed-source model, followed by a symbol denoting the p -value for statistical significance. The symbols represent: * for $p \leq 0.05$, ** for $p \leq 0.01$, & *** for $p \leq 0.001$. Light pink cells represent correlations for sets of values that fall below chance performance.

Bibliography

- [1] M. Post, *A call for clarity in reporting BLEU scores*, in *Proceedings of the Third Conference on Machine Translation: Research Papers* (O. Bojar, R. Chatterjee, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, C. Monz, M. Negri, A. N ev ol, M. Neves, M. Post, L. Specia, M. Turchi, and K. Verspoor, eds.), (Brussels, Belgium), pp. 186–191, Association for Computational Linguistics, Oct., 2018.
- [2] R. Sato, M. Yamada, and H. Kashima, *Re-evaluating word mover’s distance*, in *International Conference on Machine Learning*, pp. 19231–19249, PMLR, 2022.
- [3] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, *From word embeddings to document distances*, in *International conference on machine learning*, pp. 957–966, PMLR, 2015.
- [4] E. Clark, A. Celikyilmaz, and N. A. Smith, *Sentence mover’s similarity: Automatic evaluation for multi-sentence texts*, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (A. Korhonen, D. Traum, and L. M arquez, eds.), (Florence, Italy), pp. 2748–2760, Association for Computational Linguistics, July, 2019.
- [5] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, *Bertscore: Evaluating text generation with bert*, *arXiv preprint arXiv:1904.09675* (2019).
- [6] C.-Y. Lin, *ROUGE: A package for automatic evaluation of summaries*, in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July, 2004.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, *Advances in neural information processing systems* **30** (2017).
- [8] M. Schuster and K. K. Paliwal, *Bidirectional recurrent neural networks*, *IEEE transactions on Signal Processing* **45** (1997), no. 11 2673–2681.
- [9] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, *Neural computation* **9** (1997), no. 8 1735–1780.

- [10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, *arXiv preprint arXiv:1412.3555* (2014).
- [11] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of deep bidirectional transformers for language understanding*, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June, 2019.
- [13] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et. al.*, *Improving language understanding by generative pre-training*, .
- [14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An image is worth 16x16 words: Transformers for image recognition at scale*, *CoRR* **abs/2010.11929** (2020) [arXiv:2010.1192].
- [15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, *Backpropagation applied to handwritten zip code recognition*, *Neural computation* **1** (1989), no. 4 541–551.
- [16] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et. al.*, *Learning transferable visual models from natural language supervision*, in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.
- [17] OpenAI, “Gpt-4o.” <https://openai.com/index/hello-gpt-4o>, 2024.
- [18] OpenAI, “Gpt-4v(ision) system card.” <https://openai.com/research/gpt-4v-system-card>, 2023.
- [19] G. Team, M. Reid, N. Savinov, D. Teplyashin, Dmitry, Lepikhin, T. Lillicrap, J. baptiste Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, I. Antonoglou, R. Anil, S. Borgeaud, A. Dai, K. Millican, E. Dyer, M. Glaese, T. Sottiaux, B. Lee, F. Viola, M. Reynolds, Y. Xu, J. Molloy, J. Chen, M. Isard, P. Barham, T. Hennigan, R. Mellroy, M. Johnson, J. Schalkwyk, E. Collins, E. Rutherford, E. Moreira, K. Ayoub, M. Goel, C. Meyer, G. Thornton, Z. Yang, H. Michalewski, Z. Abbas, N. Schucher, A. Anand, R. Ives, J. Keeling, K. Lenc, S. Haykal, S. Shakeri, P. Shyam, A. Chowdhery, R. Ring, S. Spencer, E. Sezener,

L. Vilnis, O. Chang, N. Morioka, G. Tucker, C. Zheng, O. Woodman, N. Attaluri,
 T. Kocisky, E. Eltyshev, X. Chen, T. Chung, V. Selo, S. Brahma, P. Georgiev,
 A. Slone, Z. Zhu, J. Lottes, S. Qiao, B. Caine, S. Riedel, A. Tomala,
 M. Chadwick, J. Love, P. Choy, S. Mittal, N. Houlsby, Y. Tang, M. Lamm,
 L. Bai, Q. Zhang, L. He, Y. Cheng, P. Humphreys, Y. Li, S. Brin, A. Cassirer,
 Y. Miao, L. Zilka, T. Tobin, K. Xu, L. Proleev, D. Sohn, A. Magni, L. A.
 Hendricks, I. Gao, S. Ontanon, O. Bunyan, N. Byrd, A. Sharma, B. Zhang,
 M. Pinto, R. Sinha, H. Mehta, D. Jia, S. Caelles, A. Webson, A. Morris,
 B. Roelofs, Y. Ding, R. Strudel, X. Xiong, M. Ritter, M. Dehghani,
 R. Chaabouni, A. Karmarkar, G. Lai, F. Mentzer, B. Xu, Y. Li, Y. Zhang, T. L.
 Paine, A. Goldin, B. Neyshabur, K. Baumli, A. Levskaya, M. Laskin, W. Jia,
 J. W. Rae, K. Xiao, A. He, S. Giordano, L. Yagati, J.-B. Lespiau, P. Natsev,
 S. Ganapathy, F. Liu, D. Martins, N. Chen, Y. Xu, M. Barnes, R. May, A. Vezer,
 J. Oh, K. Franko, S. Bridgers, R. Zhao, B. Wu, B. Mustafa, S. Sechrist,
 E. Parisotto, T. S. Pillai, C. Larkin, C. Gu, C. Sorokin, M. Krikun, A. Guseynov,
 J. Landon, R. Datta, A. Pritzel, P. Thacker, F. Yang, K. Hui, A. Hauth, C.-K.
 Yeh, D. Barker, J. Mao-Jones, S. Austin, H. Sheahan, P. Schuh, J. Svensson,
 R. Jain, V. Ramasesh, A. Briukhov, D.-W. Chung, T. von Glehn, C. Butterfield,
 P. Jhakra, M. Wiethoff, J. Frye, J. Grimstad, B. Changpinyo, C. L. Lan,
 A. Bortsova, Y. Wu, P. Voigtlaender, T. Sainath, S. Gu, C. Smith, W. Hawkins,
 K. Cao, J. Besley, S. Srinivasan, M. Omernick, C. Gaffney, G. Surita, R. Burnell,
 B. Damoc, J. Ahn, A. Brock, M. Pajarskas, A. Petrushkina, S. Noury, L. Blanco,
 K. Swersky, A. Ahuja, T. Avrahami, V. Misra, R. de Liedekerke, M. Iinuma,
 A. Polozov, S. York, G. van den Driessche, P. Michel, J. Chiu, R. Blevins,
 Z. Gleicher, A. Recasens, A. Rrustemi, E. Gribovskaya, A. Roy, W. Gworek,
 S. M. R. Arnold, L. Lee, J. Lee-Thorp, M. Maggioni, E. Piqueras, K. Badola,
 S. Vikram, L. Gonzalez, A. Baddepudi, E. Senter, J. Devlin, J. Qin, M. Azzam,
 M. Trebacz, M. Polacek, K. Krishnakumar, S. yiin Chang, M. Tung, I. Penchev,
 R. Joshi, K. Olszewska, C. Muir, M. Wirth, A. J. Hartman, J. Newlan,
 S. Kashem, V. Bolina, E. Dabir, J. van Amersfoort, Z. Ahmed, J. Cobon-Kerr,
 A. Kamath, A. M. Hrafinkelsson, L. Hou, I. Mackinnon, A. Frechette, E. Noland,
 X. Si, E. Taropa, D. Li, P. Crone, A. Gulati, S. Cevey, J. Adler, A. Ma, D. Silver,
 S. Tokumine, R. Powell, S. Lee, K. Vodrahalli, S. Hassan, D. Mincu, A. Yang,
 N. Levine, J. Brennan, M. Wang, S. Hodgkinson, J. Zhao, J. Lipschultz, A. Pope,
 M. B. Chang, C. Li, L. E. Shafey, M. Paganini, S. Douglas, B. Bohnet, F. Pardo,
 S. Odoom, M. Rosca, C. N. dos Santos, K. Soparkar, A. Guez, T. Hudson,
 S. Hansen, C. Asawaroengchai, R. Addanki, T. Yu, W. Stokowiec, M. Khan,
 J. Gilmer, J. Lee, C. G. Bostock, K. Rong, J. Caton, P. Pejman, F. Pavetic,
 G. Brown, V. Sharma, M. Lučić, R. Samuel, J. Djolonga, A. Mandhane, L. L.
 Sjöstrand, E. Buchatskaya, E. White, N. Clay, J. Jiang, H. Lim, R. Hemsley,
 Z. Cankara, J. Labanowski, N. D. Cao, D. Steiner, S. H. Hashemi, J. Austin,
 A. Gergely, T. Blyth, J. Stanton, K. Shivakumar, A. Siddhant, A. Andreassen,

C. Araya, N. Sethi, R. Shivanna, S. Hand, A. Bapna, A. Khodaei, A. Miech,
 G. Tanzer, A. Swing, S. Thakoor, L. Aroyo, Z. Pan, Z. Nado, J. Sygnowski,
 S. Winkler, D. Yu, M. Saleh, L. Maggiore, Y. Bansal, X. Garcia, M. Kazemi,
 P. Patil, I. Dasgupta, I. Barr, M. Giang, T. Kagohara, I. Danihelka, A. Marathe,
 V. Feinberg, M. Elhawaty, N. Ghelani, D. Horgan, H. Miller, L. Walker,
 R. Tanburn, M. Tariq, D. Shrivastava, F. Xia, Q. Wang, C.-C. Chiu, Z. Ashwood,
 K. Baatarsukh, S. Samangoeei, R. L. Kaufman, F. Alcober, A. Stjerngren,
 P. Komarek, K. Tsihlas, A. Boral, R. Comanescu, J. Chen, R. Liu, C. Welty,
 D. Bloxwich, C. Chen, Y. Sun, F. Feng, M. Mauger, X. Dotiwalla,
 V. Hellendoorn, M. Sharman, I. Zheng, K. Haridasan, G. Barth-Maron,
 C. Swanson, D. Rogozińska, A. Andreev, P. K. Rubenstein, R. Sang, D. Hurt,
 G. Elsayed, R. Wang, D. Lacey, A. Ilić, Y. Zhao, A. Iwanicki, A. Lince, A. Chen,
 C. Lyu, C. Lebsack, J. Griffith, M. Gaba, P. Sandhu, P. Chen, A. Koop,
 R. Rajwar, S. H. Yeganeh, S. Chang, R. Zhu, S. Radpour, E. Davoodi, V. I. Lei,
 Y. Xu, D. Toyama, C. Segal, M. Wicke, H. Lin, A. Bulanova, A. P. Badia,
 N. Rakićević, P. Sprechmann, A. Filos, S. Hou, V. Campos, N. Kassner,
 D. Sachan, M. Fortunato, C. Iwuanyanwu, V. Nikolaev, B. Lakshminarayanan,
 S. Jazayeri, M. Varadarajan, C. Tekur, D. Fritz, M. Khalman, D. Reitter,
 K. Dasgupta, S. Sarcar, T. Ornduff, J. Snaider, F. Huot, J. Jia, R. Kemp,
 N. Trdin, A. Vijayakumar, L. Kim, C. Angermueller, L. Lao, T. Liu, H. Zhang,
 D. Engel, S. Greene, A. White, J. Austin, L. Taylor, S. Ashraf, D. Liu,
 M. Georgaki, I. Cai, Y. Kulizhskaya, S. Goenka, B. Saeta, Y. Xu, C. Frank,
 D. de Cesare, B. Robenek, H. Richardson, M. Alnahlawi, C. Yew, P. Ponnappalli,
 M. Tagliasacchi, A. Korchemniy, Y. Kim, D. Li, B. Rosgen, K. Levin, J. Wiesner,
 P. Banzal, P. Srinivasan, H. Yu, Çağlar Ünlü, D. Reid, Z. Tung, D. Finchelstein,
 R. Kumar, A. Elisseeff, J. Huang, M. Zhang, R. Aguilar, M. Giménez, J. Xia,
 O. Dousse, W. Gierke, D. Yates, K. Jalan, L. Li, E. Latorre-Chimoto, D. D.
 Nguyen, K. Durden, P. Kallakuri, Y. Liu, M. Johnson, T. Tsai, A. Talbert, J. Liu,
 A. Neitz, C. Elkind, M. Selvi, M. Jasarevic, L. B. Soares, A. Cui, P. Wang, A. W.
 Wang, X. Ye, K. Kallarackal, L. Loher, H. Lam, J. Broder, D. Holtmann-Rice,
 N. Martin, B. Ramadhana, M. Shukla, S. Basu, A. Mohan, N. Fernando,
 N. Fiedel, K. Paterson, H. Li, A. Garg, J. Park, D. Choi, D. Wu, S. Singh,
 Z. Zhang, A. Globerson, L. Yu, J. Carpenter, F. de Chaumont Quitry,
 C. Radebaugh, C.-C. Lin, A. Tudor, P. Shroff, D. Garmon, D. Du, N. Vats,
 H. Lu, S. Iqbal, A. Yakubovich, N. Tripuraneni, J. Manyika, H. Qureshi, N. Hua,
 C. Ngani, M. A. Raad, H. Forbes, J. Stanway, M. Sundararajan, V. Ungureanu,
 C. Bishop, Y. Li, B. Venkatraman, B. Li, C. Thornton, S. Scellato, N. Gupta,
 Y. Wang, I. Tenney, X. Wu, A. Shenoy, G. Carvajal, D. G. Wright, B. Bariach,
 Z. Xiao, P. Hawkins, S. Dalmia, C. Farabet, P. Valenzuela, Q. Yuan, A. Agarwal,
 M. Chen, W. Kim, B. Hulse, N. Dukkupati, A. Paszke, A. Bolt, K. Choo,
 J. Beattie, J. Prendki, H. Vashisht, R. Santamaria-Fernandez, L. C. Cobo,
 J. Wilkiewicz, D. Madras, A. Elqursh, G. Uy, K. Ramirez, M. Harvey, T. Liechty,

- H. Zen, J. Seibert, C. H. Hu, A. Khorlin, M. Le, A. Aharoni, M. Li, L. Wang, S. Kumar, N. Casagrande, J. Hoover, D. E. Badawy, D. Soergel, D. Vnukov, M. Miecnikowski, J. Simsa, P. Kumar, T. Sellam, D. Vlastic, S. Daruki, N. Shabat, J. Zhang, G. Su, J. Zhang, J. Liu, Y. Sun, E. Palmer, A. Ghaffarkhah, X. Xiong, V. Cotruta, M. Fink, L. Dixon, A. Sreevatsa, A. Goedeckemeyer, A. Dimitriev, M. Jafari, R. Crocker, N. FitzGerald, A. Kumar, S. Ghemawat, I. Philips, F. Liu, Y. Liang, R. Sterneck, A. Repina, M. Wu, L. Knight, M. Georgiev, H. Lee, H. Askham, A. Chakladar, A. Louis, C. Crous, H. Cate, D. Petrova, M. Quinn, D. Owusu-Afriyie, A. Singhal, N. Wei, S. Kim, D. Vincent, M. Nasr, C. A. Choquette-Choo, R. Tojo, S. Lu, D. de Las Casas, Y. Cheng, T. Bolukbasi, K. Lee, S. Fatehi, R. Ananthanarayanan, M. Patel, C. Kaed, J. Li, S. R. Belle, Z. Chen, J. Konzelmann, S. Pöder, R. Garg, V. Koverkathu, A. Brown, C. Dyer, R. Liu, A. Nova, J. Xu, A. Walton, A. Parrish, M. Epstein, S. McCarthy, S. Petrov, D. Hassabis, K. Kavukcuoglu, J. Dean, and O. Vinyals, *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*, 2024.
- [20] Anthropic, “Claude 3 opus.”
<https://www.anthropic.com/news/claude-3-family>, 2024.
- [21] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, A. Chowdhery, A. Roberts, A. Barua, A. Botev, A. Castro-Ros, A. Slone, A. Héliou, A. Tacchetti, A. Bulanova, A. Paterson, B. Tsai, B. Shahriari, C. L. Lan, C. A. Choquette-Choo, C. Crepy, D. Cer, D. Ippolito, D. Reid, E. Buchatskaya, E. Ni, E. Noland, G. Yan, G. Tucker, G.-C. Muraru, G. Rozhdestvenskiy, H. Michalewski, I. Tenney, I. Grishchenko, J. Austin, J. Keeling, J. Labanowski, J.-B. Lespiau, J. Stanway, J. Brennan, J. Chen, J. Ferret, J. Chiu, J. Mao-Jones, K. Lee, K. Yu, K. Millican, L. L. Sjoesund, L. Lee, L. Dixon, M. Reid, M. Miłkowska, M. Wirth, M. Sharman, N. Chinaev, N. Thain, O. Bachem, O. Chang, O. Wahltinez, P. Bailey, P. Michel, P. Yotov, P. G. Sessa, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y. hui Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral, F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, and K. Kenealy, *Gemma: Open models based on gemini research and technology*, 2024.
- [22] H. Liu, C. Li, Q. Wu, and Y. J. Lee, *Visual instruction tuning*, 2023.
- [23] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, *Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond*, 2023.

- [24] D. Jiang, X. He, H. Zeng, C. Wei, M. W. Ku, Q. Liu, and W. Chen, *Mantis: Interleaved multi-image instruction tuning*, arXiv2405.01483.
- [25] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, *et. al.*, *Emergent abilities of large language models*, *arXiv preprint arXiv:2206.07682* (2022).
- [26] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et. al.*, *Chain-of-thought prompting elicits reasoning in large language models*, *Advances in neural information processing systems* **35** (2022) 24824–24837.
- [27] K. Shuster, J. Urbanek, A. Szlam, and J. Weston, *Am I me or you? state-of-the-art dialogue models cannot maintain an identity*, in *Findings of the Association for Computational Linguistics: NAACL 2022* (M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, eds.), (Seattle, United States), pp. 2367–2387, Association for Computational Linguistics, July, 2022.
- [28] S. Wadhwa, V. Embar, M. Grabmair, and E. Nyberg, *Towards inference-oriented reading comprehension: ParallelQA*, in *Proceedings of the Workshop on Generalization in the Age of Deep Learning* (Y. Bisk, O. Levy, and M. Yatskar, eds.), (New Orleans, Louisiana), pp. 1–7, Association for Computational Linguistics, June, 2018.
- [29] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, *HotpotQA: A dataset for diverse, explainable multi-hop question answering*, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, eds.), (Brussels, Belgium), pp. 2369–2380, Association for Computational Linguistics, Oct.-Nov., 2018.
- [30] A. Chadha and V. Jain, *ireason: Multimodal commonsense reasoning using videos and natural language with interpretability*, *arXiv preprint arXiv:2107.10300* (2021).
- [31] J. Pearl, *Causality*. Cambridge university press, 2009.
- [32] Y. Sui, S. Feng, H. Zhang, J. Cao, L. Hu, and N. Zhu, *Causality-aware enhanced model for multi-hop question answering over knowledge graphs*, *Knowledge-Based Systems* **250** (2022) 108943.
- [33] W. Chen, H. Zha, Z. Chen, W. Xiong, H. Wang, and W. Y. Wang, *HybridQA: A dataset of multi-hop question answering over tabular and textual data*, in *Findings of the Association for Computational Linguistics: EMNLP 2020* (T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 1026–1036, Association for Computational Linguistics, Nov., 2020.

- [34] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, *Vqa: Visual question answering*, in *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.
- [35] J. Hessel, J. D. Hwang, J. S. Park, R. Zellers, C. Bhagavatula, A. Rohrbach, K. Saenko, and Y. Choi, *The abduction of sherlock holmes: A dataset for visual abductive reasoning*, in *European Conference on Computer Vision*, pp. 558–575, Springer, 2022.
- [36] Y. K. Lal, N. Chambers, R. Mooney, and N. Balasubramanian, *TellMeWhy: A dataset for answering why-questions in narratives*, in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (C. Zong, F. Xia, W. Li, and R. Navigli, eds.), (Online), pp. 596–610, Association for Computational Linguistics, Aug., 2021.
- [37] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, *ELI5: Long form question answering*, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (A. Korhonen, D. Traum, and L. Màrquez, eds.), (Florence, Italy), pp. 3558–3567, Association for Computational Linguistics, July, 2019.
- [38] N. F. Rajani, B. McCann, C. Xiong, and R. Socher, *Explain yourself! leveraging language models for commonsense reasoning*, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (A. Korhonen, D. Traum, and L. Màrquez, eds.), (Florence, Italy), pp. 4932–4942, Association for Computational Linguistics, July, 2019.
- [39] H. Jhamtani and P. Clark, *Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (B. Webber, T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 137–150, Association for Computational Linguistics, Nov., 2020.
- [40] L. Yang, Z. Wang, Y. Wu, J. Yang, and Y. Zhang, *Towards fine-grained causal reasoning and qa*, *arXiv preprint arXiv:2204.07408* (2022).
- [41] B. Dalvi, P. Jansen, O. Tafjord, Z. Xie, H. Smith, L. Pipatanangkura, and P. Clark, *Explaining answers with entailment trees*, in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, eds.), (Online and Punta Cana, Dominican Republic), pp. 7358–7370, Association for Computational Linguistics, Nov., 2021.
- [42] A. Talmor, J. Herzig, N. Lourie, and J. Berant, *CommonsenseQA: A question answering challenge targeting commonsense knowledge*, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational*

- Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 4149–4158, Association for Computational Linguistics, June, 2019.
- [43] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, *FEVER: a large-scale dataset for fact extraction and VERification*, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (M. Walker, H. Ji, and A. Stent, eds.), (New Orleans, Louisiana), pp. 809–819, Association for Computational Linguistics, June, 2018.
- [44] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom, *e-snli: Natural language inference with natural language explanations*, in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.
- [45] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan, *Learn to explain: Multimodal reasoning via thought chains for science question answering*, *Advances in Neural Information Processing Systems* **35** (2022) 2507–2521.
- [46] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, *Learning to retrieve reasoning paths over wikipedia graph for question answering*, *arXiv preprint arXiv:1911.10470* (2019).
- [47] J. K. Hubbard, M. A. Potts, and B. A. Couch, *How question types reveal student thinking: An experimental comparison of multiple-true-false and free-response formats*, *CBE—Life Sciences Education* **16** (2017), no. 2 ar26.
- [48] N. Inoue, P. Stenetorp, and K. Inui, *R4C: A benchmark for evaluating RC systems to get the right answer for the right reason*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds.), (Online), pp. 6740–6750, Association for Computational Linguistics, July, 2020.
- [49] G. Mohr, J. Engelkamp, and H. D. Zimmer, *Recall and recognition of self-performed acts*, *Psychological Research* **51** (1989) 181–187.
- [50] D. Neves Ribeiro, S. Wang, X. Ma, R. Dong, X. Wei, H. Zhu, X. Chen, P. Xu, Z. Huang, A. Arnold, and D. Roth, *Entailment tree explanations via iterative retrieval-generation reasoner*, in *Findings of the Association for Computational Linguistics: NAACL 2022* (M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, eds.), (Seattle, United States), pp. 465–475, Association for Computational Linguistics, July, 2022.

- [51] S. Robertson, H. Zaragoza, *et. al.*, *The probabilistic relevance framework: Bm25 and beyond*, *Foundations and Trends® in Information Retrieval* **3** (2009), no. 4 333–389.
- [52] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, *Dense passage retrieval for open-domain question answering*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (B. Webber, T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 6769–6781, Association for Computational Linguistics, Nov., 2020.
- [53] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, and R. Nogueira, *Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations*, in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2356–2362, 2021.
- [54] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, *Natural questions: A benchmark for question answering research*, *Transactions of the Association for Computational Linguistics* **7** (2019) 452–466.
- [55] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, *arXiv preprint arXiv:1907.11692* (2019).
- [56] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, *et. al.*, *Big bird: Transformers for longer sequences*, *Advances in neural information processing systems* **33** (2020) 17283–17297.
- [57] G. Izacard and E. Grave, *Leveraging passage retrieval with generative models for open domain question answering*, *arXiv preprint arXiv:2007.01282* (2020).
- [58] T. Wang, A. Roberts, D. Hesslow, T. Le Scao, H. W. Chung, I. Beltagy, J. Launay, and C. Raffel, *What language model architecture and pretraining objective works best for zero-shot generalization?*, in *International Conference on Machine Learning*, pp. 22964–22984, PMLR, 2022.
- [59] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et. al.*, *Language models are few-shot learners*, *Advances in neural information processing systems* **33** (2020) 1877–1901.
- [60] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et. al.*, *Language models are unsupervised multitask learners*, *OpenAI blog* **1** (2019), no. 8 9.

- [61] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, *arXiv preprint arXiv:1412.6980* (2014).
- [62] T. Sellam, D. Das, and A. Parikh, *BLEURT: Learning robust metrics for text generation*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds.), (Online), pp. 7881–7892, Association for Computational Linguistics, July, 2020.
- [63] P. He, X. Liu, J. Gao, and W. Chen, *Deberta: Decoding-enhanced bert with disentangled attention*, *arXiv preprint arXiv:2006.03654* (2020).
- [64] T. Goyal, J. J. Li, and G. Durrett, *News summarization and evaluation in the era of gpt-3*, *arXiv preprint arXiv:2209.12356* (2022).
- [65] L. Yoffe, A. Sharma, and T. Höllerer, *Octopus: Open-vocabulary content tracking and object placement using semantic understanding in mixed reality*, in *2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 587–588, IEEE, 2023.
- [66] B. Nuernberger, E. Ofek, H. Benko, and A. D. Wilson, *Snaptoreality: Aligning augmented reality to the real world*, in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 1233–1244, 2016.
- [67] D. E. Breen, R. T. Whitaker, E. Rose, and M. Tuceryan, *Interactive occlusion and automatic object placement for augmented reality*, in *Computer Graphics Forum*, vol. 15, pp. 11–22, Wiley Online Library, 1996.
- [68] T. Rafi, X. Zhang, and X. Wang, *Predart: Towards automatic oracle prediction of object placements in augmented reality testing*, in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pp. 1–13, 2022.
- [69] Y. Cheng, Y. Yan, X. Yi, Y. Shi, and D. Lindlbauer, *Semanticadapt: Optimization-based adaptation of mixed reality layouts leveraging virtual-physical semantic connections*, in *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 282–297, 2021.
- [70] Y. Lang, W. Liang, and L.-F. Yu, *Virtual agent positioning driven by scene semantics in mixed reality*, in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 767–775, IEEE, 2019.
- [71] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, *Segment anything*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, October, 2023.

- [72] F. Odom, “clip-text-decoder.”
<https://github.com/fkodom/clip-text-decoder>, 2022.
- [73] A. Akbik, D. Blythe, and R. Vollgraf, *Contextual string embeddings for sequence labeling*, in *Proceedings of the 27th International Conference on Computational Linguistics* (E. M. Bender, L. Derczynski, and P. Isabelle, eds.), (Santa Fe, New Mexico, USA), pp. 1638–1649, Association for Computational Linguistics, Aug., 2018.
- [74] W. Kim, B. Son, and I. Kim, *Vilt: Vision-and-language transformer without convolution or region supervision*, in *International conference on machine learning*, pp. 5583–5594, PMLR, 2021.
- [75] T. Lüddecke and A. S. Ecker, *Image segmentation using text and image prompts. in 2022 ieee*, in *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7076–7086, 2021.
- [76] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al., *Grounding dino: Marrying dino with grounded pre-training for open-set object detection*, *arXiv preprint arXiv:2303.05499* (2023).
- [77] X. Huang, Y.-J. Huang, Y. Zhang, W. Tian, R. Feng, Y. Zhang, Y. Xie, Y. Li, and L. Zhang, *Open-set image tagging with multi-grained text supervision*, *arXiv e-prints* (2023) arXiv–2310.
- [78] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan,

- T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O’Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph, *Gpt-4 technical report*, 2024.
- [79] H. Liu, C. Li, Y. Li, and Y. J. Lee, *Improved baselines with visual instruction tuning*, *arXiv preprint arXiv:2310.03744* (2023).
- [80] A. Lampinen, I. Dasgupta, S. Chan, K. Mathewson, M. Tessler, A. Creswell, J. McClelland, J. Wang, and F. Hill, *Can language models learn from explanations in context?*, in *Findings of the Association for Computational Linguistics: EMNLP 2022* (Y. Goldberg, Z. Kozareva, and Y. Zhang, eds.), (Abu Dhabi, United Arab Emirates), pp. 537–563, Association for Computational Linguistics, Dec., 2022.
- [81] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, *et. al.*, *Grounded sam: Assembling open-world models for diverse visual tasks*, *arXiv preprint arXiv:2401.14159* (2024).
- [82] T. Brooks, A. Holynski, and A. A. Efros, *Instructpix2pix: Learning to follow image editing instructions*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023.
- [83] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, *Indoor segmentation and support inference from rgbd images*, in *Computer Vision–ECCV 2012: 12th*

European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12, pp. 746–760, Springer, 2012.

- [84] J. Xiao, A. Owens, and A. Torralba, *Sun3d: A database of big spaces reconstructed using sfm and object labels*, in *Proceedings of the IEEE international conference on computer vision*, pp. 1625–1632, 2013.
- [85] N. Reimers and I. Gurevych, *Sentence-BERT: Sentence embeddings using Siamese BERT-networks*, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (K. Inui, J. Jiang, V. Ng, and X. Wan, eds.), (Hong Kong, China), pp. 3982–3992, Association for Computational Linguistics, Nov., 2019.
- [86] Apple., “ARKit — Apple Developer Documentation — developer.apple.com.” <https://developer.apple.com/documentation/arkit>.
- [87] Apple., “SceneKit — Apple Developer Documentation — developer.apple.com.” <https://developer.apple.com/documentation/scenekit>.
- [88] H. Jun and A. Nichol, *Shap-e: Generating conditional 3d implicit functions*, *arXiv preprint arXiv:2305.02463* (2023).
- [89] Z. Dong, T. Tang, L. Li, and W. X. Zhao, *A survey on long text modeling with transformers*, *arXiv preprint arXiv:2302.14502* (2023).
- [90] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, *et. al.*, *Mixtral of experts*, *arXiv preprint arXiv:2401.04088* (2024).
- [91] D. Jiang, X. He, H. Zeng, C. Wei, M. Ku, Q. Liu, and W. Chen, *Mantis: Interleaved multi-image instruction tuning*, *arXiv preprint arXiv:2405.01483* (2024).
- [92] W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Dong, M. Ding, *et. al.*, *Cogagent: A visual language model for gui agents*, *arXiv preprint arXiv:2312.08914* (2023).
- [93] J. Li, D. Li, S. Savarese, and S. Hoi, *Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models*, in *International conference on machine learning*, pp. 19730–19742, PMLR, 2023.
- [94] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi, *Instructblip: Towards general-purpose vision-language models with instruction tuning*, *Advances in Neural Information Processing Systems* **36** (2024).

- [95] G. Zhang, Y. Zhang, K. Zhang, and V. Tresp, *Can vision-language models be a good guesser? exploring vlms for times and location reasoning*, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 636–645, 2024.
- [96] C. Li, H. Xu, J. Tian, W. Wang, M. Yan, B. Bi, J. Ye, H. Chen, G. Xu, Z. Cao, *et. al.*, *mplug: Effective and efficient vision-language learning by cross-modal skip-connections*, *arXiv preprint arXiv:2205.12005* (2022).
- [97] J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang, *Git: A generative image-to-text transformer for vision and language*, *arXiv preprint arXiv:2205.14100* (2022).
- [98] V.-Q. Nguyen, M. Suganuma, and T. Okatani, *Grit: Faster and better image captioning transformer using dual visual features*, in *European Conference on Computer Vision*, pp. 167–184, Springer, 2022.
- [99] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, *et. al.*, *Pali: A jointly-scaled multilingual language-image model*, *arXiv preprint arXiv:2209.06794* (2022).
- [100] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, *et. al.*, *Image as a foreign language: Beit pretraining for vision and vision-language tasks*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19175–19186, 2023.
- [101] H. Xu, Q. Ye, M. Yan, Y. Shi, J. Ye, Y. Xu, C. Li, B. Bi, Q. Qian, W. Wang, *et. al.*, *mplug-2: A modularized multi-modal foundation model across text, image and video*, in *International Conference on Machine Learning*, pp. 38728–38748, PMLR, 2023.
- [102] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, *Eli5: Long form question answering*, *arXiv preprint arXiv:1907.09190* (2019).
- [103] L. Huang, S. Cao, N. Parulian, H. Ji, and L. Wang, *Efficient attentions for long document summarization*, *arXiv preprint arXiv:2104.02112* (2021).
- [104] A. Dilawari and M. U. G. Khan, *Asovs: abstractive summarization of video sequences*, *IEEE Access* **7** (2019) 29253–29263.
- [105] H.-T. Chen, F. Xu, S. A. Arora, and E. Choi, *Understanding retrieval augmentation for long-form question answering*, *ArXiv abs/2310.12150* (2023).
- [106] J. Phang, Y. Zhao, and P. J. Liu, *Investigating efficiently extending transformers for long input summarization*, *ArXiv abs/2208.04347* (2022).

- [107] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, *Retrieval-augmented generation for knowledge-intensive nlp tasks*, *ArXiv abs/2005.11401* (2020).
- [108] P. Xu, W. Ping, X. Wu, L. C. McAfee, C. Zhu, Z. Liu, S. Subramanian, E. Bakhturina, M. Shoeybi, and B. Catanzaro, *Retrieval meets long context large language models*, *ArXiv abs/2310.03025* (2023).
- [109] Y. Fan, J. Gu, K. Zhou, Q. Yan, S. Jiang, C.-C. Kuo, X. Guan, and X. E. Wang, *Muffin or chihuahua? challenging large vision-language models with multipanel vqa*, *arXiv preprint arXiv:2401.15847* (2024).
- [110] S. Ghosh, C. K. R. Evuru, S. Kumar, U. Tyagi, O. Nieto, Z. Jin, and D. Manocha, *Vdgd: Mitigating llm hallucinations in cognitive prompts by bridging the visual perception gap*, *arXiv preprint arXiv:2405.15683* (2024).
- [111] A. Favero, L. Zancato, M. Trager, S. Choudhary, P. Perera, A. Achille, A. Swaminathan, and S. Soatto, *Multi-modal hallucination control by visual information grounding*, *arXiv preprint arXiv:2403.14003* (2024).
- [112] D. Song, S. Chen, G. H. Chen, F. Yu, X. Wan, and B. Wang, *Milebench: Benchmarking mllms in long context*, *arXiv preprint arXiv:2404.18532* (2024).
- [113] W. Wang, S. Zhang, Y. Ren, Y. Duan, T. Li, S. Liu, M. Hu, Z. Chen, K. Zhang, L. Lu, *et. al.*, *Needle in a multimodal haystack*, *ArXiv preprint abs/2406.07230* (2024).
- [114] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi, *Ok-vqa: A visual question answering benchmark requiring external knowledge*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [115] L. Chen, J. Li, X. Dong, P. Zhang, Y. Zang, Z. Chen, H. Duan, J. Wang, Y. Qiao, D. Lin, *et. al.*, *Are we on the right way for evaluating large vision-language models?*, *arXiv preprint arXiv:2403.20330* (2024).
- [116] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, *Proceedings of the IEEE* **86** (1998), no. 11 2278–2324.
- [117] 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 *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.
- [118] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan, *Seed-bench: Benchmarking multimodal llms with generative comprehension*, *arXiv preprint arXiv:2307.16125* (2023).

- [119] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, *et. al.*, *Mmbench: Is your multi-modal model an all-around player?*, *arXiv preprint arXiv:2307.06281* (2023).
- [120] 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*, *arXiv preprint arXiv:2311.16502* (2023).
- [121] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao, *Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts*, *arXiv preprint arXiv:2310.02255* (2023).
- [122] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi, *A diagram is worth a dozen images*, in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 235–251, Springer, 2016.
- [123] Moondream, “tiny vision language model.” <https://github.com/vikhyat/moondream>, 2024.
- [124] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, *Llava-next: Improved reasoning, ocr, and world knowledge*, January, 2024.
- [125] Google, “Paligemma.” <https://ai.google.dev/gemma/docs/paligemma>, 2024.
- [126] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et. al.*, *Gemini: a family of highly capable multimodal models*, *arXiv preprint arXiv:2312.11805* (2023).
- [127] G. Kamradt, “Needle in a haystack - pressure testing llms.” https://github.com/gkamradt/LLMTest_, 2023.
- [128] M. Saxon, Y. Luo, S. Levy, C. Baral, Y. Yang, and W. Y. Wang, *Lost in translation? translation errors and challenges for fair assessment of text-to-image models on multilingual concepts*, *arXiv preprint arXiv:2403.11092* (2024).
- [129] B. Peng, C. Li, P. He, M. Galley, and J. Gao, *Instruction tuning with gpt-4*, *arXiv preprint arXiv:2304.03277* (2023).
- [130] Y. Li, S. Bubeck, R. Eldan, A. Del Giorno, S. Gunasekar, and Y. T. Lee, *Textbooks are all you need ii: phi-1.5 technical report*, *arXiv preprint arXiv:2309.05463* (2023).
- [131] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, *et. al.*, *Mistral 7b*, *arXiv preprint arXiv:2310.06825* (2023).

- [132] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, *Sigmoid loss for language image pre-training*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.
- [133] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, *et. al.*, *Gemma: Open models based on gemini research and technology*, *arXiv preprint arXiv:2403.08295* (2024).
- [134] SkunkworksAI, “Bakllava.”
<https://huggingface.co/llava-hf/bakLlava-v1-hf>, 2023.
- [135] H. Laurençon, L. Tronchon, M. Cord, and V. Sanh, *What matters when building vision-language models?*, *arXiv preprint arXiv:2405.02246* (2024).