A Closer Look at the Triad in Data-Driven Vision and Language: Curation, Representation, and Learning

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Jihyung Kil

Graduate Program in Computer Science and Engineering

The Ohio State University

2024

Dissertation Committee:

Dr. Wei-Lun Chao, Advisor

Dr. Yu Su

Dr. Andrew Perrault

Abstract

Building advanced Vision and Language (V&L) systems can offer significant societal benefits. For instance, V&L systems with visual question answering capabilities enable visually impaired individuals to perform daily tasks more independently; multimodal web agents streamline our daily activities, such as booking flights or shopping online; embodied robots enhance the efficiency and automation of manufacturing systems. However, developing such sophisticated V&L models is challenging due to the need for an integrated understanding of visual and linguistic information. This integration is particularly complex as it requires models not only to recognize and interpret detailed visual cues but also to understand and generate contextually relevant text.

At its core, data plays an essential role in learning such integrated understanding. The effectiveness of V&L systems relies on how well data is **curated**, **represented**, and **utilized for learning**. In this dissertation, we thus aim to advance V&L systems through the lens of data. **First**, we discuss “data curation” to enrich training materials and benchmarks for V&L models. **Second**, we delve into “data representation” to encode visual and linguistic information from data into meaningful representations. **Third**, we explore “data learning” to enable models to acquire V&L knowledge from data. In short, we investigate three different aspects (*i.e.*, curation, representation, and learning) of data to improve V&L understanding. We believe this comprehensive study greatly contributes to the development of advanced V&L models, ultimately providing substantial benefits to our society.

This dissertation is dedicated to my beloved parents.

Acknowledgments

This dissertation marks the conclusion of my PhD journey at The Ohio State University. Reflecting on this period, I am deeply grateful for the invaluable support and guidance from many individuals throughout this challenging endeavor. I extend my heartfelt thanks to each and every one of them.

First and foremost, I would like to express my deepest gratitude to my advisor, **Dr. Wei-Lun (Harry) Chao**. Harry has guided me in conducting research and shaping my thinking, writing, and presentation skills, all of which have been instrumental in my development as an independent researcher. Without his invaluable advice and support, I would not have been able to complete my Ph.D. journey. It has been a great honor to work with him.

I would also like to give special thanks to **Dr. Yu Su**. Collaborating on multiple projects with him has been incredibly rewarding. His insightful comments and feedback have brought new perspectives that have significantly enhanced the success and impact of my work.

I would like to extend my deepest gratitude to my dissertation defense and proposal committee members, **Dr. Andrew Perrault** and **Dr. Eric Fosler-Lussier** for their interest, valuable time, and insightful feedback.

In addition, my sincere gratitude goes to **Dr. Soravit (Beer) Changpinyo** and **Dr. Hexiang (Frank) Hu**. I truly enjoyed our meetings and conversations during my internship at Google Research. Besides, I was fortunate to work with **Dr. Dongyeop Kang** and

Dr. Joo-Kyung Kim during my internship at Amazon. The experiences at these top-tier industrial research labs have been essential in shaping me into a mature researcher.

I thank all my lab colleagues, including Dr. Cheng Zhang, Dr. Hong-You Chen, David Carlyn, Tai-Yu Pan, Ziwei Li, Cheng-Hao Tu, Zheda Mai, Lisa Wu, Arpita Chowdhury, Jinsu Yoo, Ping Zhang, Tianle Chen, Ana Xiong, Jike Zhong, Reza Averley, and Dr. Dipanjyoti Paul. Furthermore, I would like to express my sincere thanks to Dr. Xi Chen, Dr. Radu Soricut, Dr. Farideh Tavazoee, Chan Hee Song, Boyuan Zheng, and many others for the discussion and collaboration.

Lastly, I thank **my parents** for their unconditional love and support during this journey.

Vita

August, 2018 – Present	Ph.D. Student in Computer Science and Engineering The Ohio State University, Columbus, OH, USA
December, 2023	M.S. in Computer Science and Engineering The Ohio State University, Columbus, OH, USA
August, 2023	Research Intern Amazon Alexa AI, Bellevue, WA, USA
November, 2022	Research Intern Google Research, Los Angeles, CA, USA
December, 2017	B.S. in Computer Science and Engineering The Ohio State University, Columbus, OH, USA

Publications

Preprints.
[P2] *ARES: Alternating Reinforcement Learning and Supervised Fine-Tuning for Enhanced Multi-Modal Chain-of-Thought Reasoning Through Diverse AI Feedback.*
Ju-Seung Byun, Jiyun Chun, Jihyung Kil, Andrew Perrault.
Under Review.

[P1] *CompBench: A Comparative Reasoning Benchmark for Multimodal LLMs.*
Jihyung Kil, Zheda Mai, Justin Lee, Zihe Wang, Kerrie Cheng, Lemeng Wang, Ye Liu, Arpita Chowdhury, Wei-Lun Chao.
Under Review.

Peer-Reviewed Conferences.
[C7] *II-MMR: Identifying and Improving Multi-modal Multi-hop Reasoning in Visual Question Answering.*
Jihyung Kil, Farideh Tavazoee, Dongyeop Kang, Joo-Kyung Kim.
ACL Findings 2024.

[C6] *GPT-4V(ision) is a Generalist Web Agent, if Grounded*.
Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, Yu Su.
ICML 2024.

[C5] *Dual-View Visual Contextualization for Web Navigation*.
Jihyung Kil, Chan Hee Song, Boyuan Zheng, Xiang Deng, Yu Su, Wei-Lun Chao.
CVPR 2024.

[C4] *PreSTU: Pre-Training for Scene-Text Understanding*.
Jihyung Kil, Soravit Changpinyo, Xi Chen, Hexiang Hu, Sebastian Goodman, Wei-Lun Chao, Radu Soricut.
ICCV 2023.

[C3] *One Step at a Time: Long-Horizon Vision-and-Language Navigation with Milestones*.
Chan Hee Song, Jihyung Kil, Tai-Yu Pan, Brian M Sadler, Wei-Lun Chao, and Yu Su.
CVPR 2022.

[C2] *Discovering the Unknown Knowns: Turning Implicit Knowledge in the Dataset into Explicit Training Examples for Visual Question Answering*.
Jihyung Kil, Cheng Zhang, Dong Xuan, and Wei-Lun Chao.
EMNLP 2021.

[C1] *Revisiting Document Representations for Large-Scale Zero-Shot Learning*.
Jihyung Kil, and Wei-Lun Chao.
NAACL 2021.

Fields of Study

Major Field: Computer Science and Engineering

Studies in:

Artificial Intelligence	Dr. Wei-Lun Chao
Database	Dr. Arnab Nandi
Graphics	Dr. Han-Wei Shen

Table of Contents

Page

VI Conclusion 125

Appendices 129

List of Tables

List of Figures

Part I: Introduction

Chapter 1: Introduction

1.1 Motivation and Overview

Building a highly intelligent machine has long been a dream in Artificial Intelligence (AI). In the past few years, we have witnessed unprecedented progress in Deep Learning and its applications, which leads us one step closer to this dream. In Computer Vision (CV), AI system has surpassed human-level performance on image classification [94, 63] or achieved prominent improvements in various vision tasks such as object detection [216] or semantic segmentation [93, 136]. Similarly, with the emergence of foundation models [3, 255, 250], the field of NLP has seen notable achievements in multiple tasks, including language proficiency exams [54, 97], code generation [43, 286], and commonsense reasoning [291].

While remarkable progress has been made in both modalities (*i.e.*, vision and language), understanding each modality "independently" is not sufficient to build a highly intelligent AI system. As humans, we typically develop our intelligence through the "mixture" of various resources (*e.g.*, vision, language, and audio), rather than relying on a single source of information (Figure 1.1). For instance, when asked questions about images, we utilize knowledge from both vision (*i.e.*, images) and language (*i.e.*, questions and answers). Similarly, students leverage the integrated understanding of audio (*i.e.*, sound) and language

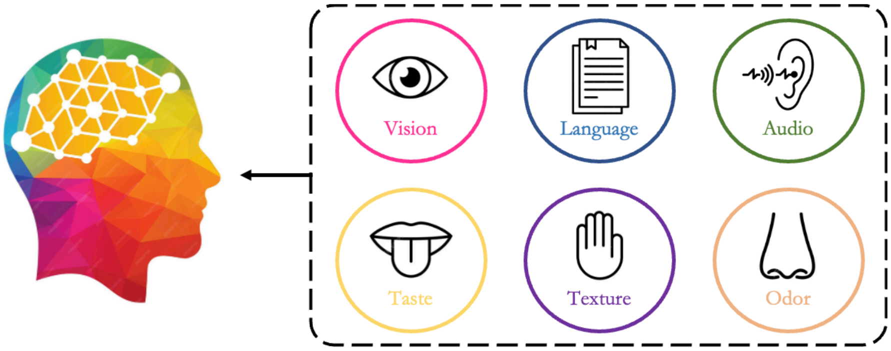


Figure 1.1: **Human intelligence developed by multimodal knowledge.**

(*i.e.*, text) to take notes during lectures. Thus, understanding how these modalities integrate into knowledge in human cognition is essential for developing human-level AI systems.

Vision and Language (V&L) is one of the primary research areas to learn such multimodal knowledge. Concretely, it involves the integrated understanding of visual and linguistic information from images and text. This integrated understanding requires diverse capabilities, such as (i) recognizing objects in images, (ii) comprehending the semantic meaning of textual descriptions, (iii) grounding images with text, (iv) answering image-related questions, (v) reading scene-text and layout structures in images, (vi) learning to reason in the context of spatiality, commonsense, and composition, (vii) understanding temporal information, such as the agent's previous actions during navigation, and so on.

Fundamentally, data is key to learning all these capabilities. Concretely, V&L systems can acquire these abilities based on how well the data is **curated**, **represented**, and **utilized for learning**. In this dissertation, we thus aim to advance V&L systems through the "lens of data". **First**, we focus on how to "curate" data for V&L. Data is the fuel of model training, and constructing sufficient and high-quality training data is essential for models to

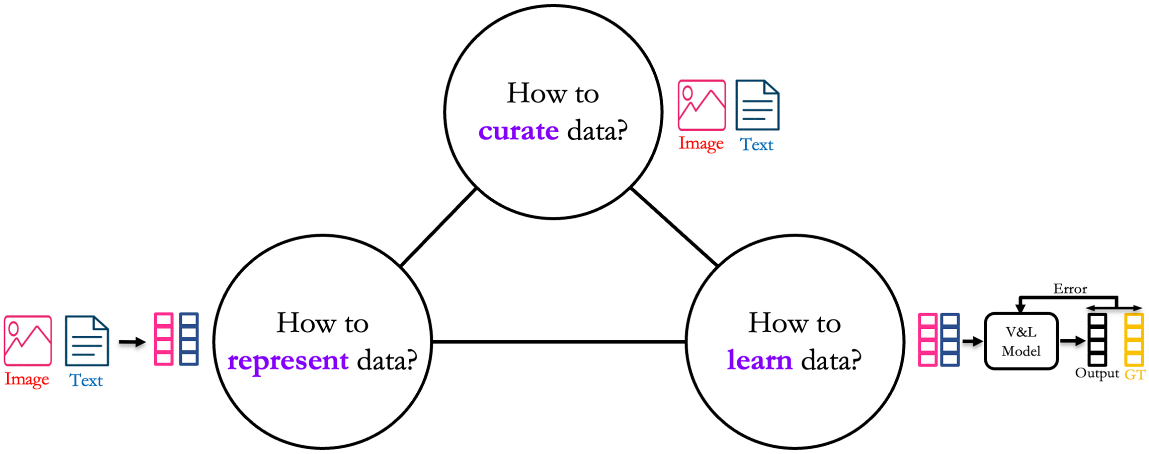


Figure 1.2: **Overview of dissertation.**

acquire knowledge. Our emphasis is on efficiently curating extensive high-quality V&L data (*e.g.*, image and text) while minimizing costs and human effort. Additionally, we discuss data curation for designing a V&L benchmark, which evaluates various reasoning capabilities of recent V&L models. This enables us to gain a deeper understanding of their current limitations in V&L tasks. **Second**, we explore how to "represent" data in V&L. For effective model training, data should be transformed into suitable representations. V&L involves two input modalities: images and text, each encoded into visual and textual representations, respectively. We discuss strategies to align these representations to enhance the grounding capabilities of V&L models. **Third**, we investigate how to "learn" from V&L data. Specifically, we focus on designing learning objectives for pre-training that enable models to solve downstream V&L tasks more effectively.

In summary, we investigate three different aspects of data (*i.e.*, curation, representation, and learning) to advance V&L systems (Figure 1.2). This thorough study will greatly enhance the development of V&L systems, bringing substantial benefits to our society.

1.2 Representatives of V&L tasks

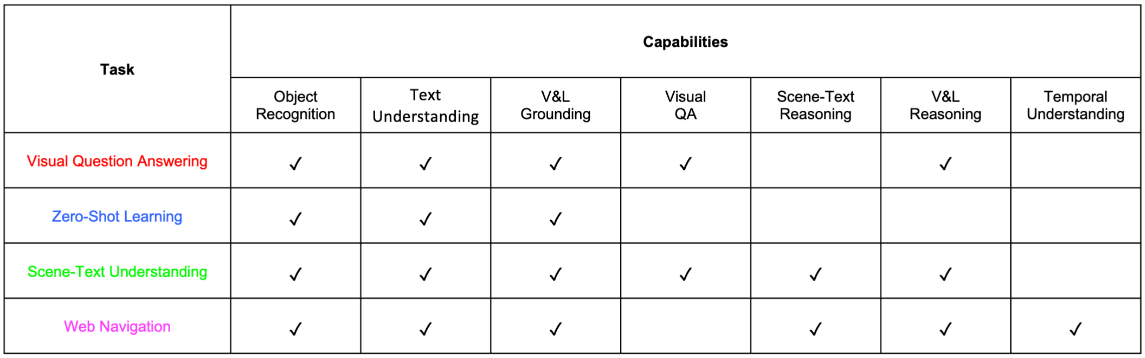
We note that currently, there is no single V&L task that simultaneously evaluates all the capabilities of V&L models mentioned in §1.1. Therefore, in this dissertation, we explore multiple V&L tasks to cover all these capabilities. For example, zero-shot learning (ZSL) in image classification involves categorizing images into classes that models have not encountered during training by leveraging semantic information about the unseen classes. This evaluates the models' ability to recognize objects, understand text, and ground images on text. Visual question answering (VQA) instead asks questions about images: It tests not only the capabilities evaluated in ZSL but also the models' question-answering and reasoning abilities, such as composition and commonsense. Scene-text understanding (STU) additionally requires scene-text reasoning, which involves interpreting text within images. Finally, web navigation requires models to follow language instructions and execute complex web-related tasks, necessitating the understanding of temporal information like the history of previous actions. In other words, a capability present in one task may be absent in another, leading us to investigate multiple V&L tasks to enhance all these abilities (Figure 1.3). We believe this multifaceted study will help the V&L community understand these capabilities better, ultimately paving the way for more advanced V&L models.

1.3 Dissertation Outline

The rest of the dissertation is structured as follows:

Part II: Related Work.

Chapter 2 discusses the recent developments in V&L systems. Specifically, we investigate the latest approaches in datasets, architectures, and learning strategies.



Task	Capabilities						
	Object Recognition	Text Understanding	V&L Grounding	Visual QA	Scene-Text Reasoning	V&L Reasoning	Temporal Understanding
Visual Question Answering	✓	✓	✓	✓		✓	
Zero-Shot Learning	✓	✓	✓				
Scene-Text Understanding	✓	✓	✓	✓	✓	✓	
Web Navigation	✓	✓	✓		✓	✓	✓

Figure 1.3: **V&L tasks and their focused capabilities.**

Part III: V&L Data Curation.

Chapter 3 introduces a new data augmentation method for visual question answering (VQA) to synthesize VQA data, enhancing the depth of training materials.

Chapter 4 proposes a novel V&L benchmark designed to understand the comparative reasoning capabilities of current V&L models.

Part IV: V&L Data Representations.

Chapter 5 introduces new semantic representations that closely align with visual features for zero-shot image classification.

Chapter 6 proposes novel HTML representations by contextualizing HTML elements through their dual views in webpage screenshots for web navigation.

Part V: V&L Data Learning.

Chapter 7 develops new pre-training learning strategies that are essential for solving scene-text-related V&L tasks.

Part VI: Conclusion.

Chapter 8 concludes this dissertation and remarks on future directions.

Other Doctoral Work. Besides my research mentioned above, I have worked on other V&L domains or techniques to further improve the V&L systems. [239] proposes a new planning method for long-horizon V&L navigation. At a high level, our planner monitors the embodied agent's progress step-by-step and determines when to proceed to the next step. This enables the agent to reach its final destination more effectively. Additionally, [299] evaluates whether the current leading V&L models such as GPT4-V [195] or Gemini-pro [251] can serve as agents for web navigation. We found that there still exists a performance gap between models and humans and thus suggest novel visual prompting approaches to notably reduce this gap. Finally, [132] focuses on improving Chain-of-Thought Reasoning (CoT) for VQA by leveraging the model's prediction to guide its reasoning toward answers.

Part II: Related Work

Chapter 2: Recent Developments in V&L systems

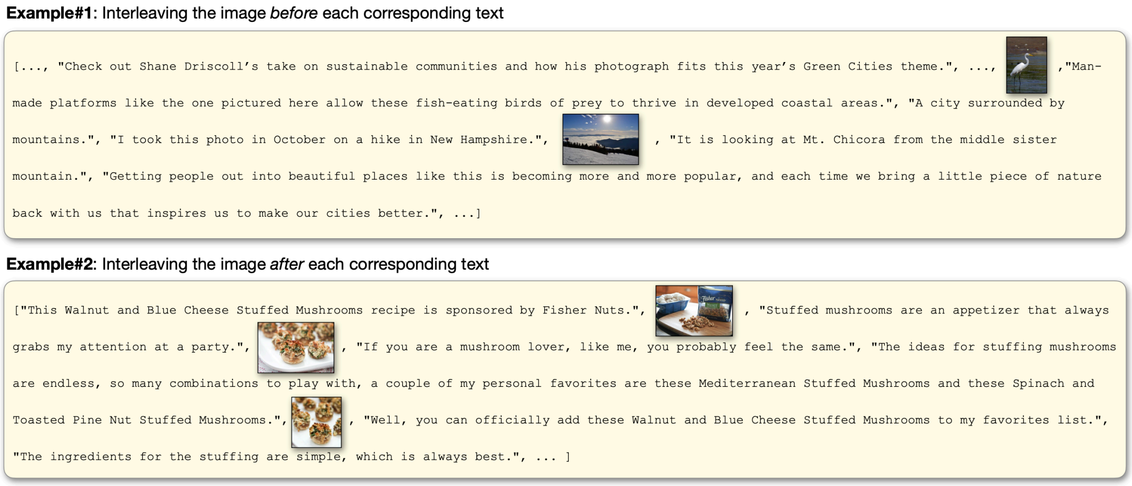
In recent years, we have observed significant advancements in Vision-Language (V&L) research. Concretely, V&L models have achieved remarkable results across various tasks, including visual question answering (VQA) [81, 106], image captioning [46, 7], image-text retreival [156, 47], scene-text understanding [237, 235], web navigation [60], expert-level multimodal understanding [288], and more.

In fact, V&L is a unified framework comprising several components such as dataset, architecture, and learning strategy. Numerous different approaches have been proposed to improve each of these components. In this chapter, we will explore the recent developments in each component to better understand the current trends in V&L.

2.0.1 Pre-training Dataset

Pre-training models is now the default paradigm for learning various V&L knowledge from data. These pre-trained models can be directly utilized to solve V&L tasks in zero-shot or in-context learning fashions, or they can be further trained on downstream datasets to perform specific tasks (i.e., fine-tuning). In either case, learning V&L knowledge from pre-training has proven effective and has become the go-to approach in V&L research.

One crucial aspect of pre-training is constructing a suitable pre-training dataset. There exist many different approaches to curating these datasets.



Example#1: Interleaving the image *before* each corresponding text

[..., "Check out Shane Driscoll's take on sustainable communities and how his photograph fits this year's Green Cities theme.", ..., ,"Man-made platforms like the one pictured here allow these fish-eating birds of prey to thrive in developed coastal areas.", "A city surrounded by mountains.", "I took this photo in October on a hike in New Hampshire.", , "It is looking at Mt. Chicora from the middle sister mountain.", "Getting people out into beautiful places like this is becoming more and more popular, and each time we bring a little piece of nature back with us that inspires us to make our cities better.", ...]

Example#2: Interleaving the image *after* each corresponding text

["This Walnut and Blue Cheese Stuffed Mushrooms recipe is sponsored by Fisher Nuts.", , "Stuffed mushrooms are an appetizer that always grabs my attention at a party.", , "If you are a mushroom lover, like me, you probably feel the same.", "The ideas for stuffing mushrooms are endless, so many combinations to play with, a couple of my personal favorites are these Mediterranean Stuffed Mushrooms and these Spinach and Toasted Pine Nut Stuffed Mushrooms.", , "Well, you can officially add these Walnut and Blue Cheese Stuffed Mushrooms to my favorites list.", "The ingredients for the stuffing are simple, which is always best.", ...]

Figure 2.1: **Examples of interleaved image text pairs from Multimodal C4 [303].**

Interleaved image-text corpus. One popular format for V&L pre-training datasets is the interleaved image-text pairs [303, 143, 187]. This format involves inserting an image before or after each corresponding text (Figure 2.1). Interleaved pairs help models learn joint representations of visual and textual information more effectively. Furthermore, this format enhances robustness in multi-image or in-context learning tasks, where multiple images or image-text pairs are given as input. Recently, several leading V&L models [13, 163, 187] have shown the effectiveness of this interleaved pre-training corpus on multiple V&L tasks.

Image-caption pairs. Pairing a caption for the corresponding image [229, 39, 224] is the most traditional method for preparing a V&L pre-training dataset. Concretely, this process entails gathering pairs from billions of web pages and applying a filtering process to generate a large-scale, clean visual-linguistic corpus. Compared to interleaved image-text pairs, this approach is more easily scalable as it only requires short captions. However, the simplicity of captions might not effectively teach models more complex V&L reasoning, such as compositional or spatial reasoning. Thus, the most popular and effective approach today is to use a mixture of interleaved image-text pairs and image-caption pairs [163, 187], aiming to increase scalability and enhance reasoning diversity simultaneously.

Domain-specific data. Interleaved image-text (or image-caption) pairs may lack sufficient domain-specific knowledge since they are collected from arbitrary websites. This limitation can be problematic, particularly if our goal is to solve domain-specific tasks. Recently, several works [85, 146, 75] have addressed this limitation by collecting large-scale domain-specific data. One example is the use of HTML content for pre-training. Specifically, [85, 146, 75] gather massive webpage screenshots and their corresponding HTML code snippets for pre-training, which significantly improves model performance on downstream HTML-related tasks, such as web navigation.

Syntehsized data. Several studies [155, 169] have attempted to automatically augment V&L pre-training datasets by using the power of generative AI models [195, 211]. ITIT [155] utilizes both image-to-text and text-to-image models to generate text and image, iteratively (Figure 2.2). LLaVA [169] synthesizes various instruction-following V&L data for conversations and complex reasoning tasks, using the LLM [195] combined with visual information such as object labels, bounding boxes, and related captions. This approach is advantageous as it can easily transform image-only (or text-only) data into image-text pairs without significant human effort.

2.0.2 V&L Architecture

At a high level, most recent V&L systems [169, 57, 163, 195, 251] adopt an architecture comprising three components: a vision encoder, a projector, and a language model (Figure 2.3). Concretely, the vision encoder extracts visual representations from the image, and the projector maps these representations into the language embedding space. The system

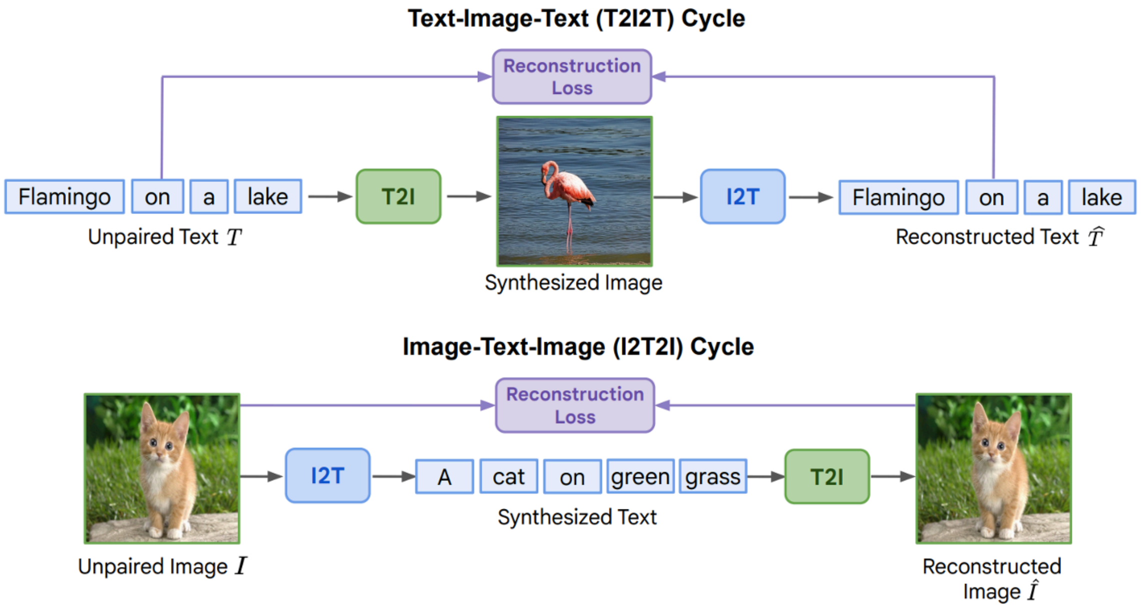


Figure 2.2: **Learning pipeline of ITIT [155].**

then passes the visual representations, along with textual representations, into the language model to solve V&L tasks.

Vision encoder aims to embed images into visual representations. The most popular vision encoder is ViT from CLIP [207], a transformer-based model pre-trained on large-scale image-text pairs using contrastive training. Concretely, it learns the visual representations by aligning images with the corresponding text. While the CLIP-based vision encoder has shown effectiveness in multiple V&L tasks, its representations are primarily encoded with "global" associations with the caption. Thus, these representations may have limitations in fine-grained V&L tasks, such as phrase grounding, where identifying fine-grained correspondences between phrases in a sentence and objects (or regions) in an image is crucial. In contrast, GLIP [154] learns visual representations at a more fine-grained level by aligning tokens in a caption with regions in the image during pre-training and thus shows notable results on such fine-grained V&L tasks. Besides, since CLIP focuses on image-text alignment, its visual representations may cover fewer details about images than those of DINO [196], a vision-only self-supervised encoder designed to learn exclusively from images. To address this, [254] proposes representing images as a combination of CLIP and DINO features to encode both image-text alignment and sufficient image details during V&L pre-training.

Projector serves as the bridge module between the two modalities, vision and language. A crucial aspect of projector design is how to map visual representations into the language embedding space. Flamingo [13] utilizes cross-attention layers to provide visual embeddings as context to the language model. InstructBLIP [57] instead uses Q-former to represent images as a set of object queries, which are then passed along with textual input to the language model. More recent V&L models, such as LLaVA [169] and VILA [163], opt for a simpler approach by using a few MLP layers as the projector.

Language model takes projected visual representations along with textual representations as input and generates textual output for V&L tasks. Nearly all recent leading V&L models employ large language models (LLMs) as their language components. Specifically, models like [195, 251, 169, 163] adopts a decoder-only LLM, while others such as [45, 44, 150] rely on an encoder-decoder LLM. These models are often enormous, with sizes ranging from 13 billion [169] to 540 billion parameters [65].

2.0.3 Learning Strategies

Various learning objectives have been proposed to enable models to learn V&L knowledge from data during pre-training. The most popular method is visual language modeling [195, 57, 251, 163, 169, 20], which generates textual tokens sequentially from visual

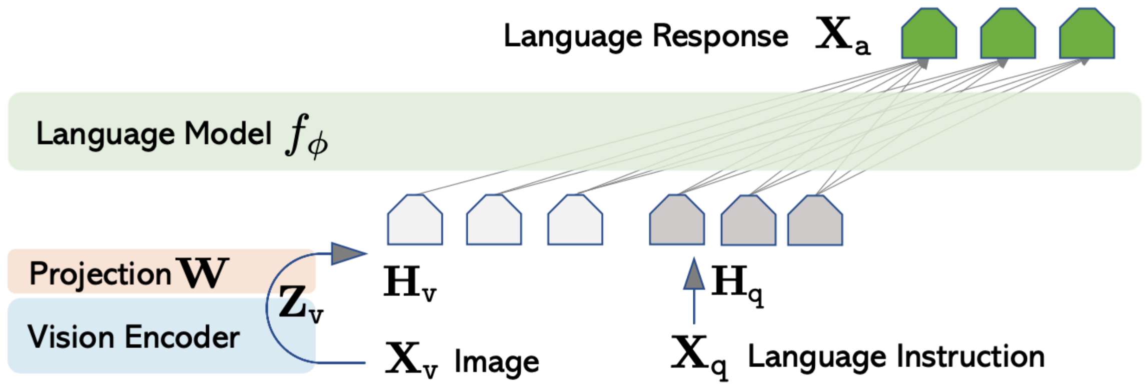


Figure 2.3: **LLaVA Architecture.**

and textual inputs. Besides, contrastive learning with image-text pairs [207, 197] has been widely used to align visual representations with textual representations. Denoising tasks such as span corruption [45], masked language modeling [176], and masked autoencoders [92] have been utilized to capture the context among textual or visual tokens. More recently, PaLI-X [44] combines all these learning objectives during pre-training to harness their collective benefits.

Instruction-tuning [52], which fine-tunes models on instruction-following data, has proven effective in various NLP tasks. Recent V&L works [169] have extended this training strategy to V&L domains by leveraging visual instruction-following data. Additionally, other studies [245, 34] have explored incorporating Reinforcement Learning from Human feedback (RLHF) or Natural Language Explanation Feedback (RLNF) into V&L models to enhance their reasoning capabilities.

Part III: V&L Data Curation

Chapter 3: Enriching V&L training materials with data augmentation

Our journey starts with data curation, with a particular focus on **data augmentation** for V&L tasks. Data augmentation is an essential technique to enhance training examples, especially since gathering these examples manually can be both time-consuming and expensive. In this chapter, we present a novel data augmentation method specifically designed for Visual Question Answering, one of the prominent V&L tasks. Essentially, we harness the implicit information within the existing VQA dataset to automatically synthesize VQA data and leverage this augmented data to improve model training.

3.1 Introduction

"A picture is worth a thousand words," which tells how expressive an image can be, but also how challenging it is to teach a machine to understand an image like we humans do. Visual question answering (VQA) [19, 82, 307] is a principled way to measure such an ability of a machine, in which given an image, a machine has to answer the image-related questions in natural language by natural language. While after years of effort, the state-of-the-art machine's performance is still behind what we expect [103, 287, 16, 177, 107].

Several key bottlenecks have been identified. In particular, a machine (*i.e.*, VQA model) learned in the conventional supervised manner using human-annotated image-question-answer (IQA) triplets is shown to overlook the image or language contents [5, 82, 40],

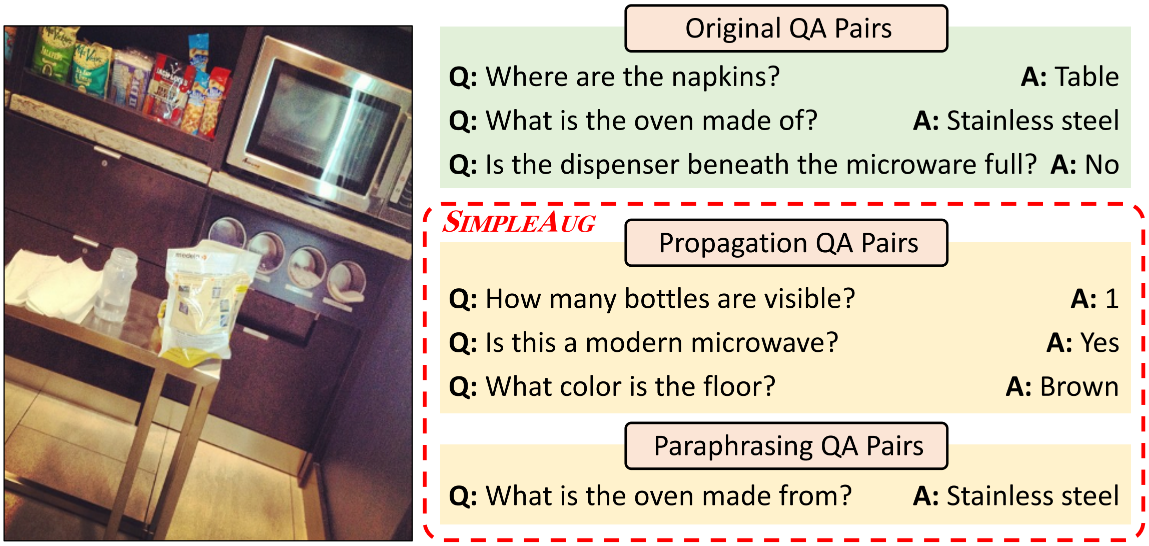

Original QA Pairs
Q: Where are the napkins? A: Table
Q: What is the oven made of? A: Stainless steel
Q: Is the dispenser beneath the microware full? A: No
SIMPLEAUG
Propagation QA Pairs
Q: How many bottles are visible? A: 1
Q: Is this a modern microwave? A: Yes
Q: What color is the floor? A: Brown
Paraphrasing QA Pairs
Q: What is the oven made from? A: Stainless steel

Figure 3.1: **Illustration of our approach SIMPLEAUG**. We show a *training* image and its corresponding question-answer pairs in VQA v2 [82], and our generated pairs. A VQA model [16] trained on the original dataset *just cannot answer these new questions on the training image correctly*, and we use them to improve model training.

over-fit the language bias [6], or struggle in capturing the diversity of human language [228, 41]. Many recent works thus propose to augment the original VQA task with auxiliary tasks or losses such as visual grounding [226, 275], de-biasing [32, 53, 210], or (cycle-)consistency [228, 78] to address these issues.

Intrigued by these findings and solutions, we investigate the bottlenecks further and argue that they may result from a more fundamental issue — there are simply not enough training examples (*i.e.*, IQA triplets). Concretely, most of the existing VQA datasets annotate each image with around ten questions, which are much fewer than what we humans can ask about an image. Take the popular VQA v2 dataset [82], for instance, the trained VQA model can answer most of the training examples (on average, six questions per image) correctly. However, if we ask some more questions about the training images — *e.g.*, by borrowing relevant questions from other training images — the same VQA model fails drastically, *even if the model has indeed seen these images and questions during training* (see Figure 3.1). Namely, the VQA model just has not learned enough through the human-annotated examples, leaving the model unaware of the huge amount of visual information in an image and how it can be asked via natural language.

At first glance, this seems to paint a grim picture for VQA. However, in this work, we propose to take advantage of this weakness to strengthen the VQA model: we turn implicit information already in the dataset, such as unique questions and the rich contents in the training images, into explicit IQA triplets which can be directly used by VQA models via conventional supervised learning.

We propose a simple data augmentation method SIMPLEAUG, which relies on (i) the original image-question-answer triplets in the dataset, (ii) mid-level semantic annotations available on the training images (*e.g.*, object bounding boxes), and (iii) pre-trained object detectors [218][1]. Concretely, we build upon the aforementioned observations — *questions annotated for one image can be valuable add-ons to other relevant images* — and design a series of mechanisms to "propagate" questions from one image to the others (Figure 3.2). More specifically, we search images that contain objects mentioned in the question and identify the answers using information provided by (ii) and (iii), such as numbers of objects, their attributes, and existences.

SIMPLEAUG requires no question generation step via templates or language models [122], bypassing the problems of limited diversity or artifacts. Besides, SIMPLEAUG is completely detached from the training phase of a VQA model and is therefore model-agnostic, making it fairly simple to use to improve VQA models.

[1] We note that (ii) is commonly provided in existing VQA datasets like VQA v2 [82], and (iii) has been widely used in the feature extraction stage of a VQA model [16].

We validate SIMPLEAUG on two datasets, VQA v2 [82] and VQA-CP [6]. The latter is designed to evaluate VQA models' generalizability under language bias shifts. With SIMPLEAUG, we can not only achieve comparable gains to other existing methods on VQA-CP, but also boost the accuracy on VQA v2, demonstrating the applicability of our method. We note that many of the prior works designed for VQA-CP indeed degrade the accuracy on VQA v2, which does not have language bias shifts between training and test data. SIMPLEAUG further justifies that mid-level vision tasks like object detection can effectively benefit high-level vision tasks like VQA.

In summary, our contributions are three-folded:

- We propose SIMPLEAUG, a simple and model-agnostic data augmentation method that turns information already in the datasets into explicit IQA triplets for training VQA models.
- We show that SIMPLEAUG can notably improve VQA models' accuracy on both VQA v2 [82] and VQA-CP [53].
- We provide comprehensive analyses on SIMPLEAUG, including its applicability to weakly-labeled and unlabeled images.

3.2 Prior VQA studies

VQA datasets. More than a dozen datasets have been released [165, 19, 307, 82, 141, 107, 87]. Most of them use natural images from large-scale image databases, *e.g.*, MSCOCO [165]. For each image, human annotators are asked to generate questions (Q) and provide the corresponding answers (A). Doing so, however, is hard to cover all the knowledge in the visual contents.

Leveraging side information for VQA. A variety of side information beyond the IQA triplets has been used to improve VQA models. For example, human attentions are used to enhance the explainability and visual grounding of VQA models [200, 226, 58, 275]. Image captions contain substantial visual information and can be used as an auxiliary task (*i.e.*, visual captioning) to strengthen VQA models' visual and language understanding [274, 134, 272, 21, 130]. Several papers leveraged scene graphs and visual relationships as auxiliary knowledge for VQA [120, 294, 107, 232]. A few works utilized mid-level vision tasks (*e.g.*, object detection and segmentation) to benefit VQA [72, 122]. Most of these works use side information by defining auxiliary learning tasks or losses to the original VQA task. In contrast, we directly turn the information into IQA triplets for training.

Data augmentation for VQA. Several existing works investigate data augmentation. One stream of works creates new triplets by manipulating images or questions [42, 4, 248, 78]. See § 3.4.1.4 for some more details. The other creates more questions by using a learned language model to paraphrase sentences [212, 228, 273, 126, 21] or by learning a visual question generation model [122, 160, 140].

The closest to ours is the pioneer work of data augmentation by [122], which also creates new questions by using mid-level semantic information annotated by humans (*e.g.*, object bounding boxes). Their question generation relies on either pre-defined templates or a learned language model, which may suffer limited diversity or labeling noise. In contrast, we directly reuse questions already in the dataset and show that they are sufficient to augment high-quality questions for other images. Besides, we further explore machine generated annotations (*e.g.*, via an object detector [218]), opening the door to augment triplets using extra unlabeled images. Furthermore, we benchmark our method on the popular VQA v2 [82] and challenging VQA-CP [6] datasets, which are released after the publication of

[122]. *Overall, we view our paper as an attempt to revisit simple data augmentation like [122] for VQA, and show that it is indeed quite effective.*

Robustness of VQA models. [82, 40, 6] pointed out the existence of superficial correlations (*e.g.*, language bias) in the datasets and showed that a VQA model can simply exploit them to answer questions. Existing works to address this can be categorized into three groups. The first group attempts to reduce the language bias by designing new VQA models or learning strategies [6, 210, 32, 53, 84, 53, 119, 192, 233, 76]. For example, RUBi and Ensemble [32, 53] explicitly modeled the question-answer correlations to encourage VQA models to explore other patterns in the data that are more likely to generalize. The second group leverages side information to facilitate visual grounding [275, 226, 252]. For example, [275] used extra visual or textual annotations to determine important regions where a VQA model should focus on. The third group implicitly or explicitly augments the VQA datasets, *e.g.*, via self-supervised learning, counterfactual sampling, adversarial training, or image and question manipulation [1, 304, 253, 42, 78, 161, 79, 153, 219, 227]. SIMPLEAUG belongs to the third group but is simpler in terms of methodology. Besides, SIMPLEAUG is completely detached from VQA model training and thus model-agnostic. Moreover, SIMPLEAUG can improve on both VQA-CP [6] and VQA v2 [82].

3.3 SIMPLEAUG for Data Augmentation

3.3.1 Implicit information

SIMPLEAUG leverages three sources of information that implicitly suggest extra IQA triplets beyond those provided in a VQA dataset. The **first** one is the original IQA triplets in the dataset. We find that for two similar images that locally share common objects or globally share common layouts, their corresponding annotated questions can either be treated as paraphrases or extra questions for each other. The **second** one is the object instance labels like object bounding boxes that are annotated on images, *e.g.*, MSCOCO images [165]. These labels provide accurate answers to "how many" or some of the "what" questions, and many VQA datasets are built upon MSCOCO images. The **third** one is an object detector pre-trained on an external densely-annotated dataset like Visual Genome (VG) [141]. This detector can provide information not commonly annotated on images, such as attributes or fine-grained class names. We note that since the seminal work by [16], many following-up VQA models use the Faster R-CNN detector [218] pre-trained on VG for feature extraction.

3.3.2 The SIMPLEAUG pipeline

SIMPLEAUG processes each annotated IQA triplet (i, q, a) in term, and propagates q to other relevant images. To begin with, SIMPLEAUG extract meaningful words from the question, similar to [42, 275]. We leverage a spaCy part-of-speech (POS) tagger [100] to extract "nouns" and tokenize their singular and plural forms. We remove words such as "picture" or "photo", which appear in many questions but are not informative for VQA[2].

Given the meaningful "nouns" of a question, we then retrieve relevant images and derive the answers, using MSCOCO annotations or Faster R-CNN detection. Concretely, we split questions into four categories and develop specific *question propagation rules*. Figure 3.2 illustrates the pipeline.

Yes/No questions. We apply the Faster R-CNN detector trained on VG to each image i' beside image i. The detector returns a set of bounding boxes and their labels. We ignore

[2] For example, in a question, "What is the person doing in the picture?", the word "picture" refers to the image itself, not an object within it. We found 8% of the questions like this in VQA v2 [82]. Some questions really refer to "pictures" or "photos" within an image (*e.g.*, "How many pictures on the wall?"), but there are <1% such questions.

images whose object labels have no overlap with the nouns of question q, and assign answer "yes" or "no" to the remaining images as follows.

- "Yes": if the labels of image i' cover all nouns of question q, we create (i', q, yes).
- "No": if the labels of image i' only cover some of the nouns of q, we create (i', q, no). For instance, if the question is "Is there a cat on the pillow?" but the image only contains "pillow" but no "cat", then the answer is "no".

We develop two verification strategies at the end of this subsection to filter out outlier cases.

Color questions. To prevent ambiguous cases, we only consider questions with a single noun (besides the word "color"). We again apply the Faster R-CNN detector, which returns for each image a set of object labels that may also contain attributes like colors. We keep images whose labels cover the noun of question q. For each such image i', we create a triplet $(i', q, \hat{a})$, where $\hat{a}$ is the color attribute provided by the detector. As there are likely some other object-color pairs in i', we investigate replacing the noun in q by each detected object name and create some more IQA triplets about colors.

Number questions. We again focus on questions with a single noun (besides the word "number"). We use MSCOCO annotations, which give each image a set of object bounding boxes and labels. We find all the images whose labels cover the noun of question q. For each such image i', we derive the answer by counting annotated instances of that noun and create a triplet $(i', q, \hat{a})$, where $\hat{a}$ is the count. Some of the nouns (*e.g.*, "animal") are super-categories of sub-category objects (*e.g.*, "dog" or "cat"). Thus, if the noun of q is a super-category (*e.g.*, q is "How many animals are there?"), we follow the category hierarchy provided by MSCOCO and count all its sub-category instances.

Other questions. We focus on "what" questions with a single noun and use MSCOCO annotations. We find all the images whose labels cover the noun of question q. (We also take the super-category cases into account.) For each such image i', we check whether its MSCOCO labels contain the answer a of question q, *i.e.*, according to the original (i, q, a) triplet. For instance, if q is "What animal is this?" and a is "sheep", then we check if image i''s labels contain "sheep". If yes, we create a triplet $(i', q, \hat{a} = a)$. This process essentially discovers "what" questions that can indeed be asked about i'.

Verification. The above rules simplify a question by only looking at its nouns, so they may lead to triplets whose answers are incorrect. To mitigate this issue, we develop two verification strategies.

The first strategy performs self-verification on the original (i, q, a) triplet, checking if our rules can reproduce it. That is, it applies the aforementioned rules to image i to derive the new triplet $(i, q, \hat{a})$. If $\hat{a}$ does not match a, *i.e.*, using the rules creates a different answer, we skip this question q.

The second strategy verifies our rules using IQA triplets annotated on retrieved image i'. For example, if image i' has an annotate triplet (i', q', a') whose q' has the same category and nouns as q, then we compare it to the created triplet $(i', q, \hat{a})$. If $\hat{a}$ does not match a', then we disregard $(i', q, \hat{a})$.

3.3.3 Paraphrasing by similar questions

Besides the four question propagation rules that look at the image contents, we also investigate a simple paraphrasing rule by searching similar questions in the dataset. Concretely, we apply the averaged word feature from BERT [62] to encode each question as it better captures the object-level semantics for searching questions mentioning the same

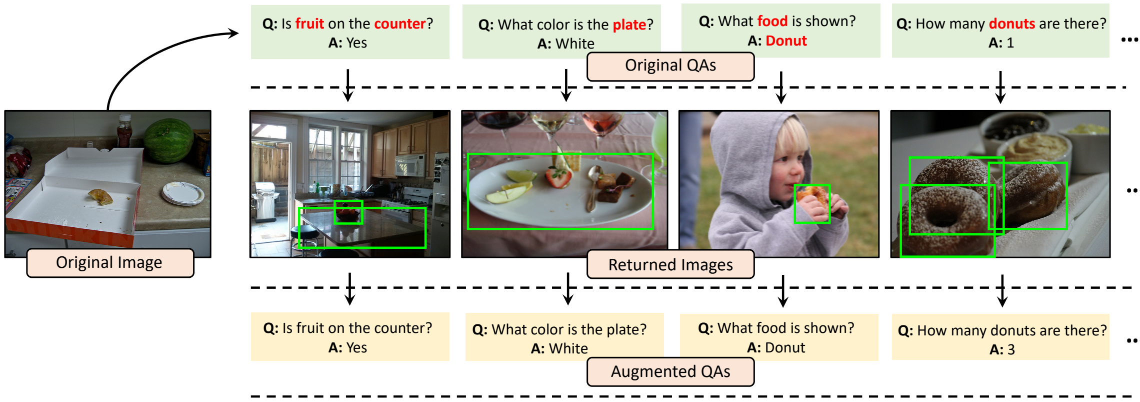

Q: Is fruit on the counter?
A: Yes
Q: What color is the plate?
A: White
Q: What food is shown?
A: Donut
Q: How many donuts are there?
A: 1
...
Original QAs
Original Image
Returned Images
...
Q: Is fruit on the counter?
A: Yes
Q: What color is the plate?
A: White
Q: What food is shown?
A: Donut
Q: How many donuts are there?
A: 3
...
Augmented QAs

Figure 3.2: **The SimpleAug pipeline.** We show four original question-answer pairs of the image on the left in VQA v2, and how they are propagated to other images. The green boxes are annotated in MSCOCO or detected by Faster R-CNN; each of them is associated with an object name and/or attribute. We only show boxes matched by nouns or used to derive answers.

objects. Two questions are similar if their cosine similarity is above a certain threshold (0.98 in the experiments). If two IQA triplets (i,q,a) and (i',q',a') have similar questions, we create two extra triplets (i,q',a) and (i',q,a') by switching their questions as paraphrasing.

We choose a high threshold 0.98 to avoid false positives. On average, each question finds 11.4 similar questions, and we only pick the top-3 questions. We found that with this design, an extra verification step, like checking if the image i' contains nouns of the paraphrasing question q, does not further improve the overall VQA accuracy. Thus, we do not include an extra verification step for paraphrasing.

3.4 Experiments

3.4.1 Experimental setup

3.4.1.1 VQA datasets and evaluation metrics

We validate SimpleAug on two popular datasets.

VQA v2 [82] collects images from MSCOCO [165] and uses the same training/validation/testing splits. On average, six questions are annotated for each image. In total, VQA v2 has 444K/214K/448K training/validation/test IQA triplets.

VQA-CP v2 [6] is a challenging *adversarial* split of VQA v2 designed to evaluate the model's capability of handling language bias/prior shifts between training and testing. For instance, "white" is the most frequent answer for questions that start with "what color..." in the training set whereas "black" is the most common one in the test set. Such *prior changes* also reflect in individual questions, *e.g.*, the most common answer for "What color is the banana?" changes from "yellow" during training to "green" during testing. VQA-CP v2 has 438K/220K training/test IQA triplets.

Evaluation metrics. We follow the standard evaluation protocol [19, 82]. For each test triplet, the predicted answer is compared with answers provided by ten human annotators in a leave-one-annotator-out fashion for robust evaluation. We report the averaged scores over all test triplets as well as over test triplets of Yes/No, number, or other answer types.

3.4.1.2 Implicit knowledge sources

MSCOCO annotations [165]. MSCOCO is the most popular benchmark nowadays for object detection and instance segmentation, which contains 80 categories (*e.g.*, "cat") as well as the corresponding super categories (*e.g.*, "animal"). Object instances of all 80 categories are exhaustively annotated in all images, leading to approximately 1.2 million instance annotations. **Faster R-CNN detection** [16]. We use the object detection results from a Faster R-CNN [218] pre-trained with Visual Genome (VG) [141]. This pre-trained detector can provide object attributes (*e.g.*, color and material) whereas MSCOCO annotations only contain object names (*e.g.*, “person” and “bicycle”). We use the detector provided by [16], which detects 36 objects per image.

3.4.1.3 Base VQA models

SIMPLEAUG is model-agnostic, and we evaluate it by using its generated data to augment the training set for training three base VQA models.

Bottom-Up Top-Down (UpDn) [16]. UpDn is a widely used VQA model. It first detects objects from an image and encodes them into visual feature vectors. Given a question, UpDn uses a question encoder to produce a set of word features. Both visual and language features are then fed into a multi-modal attention network to predict the answer.

Learned-Mixin+H (LMH) [53]. LMH is a learning strategy to de-bias a VQA model, *e.g.*, UpDn. During training, LMH uses an auxiliary question-only model to encourage the VQA model to explore visual-question related information. During testing, only the VQA model is used. LMH is shown to largely improve the performance on VQA-CP v2 but can hurt that on VQA v2. **LXMERT** [247]. We also study SIMPLEAUG with a stronger, transformer-based VQA model named LXMERT. LXMERT leverages multi-modal transformers to extract multi-modal features, and exploits a masking mechanism to better (pre-)train the model. While such a masking mechanism can be viewed as a way of data augmentation, SIMPLEAUG is fundamentally different from it in two aspects. First, SIMPLEAUG generates new triplets while masking manipulates existing triplets. Second, SIMPLEAUG is detached from model training and is therefore compatible with masking. As will be shown in the experimental results, SIMPLEAUG can provide solid gains to LXMERT on both VQA v2 and VQA-CP.

3.4.1.4 Compared data augmentation methods

We compare SIMPLEAUG with three existing data augmentation methods for VQA.

Template-based augmentation proposed by [122] generates new question-answer pairs using MSCOCO annotations. We re-implement the method following the paper.

Counterfactual Samples Synthesizing (CSS) [42] generates counterfactual triplets by masking critical objects in images or words in questions and assigning different answers. These new training examples force the VQA model to focus on those critical objects and words, improving both visual explainability and question sensitivity.

MUTANT [78] is a state-of-the-art data augmentation method by manipulating images and questions. For example, it applies a GAN-based inpainting network to change the object's color to create extra color questions; it manipulates object numbers using MSCOCO annotations; it masks or negates words to mutate questions.

Comparison. SIMPLEAUG is different from CSS and MUTANT in two aspects. First, CSS and MUTANT can only manipulate already annotated questions for an image, while we can create new questions for an image by borrowing them from other images. Second, CSS needs a pre-trained attention-based VQA model to identify critical objects/words while MUTANT's best version requires additional loss terms for training. In contrast, SIMPLEAUG is completely detached from model training.

3.4.2 Implementation details

Data augmentation by SIMPLEAUG. We speed up the implementation by grouping IQA triplets of the same unique question and only propagating the question once. We remove redundant triplets if the retrieved image already has the same question. To prevent creating too many triplets from paraphrasing (§3.3.3), for each question q we only search for its

Table 3.1: **Statistics on VQA-CP v2 training data.** Miss-answered: the number of SIMPLEAUG examples that a UpDn model trained on the original dataset cannot answer correctly.

# of samples	**All**	**Y/N**	**Num**	**Other**
Original	438K	183K	52K	202K
SIMPLEAUG	5,457K	2,062K	1,937K	1,458K
Miss-answered	3,081K	974K	1,489K	618K

top-3 similar questions q' and only create (i', q, a') if a' is a rare answer to q — we define a' to be a rare answer if there are fewer than five (q, a') pairs in the dataset. *We emphasize that we only apply* SIMPLEAUG *to IQA triplets in the training set and search images in the training set.*

VQA models. For the base VQA models, we use the released code from corresponding papers.

Training with SIMPLEAUG triplets. We explore three ways to train with the original (O) triplets and augmented triplets (A). The first is to train with both from the beginning ($A+O$); the second is to train with O first and then with both ($O \rightarrow A+O$); the third is to train with O first, then with A, and then with O again ($O \rightarrow A \rightarrow O$). The rationale of training with multiple stages is to prevent the augmented data from dominating the training process (see Table 3.1 for the statistics). We note that, there is a huge number of SIMPLEAUG examples that a VQA model trained with O only cannot answer. Thus, when training with multiple stages, we remove SIMPLEAUG examples that the model can already answer. We mainly report results using $O \rightarrow A \rightarrow O$, but compare the three ways in §3.4.4.

Base	Method	VQA v2 val				VQA-CP test			
		All	Y/N	Num	Other	All	Y/N	Num	Other
UpDn	Baseline (Anderson et al., 2018)	63.48	81.18	42.14	55.66	39.74	42.27	11.93	46.05
	AdvReg (Ramakrishnan et al., 2018)	62.75	79.84	42.35	55.16	41.17	65.49	15.48	35.48
	RUBi (Cadene et al., 2019)	61.16	–	–	–	44.23	67.05	17.48	39.61
	CF-VQA (SUM) (Niu et al., 2021)	63.54	82.51	43.96	54.30	53.55	91.15	13.03	44.97
	SimpleReg (Shrestha et al., 2020)	62.60	–	–	–	48.90	69.80	11.30	47.80
	HINT (Selvaraju et al., 2019)	63.38	81.18	42.99	55.56	46.73	67.27	10.61	45.88
	SCR+VQA-X (Wu and Mooney, 2019)	62.20	78.80	41.60	54.50	49.45	72.36	10.93	48.02
	RandImg (Teney et al., 2020)	57.24	76.53	33.87	48.57	55.37	83.89	41.60	44.20
	Template-based (Kafle et al., 2017)	63.83	81.61	41.98	56.10	39.75	43.03	14.98	44.83
	CSS (Chen et al., 2020)	63.47	80.81	43.33	55.62	41.16	43.96	12.78	47.48
	MUTANT (plain) (Gokhale et al., 2020a)	–	–	–	–	50.16	61.45	35.87	50.14
	MUTANT (loss) (Gokhale et al., 2020a)	62.56	82.07	42.52	53.28	61.72	88.90	49.68	50.78
	SimpleAug (paraphrasing)	63.66	81.44	42.56	55.72	52.57	86.56	13.52	45.47
	SimpleAug (propagation)	64.37	81.91	44.13	56.40	52.27	65.15	45.32	47.42
	SimpleAug (propagation + paraphrasing)	64.34	81.97	43.91	56.35	52.65	66.40	43.43	47.98
LMH	Baseline (Clark et al., 2019)	56.34	65.05	37.63	54.68	52.01	72.58	31.11	46.96
	RMFE (Gat et al., 2020)	–	–	–	–	54.44	74.03	49.16	45.82
	CSS (Chen et al., 2020)	59.91	73.25	39.77	55.11	58.95	84.37	49.42	48.21
	CSS+CL (Liang et al., 2020)	57.29	67.27	38.40	54.71	59.18	86.99	49.89	47.16
	MUTANT (loss) (Gokhale et al., 2020a)	–	–	–	–	55.38	90.99	39.74	40.99
	SimpleAug (paraphrasing)	61.67	78.70	40.21	54.41	53.29	74.12	33.06	47.93
	SimpleAug (propagation)	62.67	79.24	41.44	55.70	53.58	73.58	37.07	47.63
	SimpleAug (propagation + paraphrasing)	62.63	79.31	41.71	55.48	53.70	74.79	34.32	47.97
LXMERT	Baseline (Tan and Bansal, 2019)	73.06	88.30	56.81	65.78	48.66	47.49	22.24	56.52
	Template-based (Kafle et al., 2017)	72.30	85.36	54.47	67.10	49.63	49.96	36.33	53.10
	MUTANT (plain) (Gokhale et al., 2020a)	–	–	–	–	59.69	73.19	32.85	59.29
	MUTANT (loss) (Gokhale et al., 2020a)	70.24	89.01	54.21	59.96	69.52	93.15	67.17	57.78
	SimpleAug (paraphrasing)	74.37	88.78	57.95	67.76	59.09	73.17	28.72	60.04
	SimpleAug (propagation)	74.96	89.00	60.00	68.25	61.82	68.39	53.35	60.69
	SimpleAug (propagation + paraphrasing)	74.98	89.04	59.98	68.25	62.24	69.72	53.63	60.69

Figure 3.3: **Performance on VQA v2 val set and VQA-CP v2 test set.** Our method SimpleAug (cyan background) consistently improves all answer types for different base models on both VQA v2 and VQA-CP. Note that MUTANT (loss) [78] (gray color) applies extra loss terms besides data augmentation.

3.4.3 Main results on VQA v2 and VQA-CP v2

Figure 3.3 summarizes the main results on VQA v2 val and VQA-CP v2 test. We experiment SIMPLEAUG with different base VQA models and compare it to state-of-the-art methods. SIMPLEAUG achieves consistent gains against the base models on all answer types (columns). When paired with LXMERT, SIMPLEAUG obtains the highest accuracy on both datasets, except MUTANT (loss) which applies extra losses besides data augmentation.

SIMPLEAUG improves all answer types. On **VQA-CP v2**, SIMPLEAUG boosts the overall accuracy of UpDn from 39.74% to 52.65%, outperforming all but three methods. One key strength of SIMPLEAUG is that it improves all the answer types, including a $\sim$2% gain on "Other" where many methods suffer. Specifically, compared to CF-VQA [192] and RandImg [253] which have higher overall accuracy than SIMPLEAUG, SIMPLEAUG outperforms them on the challenging "Num" and "Other". On **VQA v2**, SIMPLEAUG achieves the highest accuracy using UpDn, improving +0.86% on "All", +0.79% on "Yes/No", +1.77% on "Num", and +0.69% on "Other". Other methods specifically designed for VQA-CP v2 usually degrade on VQA v2.

SIMPLEAUG is model-agnostic. SIMPLEAUG can directly be applied to other VQA models. Besides UpDn, in Figure 3.3 we show that SIMPLEAUG can lead to consistent gains for two additional VQA models. LMH is a de-biasing method for UpDn, which however hurts the accuracy on VQA v2. With SIMPLEAUG, LMH can largely improve on VQA v2. LXMERT is a strong transformer-based VQA model, and SIMPLEAUG can also improve upon it, achieving the highest accuracy on VQA v2 (all answer types) and on VQA-CP v2 ("Other").

Comparison to data augmentation baselines. SIMPLEAUG notably outperforms the **template-based method** [122], the closest method to ours. We attribute this to the question

Table 3.2: SIMPLEAUG **(propagation) w/ or w/o verification (§3.3.2) on VQA-CP v2, using the UpDn model.**

Method	**Verification**	**All**	**Y/N**	**Num**	**Other**
UpDn	–	39.74	42.27	11.93	46.05
SIMPLEAUG	✗	51.96	64.02	44.44	47.70
	✓	52.27	65.15	45.32	47.42

diversity via question propagation and paraphrasing. Compared to **CSS** [42, 161], SIMPLEAUG performs better on all answer types on both datasets, using UpDn. While CSS outperforms SIMPLEAUG on VQA-CP v2 using the de-biasing LMH, its improvement on VQA v2 is smaller than SIMPLEAUG. Since LXMERT is a general VQA method like UpDn, we expect that SIMPLEAUG will outperform CSS. Finally, compared to **MUTANT** [78], SIMPLEAUG achieves better results on VQA-CP v2 against the version without extra loss terms (*i.e.*, MUTANT(plain)). It is worth noting that while CSS and MUTANT both generate extra data, they cannot improve but degrade on VQA v2 (when using UpDn or LXMERT). In contrast, SIMPLEAUG improves on all cases, suggesting it as a more general data augmentation method for VQA.

3.4.4 Ablation studies of SIMPLEAUG

Question propagation vs. paraphrasing. SIMPLEAUG leverages the original IQA triplets by propagating questions to other images (§3.3.2) or by paraphrasing question using similar questions (§3.3.3). Propagation can ask more questions about an image. For example, the propagated questions in Figure 3.1 and Figure 3.4 ask about image contents different from the original questions. In contrast, paraphrasing only paraphrases the original questions of that image. As shown in Figure 3.3, question propagation generally leads to better

Table 3.3: **A comparison of training strategies on VQA-CP v2 with the UpDn model.** O: original triplets. A: augmented triplets by SIMPLEAUG.

Method	Strategy	All	Y/N	Num	Other
UpDn	O	39.74	42.27	11.93	46.05
	$O \to O \to O$	39.47	43.11	11.75	45.16
SIMPLEAUG	$A + O$	47.50	59.76	38.18	43.63
	$O \to A + O$	49.73	59.67	36.58	48.12
	$O \to A \to O$	52.65	66.40	43.43	47.98

performance, especially on "Num" and "Other" answers, suggesting the importance of creating additional questions to cover image contents more exhaustively.

On verification for question propagation. Table 3.2 compares SIMPLEAUG (propagation) with and without the verification strategies (§3.3.2). Verification improves accuracy at nearly all cases.

Multiple-stage training. In Table 3.3, we compare the three training strategies with original triplets (O) and augmented triplets (A). We also train on O for multiple stages (*i.e.*, more epochs) for a fair comparison. $O \to A \to O$ in general outperforms others, and we attribute this to the clear separation of clean and noisy data — the last training stage may correct noisy information learned in early stages [295].

Training with SIMPLEAUG triplets alone. We further investigate training the UpDn model with augmented triplets alone (A). On VQA-v2, we get 39.62% overall accuracy, worse than the baseline trained with original data (63.48%). This is likely due to the noise in the augmented data. On VQA-CP, we get 51.60%, much better than the baseline (39.74%) but worse than training with both augmented and original triplets (52.65%). We surmise that SIMPLEAUG triplets help mitigate the language bias shifts in VQA-CP.

Table 3.4: **Effects of different augmention types (§3.3.2).** We report results on VQA-CP v2, using the UpDn model.

Method	Aug Type	All	Y/N	Num	Other
UpDn	–	39.74	42.27	11.93	46.05
SimpleAug	Y/N	47.20	**68.63**	12.12	45.68
	Num	44.62	42.87	**43.80**	45.77
	Color	40.97	43.11	12.39	47.68
	Other	41.22	43.17	12.19	**48.16**
	All	**52.65**	66.40	43.43	47.98

Effects of augmentation types. We experiment with propagating each question type alone on VQA-CP v2, using UpDn as the base model. In Table 3.4, we show the separate results of SimpleAug with different question types. The augmented questions notably improve the corresponding answer type.

3.4.5 SimpleAug in additional scenarios

We explore SimpleAug in the scenarios where there are (i) limited questions per image, and (ii) extra weakly-labeled or unlabeled images. For (ii), both have no IQA triplets but the weakly-labeled ones have human-annotated object instances.

Learning with limited triplets. We randomly keep a fraction of annotated QA pairs for each training image on VQA-CP v2. Table 3.5 shows that even under this annotation-scarce setting (*e.g.*, only 10% of QA pairs are kept), SimpleAug can already be effective, outperforming the baseline UpDn model trained with all data. This demonstrates the robustness of SimpleAug on dealing with the challenging setting with limited triplets.

Learning with weakly-labeled or unlabeled images. We simulate the scenarios by keeping the QA pairs for a fraction of images (*i.e.*, labeled data) and removing the QA pairs

Table 3.5: **Learning with limited IQA triplets on VQA-CP v2.** We keep a certain fraction of QA pairs per image.

Method	Fraction	All	Y/N	Num	Other
UpDn	1.00	39.74	42.27	11.93	46.05
SIMPLEAUG	1.00	52.65	66.40	43.43	47.98
	0.50	47.67	57.65	37.77	45.17
	0.25	46.03	52.01	37.96	45.12
	0.10	42.91	45.09	30.24	45.25

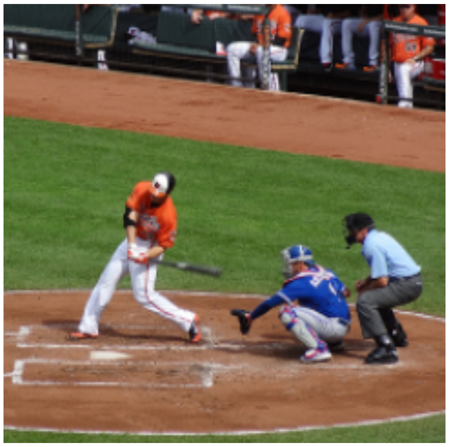
entirely for the other images. Conventionally, a VQA model cannot benefit from the images without QA pairs, but SIMPLEAUG could leverage them by propagating questions to them. Specifically, for images without QA pairs, we consider two cases. We either keep their MSCOCO object instance annotations (*i.e.*, weakly-labeled data) or completely rely on object detectors (*i.e.*, unlabeled data). Table 3.6 shows the results, in which we only apply SIMPLEAUG to the weakly-labeled and unlabeled images. As shown, SIMPLEAUG yields consistent improvements, opening up the possibility of leveraging additional images to improve VQA.

3.4.6 Qualitative results

We show a training image and its augmented QA pairs by SIMPLEAUG in Figure 3.4. A VQA model trained on the original IQA triplets cannot answer many of the newly generated questions, even if the image is in the training set, showing the necessity to include them for training a stronger model. More qualitative results can be found in Figure 3.5.

Table 3.6: **Learning with weakly-labeled and unlabeled images for VQA v2.** Fraction: the portion of images with annotated QA pairs. GT: MSCOCO ground truth annotations. OD: Faster R-CNN object detection. ✗: supervised training with only labeled VQA training examples.

Fraction	SimpleAug	All	Y/N	Num	Other
1.00	✗	63.48	81.18	42.14	55.66
	✓	64.34	81.97	43.91	56.35
0.50	✗	60.93	78.45	40.74	52.96
	GT	61.47	78.92	41.43	53.50
	OD	61.47	78.93	41.42	53.50
0.25	✗	56.70	74.02	37.81	48.53
	GT	57.54	74.49	39.08	49.54
	OD	57.56	74.63	38.67	49.57
0.10	✗	51.06	69.18	33.46	41.93
	GT	52.18	69.76	35.95	43.10
	OD	52.27	69.98	35.07	43.34



Original Question	Answer	
What color are the empty seats?	*Green*	✗
How many people are on the field?	*3*	✓
What team is playing?	*Orioles*	✓

Augmented Question	Answer	
How many baseball bats are in the picture?	*1*	✗
How many baseball gloves are showing?	*1*	✗
What color is the helmet?	*Blue*	✗
How many people are in the field?	*3*	✓

Figure 3.4: **Qualitative results.** We show the training image and its QA pairs from VQA-CP, and the generated QA pairs by SimpleAug. ✓/✗ indicates if the baseline VQA model (trained without SimpleAug) answers correctly/incorrectly. In augmented QA pairs, the first three are from question propagation and the last one is by paraphrasing.

3.5 Summary

We proposed SIMPLEAUG, a data augmentation method for VQA that can turn information already in the datasets into explicit IQA triplets for training. SIMPLEAUG is simple but by no means trivial. First, it justifies that mid-level vision tasks like object detection can effectively benefit VQA. Second, we probably will never be comprehensive enough in annotating data, and SIMPLEAUG can effectively turn what we have at hand (*i.e.*, "knowns") to examples a VQA model wouldn't have known (*i.e.*, "unkowns"). SIMPLEAUG can notably improve the accuracy of VQA models on both VQA v2 [82] and VQA-CP v2 [53].

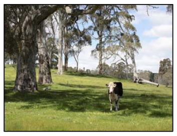
Image	Augmented QA Pairs			Original QA Pairs	
	Question	Answer		Question	Answer
	• How many giraffe? • What color is his eyes? • What color is the giraffe? • What color is the sign?	3 Black Brown Red	✗ ✓ ✗ ✗	• Are the giraffes resting their heads? • Are the humans on the ground? • What is the fence made of?	Yes Yes Wood
	• How many animals are in this? • How many cows are visible? • Are the cows hornless? • What color is the cow? • What color are the rocks?	1 1 Yes White Gray	✗ ✗ ✗ ✗ ✗	• Which animals are seen? • Is this a farm? • Is the cow under a tree?	Cow No Yes
	• How many of these animals are laying down? • How many cats are pictured? • What color is the suitcase? • How many suitcases are they?	1 1 Brown 1	✗ ✗ ✗ ✗	• Is this a good place for the cat to sleep? • Is this an old suitcase? • What kind of cat? • What are the cats laying on?	Yes Yes Black Suitcase
	• How many people are clearly visible in this picture? • How many people are standing around? • How many people are actually in the photo? • What color is the sky?	2 2 2 Blue	✗ ✗ ✗ ✗	• How many tires are there? • What sport is the equipment for? • About what time of day is it?	6 Biking Daytime
	• What color is the elephant? • What color are his legs?	Gray Gray	✗ ✗	• Are the elephants mad? • How many animals are here? • What animals are shown?	Yes 4 Elephant
	• What color is the stove? • What color is the hair? • What color is the pants? • What color is the table? • How many people are actually in this photo?	White Black Black Gray 2	✓ ✗ ✗ ✗ ✗	• How sanitary does the counter look? • How many clear glass bowls are on the counter? • What is the counter made of?	Clean 5 Steel
	• What color are the leaves on the tree? • What color is the field? • What is the color of the cloud? • How many elephants in the picture? • How many animals are seen?	Green Green White 4 5	✓ ✗ ✗ ✗ ✗	• Sunny or overcast? • Is the water fresh looking? • Is there a man in the picture? • Are any of the elephants on the dirt road? • What is the water on the ground?	Overcast No Yes Yes Mud

Figure 3.5: **Additional qualitative results on VQA-CP.** We show the original image, the generated QA pairs by SIMPLEAUG, and the original QA pairs. ✓/✗ indicates if the baseline VQA model (trained without SIMPLEAUG) predicts correctly/incorrectly.

Chapter 4: Curating data for a V&L benchmark

Data curation is useful not only for enriching training materials but also for developing benchmarks to evaluate the capabilities of V&L models. In this chapter, we explore the direction of **benchmark** development and highlight the strengths and weaknesses of the current leading V&L models.

The ability to compare objects, scenes, or situations is crucial for effective decision-making and problem-solving in everyday life. For instance, comparing the freshness of apples enables better choices during grocery shopping, while comparing sofa designs helps optimize the aesthetics of our living space. Despite its significance, the comparative capability is largely unexplored in recent V&L models.

In this work, we introduce COMPBENCH, a V&L benchmark designed to evaluate the comparative reasoning capability of V&L models, also known as multimodal large language models (MLLMs). COMPBENCH mines and pairs images through visually oriented questions covering eight dimensions of relative comparison: visual attribute, existence, state, emotion, temporality, spatiality, quantity, and quality. We curate a collection of around 40K image pairs using metadata from diverse vision datasets and CLIP similarity scores. These image pairs span a broad array of visual domains, including animals, fashion, sports, and both outdoor and indoor scenes. The questions are carefully crafted to discern relative characteristics between two images and are labeled by human annotators for accuracy and relevance. We use CompBench to evaluate recent MLLMs, including GPT-4V(ision), Gemini-Pro, and LLaVA-1.6. Our results reveal notable shortcomings in their comparative abilities. We believe CompBench not only sheds light on these limitations but also establishes a solid foundation for future enhancements in the comparative capability of MLLMs.

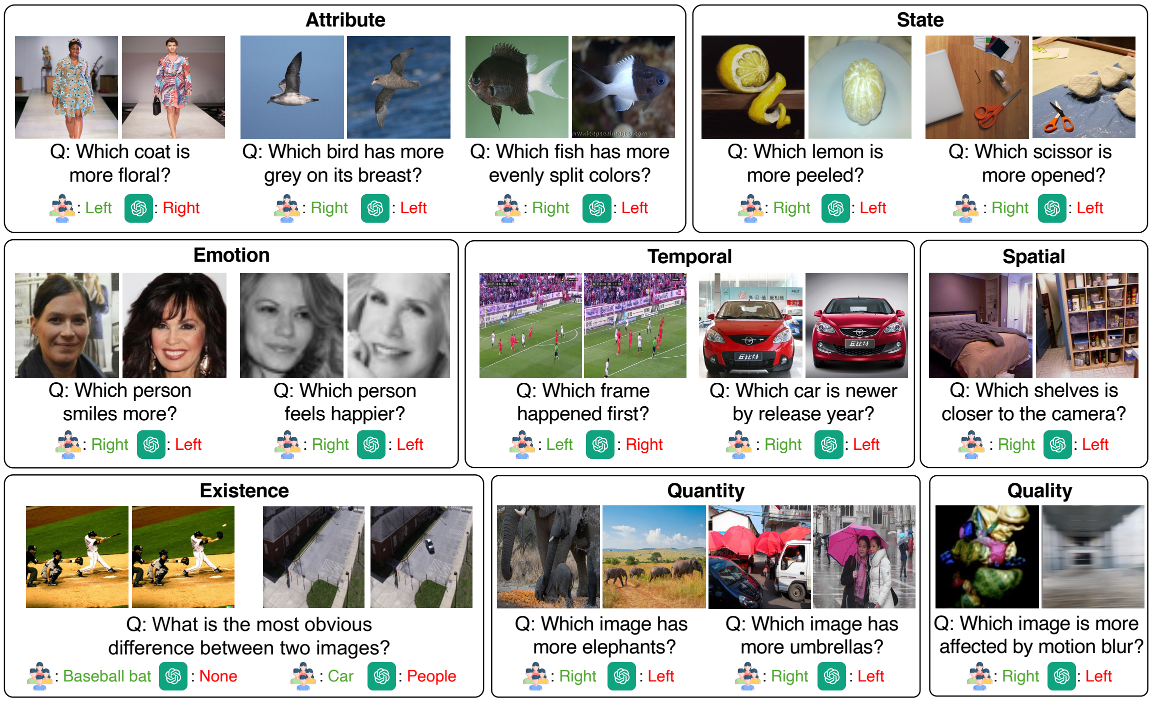

Attribute
Q: Which coat is more floral?
: Left : Right
Q: Which bird has more grey on its breast?
: Right : Left
Q: Which fish has more evenly split colors?
: Right : Left
State
Q: Which lemon is more peeled?
: Right : Left
Q: Which scissor is more opened?
: Right : Left
Emotion
Q: Which person smiles more?
: Right : Left
Q: Which person feels happier?
: Right : Left
Temporal
Q: Which frame happened first?
: Left : Right
Q: Which car is newer by release year?
: Right : Left
Spatial
Q: Which shelves is closer to the camera?
: Right : Left
Existence
Q: What is the most obvious difference between two images?
: Baseball bat : None
: Car : People
Quantity
Q: Which image has more elephants?
: Right : Left
Q: Which image has more umbrellas?
: Right : Left
Quality
Q: Which image is more affected by motion blur?
: Right : Left

Figure 4.1: **CompBench** offers diverse triplets comprising two images, a question about their relativity, and an answer to cover eight types of relativity (see §4.1). See examples along with predictions of GPT-4V [2].

4.1 Introduction

The concept of "relativity" is integral in our daily lives. For example, relative freshness affects our decision to purchase fruits; relative spaciousness affects our decision to choose living or working space; relative crowdedness indicates which paths to select; (relative) change between two scenes reveals what happened to the environment. In short, the ability to compare objects, scenes, or situations and reason about their relativity is vital for us to make informed decisions, solve problems effectively, and acquire knowledge efficiently, enabling us to make sense of the surrounding world.

The recent advance of multimodal large language models (MLLMs), a.k.a. large multimodal models (LMMs), [2, 12, 251, 169, 163, 57, 20] has demonstrated promising progress toward artificial general intelligence (AGI) [288, 179] and achieved unprecedented results in a variety of vision and language (V&L) tasks, ranging from free-formed visual recognition [59, 46, 56] and visual captioning [46, 7] to visual question answering [81, 106, 225]. Yet, much less attention has been paid to tasks that involve relativity and comparison between multiple visual inputs, *e.g.*, two images. In essence, most of the existing datasets for visual recognition [59, 46, 56] and V&L tasks [81, 7, 186, 156, 60, 288] comprise examples with only single visual inputs (*e.g.*, an image or a video clip), making them infeasible to assess MLLMs' comparative capability.

In this paper, we introduce COMPBENCH, a V&L benchmark dedicated to evaluating the comparative reasoning capabilities of MLLMs (Figure 4.1). COMPBENCH comprises 39.8K triplets, each containing 1) a *pair* of visually or semantically relevant images 2) a question about their relativity, and 3) a ground-truth answer. We consider a wide range of questions categorized into eight aspects of relativity. **Attribute Relativity** tests the ability to recognize relative attributes [199] such as size, color, texture, shape, and pattern. For instance, given two images of birds, we ask MLLMs to compare the length of their beaks (*e.g.*, "Which bird has longer beaks?"). **Existential Relativity** assesses the comprehension of existence in comparisons, asking questions like "Which trait is in the left butterfly but not in the right butterfly?" **State/Emotion Relativity** examines if MLLMs can identify state variations, such as different degrees of baking and smiling. **Temporal Relativity** evaluates the understanding of time-related changes between two objects or scenes (*e.g.*, "Which video frame happens earlier during a free kick?"). **Spatial Relativity** checks the ability to tell spatial differences (*e.g.*, "Which cup looks further?"). Finally, **Quantitiy/Quality Relativity** investigates whether an MLLM understands the relativity of quantity and quality (*e.g.*, "Which image contains more animal instances?").

We systematically benchmark representative MLLMs on COMPBENCH, including GPT-4V [2], Gemini1.0-Pro [251], LLaVA-1.6 [169], and VILA-1.5 [163]. Specifically, we concatenate two images horizontally (*i.e.*, left and right) as the visual input. We then prompt MLLMs to answer questions about the relativity between these two images. When applicable, we also investigate a two-stage reasoning strategy, starting by asking a refined question about each image independently (*e.g.*, "How many animal instances are in the image?"), followed by a pure language question (*e.g.*, "Based on the descriptions, which image has more animal instances?"). Our results reveal notable shortcomings in existing MLLMs' comparative abilities, especially in Existence, Spatiality, and Quantity Relativity. We conduct further analyses of error cases, offering insights for future MLLMs' improvements.

In sum, COMPBENCH has several advantages: (i) COMPBENCH introduces new perspectives to evaluate MLLMs — comparative reasoning capabilities about relativity. (ii) COMPBENCH provides extensive coverage across eight relativities and fourteen domains. (iii) COMPBENCH benchmarks recent MLLMs, accompanied by detailed analyses and insights for future improvement. (iv) COMPBENCH is extensible — we identify multiple data sources that can be further incorporated.

4.2 Background in Multimodal LLMs

Multimodal LLMs (MLLMs). Large Language Models (LLMs) [2, 251, 17, 18, 110, 256, 276] have made significant strides in various NLP and AI tasks. Many recent works [2, 12, 251, 169, 163, 57, 20, 151, 301, 202, 267] have extended LLMs' capabilities into the multimodal domain, particularly for vision and language (V&L) tasks. At a higher level, this advancement involves integrating a pre-trained vision encoder (*e.g.*, CLIP [207]) with LLMs via a bridge module (*e.g.*, an adaptor [169, 57]). Different strategies are developed to pre-train these multimodal LLMs (MLLMs), such as optimizing the LLMs and bridge module while keeping the vision encoder frozen [169] or training the bridge part only [57].

MLLM benchmarks. Earlier, MLLMs were evaluated on traditional V&L tasks, such as visual question answering (VQA) [81, 106, 225], image captioning [46, 7], and image-text retreival [156, 47]. Recently, a range of new and intriguing V&L tasks [180, 237] have emerged to assess MLLMs' capabilities across various dimensions. These include comprehension and reasoning about charts [184], diagrams [185], scene text [235, 186], web navigation [60], expert-level multimodal understanding [288], etc. Our COMPBENCH complements these efforts by focusing on a new dimension, MLLMs' comparative reasoning capacity on a pair of visually or semantically relevant images.

Multi-image datasets. Several existing datasets [199, 242, 68, 114, 297, 117] provide multi-image data (*e.g.*, pairs of images), but they serve different purposes (*e.g.*, not for evaluating MLLMs) or have relatively limited scopes. NLVR2 [242] labels each image pair with a caption that may or may not be relevant to the images, asking models to predict the caption's relevance (*i.e.*, image-text matching). A few datasets [117, 28, 296] synthesize multi-image data for instruction tuning (*e.g.*, image editing). More relevant to ours are [68, 114, 297, 199]. Birds-to-Words [68] aims to describe the difference between two birds; Sopt-the-diff [114] focuses on the difference between two outdoor scenes; Q-bench2 [297] compares the quality (*e.g.*, blurriness) between two images; Relative Attributes [199] compares the relativeness of attributes between two facial or natural images. However, these datasets have limited scopes, only targeting specific domains or questions. In contrast, our COMPBENCH defines eight relative comparisons, covering a wide range of relativities in the real world. Our image pairs are curated from fourteen diverse visual domains. We believe this offers the V&L community a more comprehensive benchmark to assess the comparative capabilities of current leading MLLMs.

Learning to rank & learning with preference. Several research topics are relevant to ours and may benefit from our COMPBENCH. Learning to rank (LTR) [152, 172, 33] aims to realize a scoring function that can rank examples (*e.g.*, images) based on certain aspects, such as facial ages [193, 35] and degrees of attributes' presence [199]. Typically, an LTR model takes one example as input; the model is trained with pairs of examples such that the output scores match the ground-truth orders. Recently, learning with preference information [70] has become a mainstream approach to fine-tuning LLMs for alignment [208, 51]. Unlike our focus, these works usually collect pairs of outputs (*e.g.*, answers to a question) with humans' preferences to supervise model fine-tuning.

4.3 Why Do We Study Comparative Reasoning?

To date, most of the existing visual recognition and V&L benchmarks focus on a single visual input (*e.g.*, an image or a video clip), aiming to assess and promote *absolute* inference and reasoning within it, for example, identifying objects, recognizing their properties/states/actions, and describing and reasoning about their interactions within in the scene.

In reality, not all the inference and reasoning could be made absolute, or need to be absolute. For example, it is hard and ambiguous to describe the absolute degree of smiling [199], but it is relatively easy to compare two faces and tell which one smiles more. This fact applies to other visual properties like attributes (*e.g.*, length), states (*e.g.*, steps in cooking), and spatial locations (*e.g.*, longitude and latitude). Often, comprehending the *relativity* is sufficient for us to make sense of the real world.

Furthermore, learning to infer and reason about *relativity* could naturally and more efficiently facilitate AI models to grasp *fine-grained* details. For instance, learning to describe a complex scene (*e.g.*, captioning) often results in a model mastering common objects and properties but missing rare and subtle ones. In contrast, learning to tell the difference between two scenes promotes the model to identify subtle changes and describe them.

Last but not least, the ability to perform comparative reasoning is integral to our daily decision-making and problem-solving (see §4.1 for some examples). Humans' comparative capability, *e.g.*, providing preferences between instances, has also been widely leveraged to supervise foundation models like LLMs to align their outputs with application requirements and societal expectations [208, 51]. We thus believe it is crucial to assess and promote comparative reasoning about relativity in AGI.

4.4 COMPBENCH Benchmark

We introduce COMPBENCH, a multimodal benchmark designed to assess the comparative reasoning abilities of MLLMs across various dimensions. In what follows, we first describe the types of comparative capabilities that COMPBENCH aims to evaluate (§4.4.1). Next, we outline our methodology for collecting images, followed by how we annotate

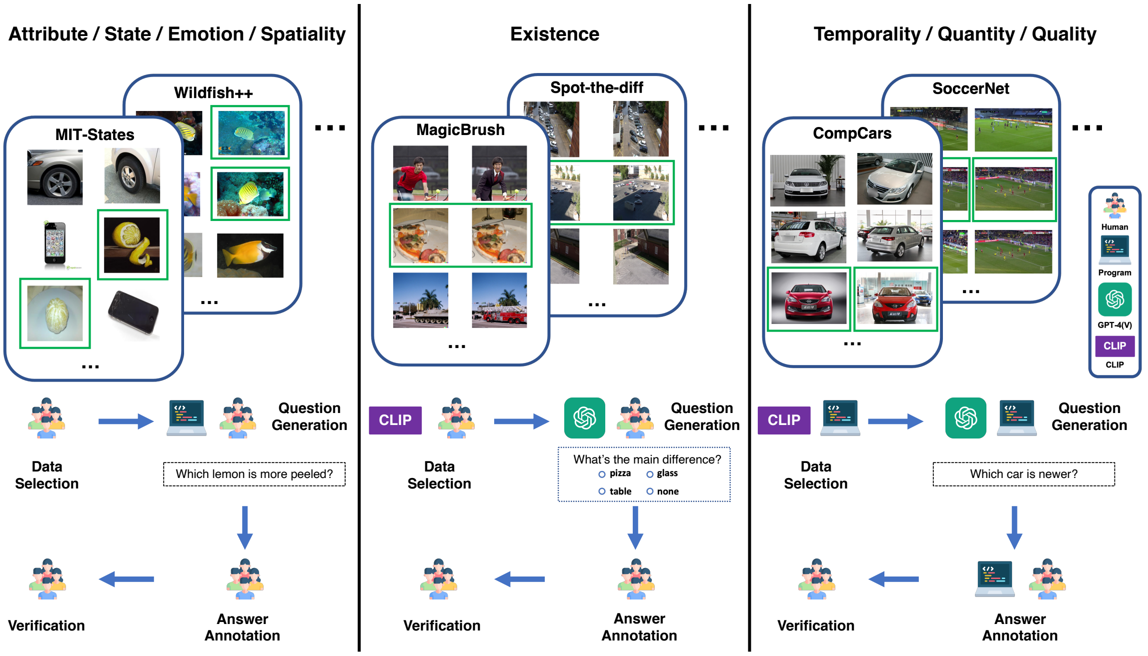


Figure 4.2: **CompBench curation pipeline**, including data selection, question generation, answer annotation, and verification. We rely on combinations of humans, computer programs, MLLMs (specifically GPT-4V [2]), and CLIP similarity [207] to select images and generate questions, based on relativity types and available metadata.

associated questions and answers to evaluate these capabilities (§4.4.2). Lastly, we provide detailed statistics on CompBench and discuss its data quality (§4.4.3). Figure 4.2 illustrates the overall pipeline used to develop CompBench.

4.4.1 Types of Relativity

Building upon §4.3, we consider eight comparison categories to evaluate MLLMs' abilities to discern differences between two similar images (Figure 4.1).

(1) Visual Attribute focuses on five common visual properties — Size, Color, Texture, Shape, and Pattern — and tests whether the model can identify the relative magnitude of these attributes between images. **(2) Existence** assesses the model's capacity to identify fine-grained variations by detecting subtle changes between images. **(3) State** involves comparing the conditions or status of objects. **(4) Emotion** assesses the model's capability to interpret degrees of human emotions. **(5) Temporality** and **(6) Spatiality** evaluate the model's ability to recognize differences in images caused by temporal or spatial differences. These categories require both commonsense and comprehension of the physical world. Lastly, **(7) Quantity** measures the relative counting skills, and **(8) Quality** compares the quality of two images, examining the model's low-level visual perceptual skills.

4.4.2 Dataset Curation

One major challenge in constructing CompBench is mining image pairs that reflect the aforementioned relativities. Fortunately, many publicly accessible datasets in vision and V&L offer detailed annotations and metadata. We carefully investigate these datasets and identify a *seed set* of fourteen datasets that align with the eight relativity types (§4.4.1), covering a wide range of domains like open-domain, fashion, animal, sports, automotive, facial, and both outdoor and indoor scenes (cf. Right in Table 4.1). Below, we outline the datasets for each relativity type and the process for generating triplets of image pairs, a question, and an answer. *Please see the supplementary material for details.*

4.4.2.1 Visual Attribute

Data collection. We consider five visual attribute datasets. **MIT-States [111]** includes 245 objects with 115 visual attributes, from online sources such as food or device websites. **Fashionpedia [115]** is tailored to clothing and accessories and contains 27 types of apparel along with 294 detailed attributes. **VAW [205]**, similar to MIT-States, offers a large-scale collection of 620 unique attributes, including color, shape, and texture. **CUB-200-2011 [261]** and **Wildfish++ [205]** specifically provide attributes for birds and fish. The former catalogs 15 bird parts and their attributes (e.g., "notched tail"); the latter details 22 characteristics (e.g., “yellow pelvic fins”) of various fish species. For each dataset, we cluster images by objects or parts with the same attributes (*e.g.*, “round table”, “asymmetrical blouse”, “curved bill”, “yellow dorsal fin”) and extract visually similar image pairs from each group.

Annotation. We apply rule-based approaches to generate questions about relative degrees of attributes between objects (*e.g.*, “Which coat is more floral?”). We then pair the questions with the corresponding image pairs and present them to six human annotators. The annotators are tasked with labeling the correct answers (binary: left/right) and filtering out any irrelevant or nonsensical questions about the images. In total, we construct a collection of **5.3K triplets**.

4.4.2.2 Existence

Data collection. We consider datasets for image editing, which provide image pairs with similar layouts but subtle changes. We adopt **MagicBrush [296]**, a recently released dataset for instruction-guided editing. It consists of (source image, instruction, target image) triplets, where the instruction specifies a subtle change between the source and target images. We also consider **Spot-the-diff [114]**, which provides image pairs in outdoor scenes, along with descriptions of their differences.

Annotation. We curate *multiple-choice* questions to ease automatic evaluation. We prompt GPT-4V [2] with in-context learning to generate questions; the options are formed by the extracted objects and their attributes from images. We then pass the questions (along with image pairs) to the annotators to verify the options and label the correct ones. In total, we curate **2.2K triplets**.

4.4.2.3 State

Data collection. We explore vision datasets covering the condition or status of objects (*e.g.*, “pureed tomato” or “mashed potatoes”). Specifically, we use two large-scale, open-domain visual attribute datasets: **MIT-States [111]** and **VAW [205]**. They annotate not only the five common visual properties used in **Visual Attribute** but also some other properties about object states. We ask human annotators to manually review the datasets to identify image pairs relevant to state attributes.

Annotation. We follow the annotation protocol in §4.4.2.1 to curate a total of **1.1K triplets**.

4.4.2.4 Emotion

Data collection. We gather facial images from two publicly available datasets, **CelebA [175]** and **FER-2013 [80]**, focusing on eight annotated human emotional states: smiling, angry, disgusted, fearful, happy, neutral, sad, and surprised. We form image pairs from the same emotional state.

Annotation. We follow the annotation protocol in §4.4.2.1 to curate a total of **5.3K triplets**.

4.4.2.5 Temporality

Data collection. We consider images with time-related tags. One pertinent source is videos. Specifically, we use **SoccerNet [77]**, a dataset for soccer video understanding. It annotates various soccer actions (*e.g.*, free-kicks, corner-kicks, etc.) and specifies their exact periods (start-end frame indices). Using this temporal metadata, we extract two frames from each annotated action, creating an image pair that allows temporal comparison. We also consider **CompCars [283]**, a dataset designed for fine-grained categorization of vehicles. This dataset offers a detailed ontology of car attributes, such as make, model, and year. We generate image pairs that feature the same car model from different production years, for instance, a 2017 Honda Civic vs. its 2015 counterpart.

Annotation. We automatically generate (rule-based) questions and answers about which frame or object is associated with an earlier/later time-related tag, for example, “Which frame happened first during the free-kick?” To ensure that the two images are relevant enough to offer sufficient temporal cues, we compute the CLIP visual similarity [207], selecting only image pairs with similar layouts and object poses. In total, we curate **13.3K triplets**.

4.4.2.6 Spatiality

Data collection. We collect images with spatial tags, *e.g.*, object locations. Specifically, we use **NYU-Depth V2** [236], featuring indoor scenes with object segments and depths. Using the segmentation maps, we identify objects within each image, and group images containing the same objects.

Annotation. We follow the annotation protocol in §4.4.2.1, leveraging pre-defined templates and object information to generate questions about spatial relative comparisons (*e.g.*, “Which shelf is closer to the camera?”), followed by human answer annotation. Overall, we curate **1.9K triplets**.

4.4.2.7 Quantity

Data collection. We consider images with labels related to object instances. One prominent source is object detection datasets. Here, we use **VQAv2 [81]**, which is built upon MSCOCO [46] and encompasses a variety of question types, such as object counting and color. We focus on the counting questions, grouping images with similar questions and sampling image pairs within each group.

Annotation. We use GPT-4 [2] to convert original absolute counting questions (*e.g.*, "How many elephants are there?") to relative counting questions (*e.g.*, "Which image has more elements?"). The answers are derived automatically from VQAv2's ground-truth answers. We curate **9.8K triplets**.

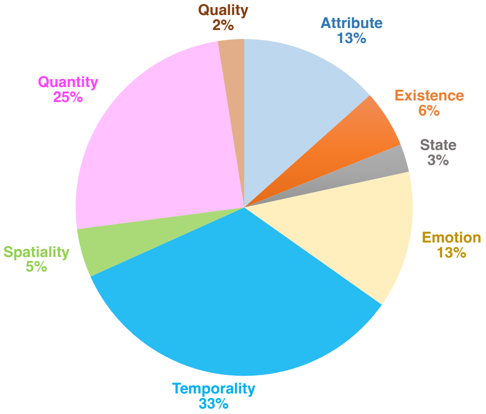
4.4.2.8 Quality

Data collection. We use **Q-bench2 [297]**, a recently introduced dataset to evaluate low-level visual perception. Concretely, it challenges MLLMs to determine the quality (*e.g.*, blurriness or distortion) of a single image or to compare the quality between two images.

Annotation. Through a meticulous filtering process (cf. §4.4.2.1), we select paired images from Q-bench2, along with the annotated multiple-choice questions and answers, resulting in **1K triplets**.

4.4.3 Quality Control and Dataset Statistics

To ensure the integrity of COMPBENCH, we ask annotators to exclude poor-quality examples, such as those with low-resolution images or questions that are irrelevant or nonsensical about the images. The annotators also filter out image pairs with ambiguous relativities, for example, image pairs with indistinguishable smiling degrees. To faithfully assess fine-grained capabilities, we also apply the CLIP visual similarity to **Existence**, removing image pairs with salient differences. Additionally, we implement a rigorous cross-verification process, where each annotator confirms the accuracy of others' answers. Only samples that receive unanimous approval from annotators are kept. Consequently, our COMPBENCH benchmark comprises **39.8K** diverse triplets (eight relativities from fourteen visual domains) with high quality and reliability. Please see Table 4.1 for the statistics.



Relativity	Dataset	Domain	# our samples
Attribute	MIT-States [111]	Open	0.2K
	Fashionpedia [115]	Fashion	2.4K
	VAW [205]	Open	0.9K
	CUB-200-2011 [261]	Bird	0.9K
	Wildfish++ [308]	Fish	0.9K
Existence	MagicBrush [296]	Open	0.9K
	Spot-the-diff [114]	Outdoor Scene	1.2K
State	MIT-States [111]	Open	0.6K
	VAW [205]	Open	0.5K
Emotion	CelebA [175]	Face	1.5K
	FER-2013 [80]	Face	3.8K
Temporality	SoccerNet [77]	Sport	8.3K
	CompCars [283]	Car	5K
Spatiality	NYU-Depth V2 [236]	Indoor Scene	1.9K
Quantity	VQAv2 [81]	Open	9.8K
Quality	Q-Bench2 [297]	Open	1K
Total	-	-	39.8K

Table 4.1: **Overall statistics of CompBench.**

4.5 Experiments

4.5.1 Experimental Setup

Baselines. We use our CompBench to evaluate several leading MLLMs. This includes two powerful proprietary models, GPT-4V(ision) [2] and Gemini1.0-Pro[3] [251], and two open-source alternatives, LLaVA-1.6 [169] and VILA-1.5 [163]. GPT-4V(ision) and Gemini excel in various vision and language tasks, such as VQA [81], OCR interpretation [186], spatial reasoning [184], and college-level subject knowledge [288]. LLaVA-1.6 and VILA-1.5 also demonstrate competitive performance against these proprietary giants on some tasks. Our focus is to investigate whether these cutting-edge models can extend their capabilities to the realm of multi-image relative comparison. We evaluate proprietary models via their

[3] Due to limited public testing quota available for Gemini-1.5 during our study, we opted for Gemini-1.0 Pro.

official APIs and open-source models using (or fine-tuning on) NVIDIA RTX 6000 Ada GPUs. For more details, please refer to the supplementary material.

Evaluation tasks & metrics. We divide our COMPBENCH into a test split (31.8K) and a held-out split (7.9K), using an 80:20 ratio. The latter is reserved for future developments (*e.g.*, prompt engineering). By default, we concatenate the image pairs horizontally (i.e., left and right) as the visual input to MLLMs, and prompt MLLMs to answer questions about the relativity between these images. To facilitate automated evaluation, we include the possible answers as options in the questions. For **Existence** and **Quality**, there are multiple options (typically more than two). For **Quantity**, there are three options: left/right/same. For other types, there are binary options: left/right. We employ the standard accuracy as our evaluation metric. A question is answered correctly if the model prediction exactly matches the ground-truth answer. Further details are included in the supplementary material.

4.5.2 Main Results (Table 4.2)

Overall challenges in COMPBENCH. We observe that current MLLMs face challenges in answering relative questions in COMPBENCH (see Table 4.2). All MLLMs achieve averaged accuracies over the sixteen tasks (columns) below 80%, with GPT-4V reaching the highest accuracy at 74.7%. Further, a human evaluation study on a subset of our examples indicates that GPT-4V's performance remains notably behind human capabilities, highlighting the need for substantial improvement (Table 4.4).

Superiority in State & Emotion. State relativity is an area where MLLMs demonstrate strength. For instance, GPT-4V/LLaVA-1.6 achieve 92.2%/89.7%, respectively, on MIT-states [111] for state relativity. Similarly, they demonstrate impressive performance in emotion relativity (91.8%/96.2% on CelebA [175]). Our preliminary analysis suggests that

Model	Attribute					Exist.		State		Emot.		Temp.		Spat.	Quan.	Qual.	Avg
	ST	FA	VA	CU	WF	MB	SD	ST	VA	CE	FE	SN	CC	ND	VQ	QB	
GPT-4V	**91.8**	**89.0**	76.9	71.4	**72.1**	**58.3**	41.9	**92.2**	**87.8**	91.8	83.4	**71.4**	**73.7**	56.1	**63.8**	**73.0**	**74.7**
Gemini1.0-Pro	71.9	76.3	69.3	59.9	54.9	53.7	**53.0**	81.8	70.7	60.6	71.2	55.1	58.2	56.6	54.6	59.5	63.0
LLaVA-1.6	84.9	72.1	**77.7**	**72.6**	68.7	26.5	20.7	89.7	79.3	**96.2**	**83.5**	51.0	50.2	**67.2**	50.1	64.8	66.0
VILA-1.5	69.9	66.2	70.9	55.9	52.0	49.5	36.8	71.9	74.5	57.1	55.6	51.1	52.9	51.8	47.7	64.8	58.0
Chance level	50.0	50.0	50.0	50.0	50.0	8.6	9.7	50.0	50.0	50.0	50.0	50.0	50.0	50.0	33.3	37.4	43.1

Table 4.2: **Overall results on CompBench test split.** We evaluate four leading MLLMs across eight relative comparisons spanning sixteen tasks. The top-performing model in each task is indicated **in bold**. ST: MIT-States [111], FA: Fashionpedia [115], VA: VAW [205], CU: CUB-200-2011 [261], WF: Wildfish++ [308], MB: MagicBrush [296], SD: Spot-the-diff [114], CE: CelebA [175], FE: FER-2013 [80], SN: SoccerNet [77], CC: CompCars [283], ND: NYU-Depth V2 [236], VQ: VQAv2 [81], QB: Q-Bench2 [297].

their capacity to determine the degree of emotion (*e.g.*, smiling) relies on specific facial features such as lip curvature or visible teeth.

Challenges in Existence. All MLLMs show weak performance in existence relativity tasks. We attribute this to the multiple capabilities these tasks demand, including spatial understanding and precise object recognition/comparison. For instance, when an object in the left image is moved to a different location in the right image, the models need to not only recognize the same object in the right image but also understand the relative change in its position. This necessitates both robust object recognition and accurate spatial reasoning. Given that an image can contain numerous objects, the model should have a deep understanding of how the existence of them changes between images.

Challenges in Temporality and Spatiality. MLLMs encounter difficulties with both temporal relativity, which requires commonsense, and spatial relativity, which demands comprehension of depth perception between objects. Specifically, for the spatial task, all

MLLMs perform below 70%, and notably, both proprietary models, GPT-4V and Gemini1.0-Pro, only achieve slightly above chance levels (56.1% and 56.6%, respectively). This underscores the need for further research in improving spatial relativity to advance models towards artificial general intelligence (AGI).

Challenges in Quantity & Quality. We observe the mediocre performance of MLLMs in quantity relativity (*e.g.*, GPT-4V: 63.8%, VILA-1.5: 47.7%). We attribute this to the models' weak capability in accurately counting objects in images. Similarly, MLLMs struggle with assessing image quality (*e.g.*, 73.0% of GPT-4V's accuracy). These capabilities are crucial for making informed decisions in our daily lives (cf. §4.1), highlighting the need for MLLMs to improve in these aspects.

Variability in performance across domains. The performance of MLLMs varies in different domains. For instance, they excel at comparing visual attributes of daily objects [111] and clothing [115] while struggling with those of animals (*e.g.*, birds [261], fish [308]). This could be due to the complexity of animal features, such as feathers, scales, or markings, which are more challenging for the model to interpret compared to simpler attributes in everyday objects.

4.5.3 Further Analyses

Two-stage reasoning. What if we first ask MLLMs to analyze each image in a pair separately (*e.g.*, "How far is the table from the camera that took this photo? Return a number in feet.") and use their language responses to answer a follow-up pure language question (*e.g.*, "Based on the responses, which object is closer to the camera?")? We evaluate this two-stage reasoning approach on three comparison tasks: Existence, Emotion, and Spatiality. We find that GPT-4V, using this two-stage reasoning, performs less effectively on all three

Model	Exist.	Emot.	Spat.
	MB	CE	ND
GPT-4V	58.3	91.8	56.1
GPT-4V$_{\text{two-stage}}$	45.9	90.3	36.3

Model	Temp.	Quan.
	SN	VQ
LLaVA-1.6	51.0	50.1
LLaVA-1.6$_{\text{fine-tuned}}$	93.9	56.6

Table 4.3: **Left: Two-stage reasoning.** Analyzing images separately and then comparing them via a pure language question reduces performance, due to challenges in absolute inference and reasoning. **Right: Fine-tuning results.** Fine-tuned LLaVA-1.6 excels in temporal relativity but falls short in quantity, struggling with counting.

tasks (Left in Table 4.3). This is likely because analyzing images separately can sometimes be more challenging than comparing images directly. For instance, calculating the exact distance from an object to the camera may be difficult, leading to inaccurate numbers. In contrast, directly answering a question, "Which object is closer to the camera?" may be easier, as models only need to determine the relative closeness between objects.

Fine-tuning experiments. We conduct a study to see if fine-tuning helps improve the comparative capabilities of MLLMs. We focus on Temporality and Quantity and fine-tune LLaVA-1.6 separately for each task. Concretely, following LLaVA's paper [169], we only train the bridge and the language modules while keeping the vision encoder frozen. As shown in Table 4.3 (Right), fine-tuning significantly benefits LLaVA-1.6 in the temporal task (SoccerNet). However, interestingly, it only marginal gains in quantity questions. We attribute this to its vision encoder, CLIP [207], which may have weak capabilities in counting the number of objects, as reported by several prior works [207, 197, 204]. This suggests considering new architectures or training strategies to improve its counting capabilities as future work. Please see the supplementary material for further details.

Error Analysis. We analyze error cases by GPT-4V and offer insights to enhance its performance (Figure 4.3). **First**, GPT-4V may not effectively distinguish the color between

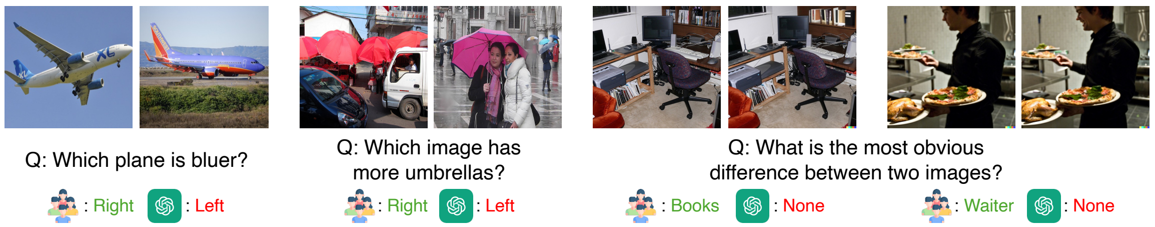

Q: Which plane is bluer?
: Right
: Left
Q: Which image has more umbrellas?
: Right
: Left
Q: What is the most obvious difference between two images?
: Books
: None
: Waiter
: None

Figure 4.3: **Error Analysis on CompBench.** We observe four types of errors where GPT-4V [2] falls short: (i) differentiating colors between objects and backgrounds, (ii) counting small or distant objects, (iii) identifying objects within crowded scenes, and (iv) recognizing out-of-focus details.

Model	Accuracy
GPT-4V	68.6%
Humans	86.5%

Table 4.4: **Preliminary human evaluation** on 140 samples.

objects and backgrounds. For instance, in the first example of Figure 4.3, the object — a plane — shares a similar color (i.e., blue) with the background, causing GPT-4V to fail in selecting the bluer plane. **Second**, GPT-4V struggles to count accurately for small or distant objects (*e.g.*, people further away wearing umbrellas), as shown in the second example. **Third**, GPT-4V finds it challenging to identify the target object if numerous items exist within images. In the third example, both images contain multiple objects, such as monitors, laptops, keyboards, desks, and books, and GPT-4V fails to pinpoint the target object (*i.e.*, books). **Lastly**, GPT-4V may overlook details in out-of-focus areas of images. For instance, in the fourth example, the camera focuses on a pizza, leaving a waiter out of focus. Consequently, GPT-4V fails to detect facial changes in the waiter, highlighting its struggle with details in out-of-focus areas.

Human evaluation. We investigate how much current MLLMs (*e.g.*, GPT-4V [2]) lag behind human performance. We conduct a preliminary human evaluation using 140 examples randomly sampled from the sixteen tasks (columns) in Table 4.1. We ask five human evaluators, different from our annotators, to answer these questions and average their performance. As shown in Table 4.4, the performance of GPT-4V on these examples is approximately 18% below that of humans. This not only highlights the challenge of our CompBench but also underscores the limited capabilities of current MLLMs in multi-image relative comparison.

4.6 Summary

We introduce CompBench, a comprehensive benchmark designed to evaluate comparative reasoning in multimodal LLMs (MLLMs). CompBench offers extensive coverage of eight relative comparisons between pairs of images drawn from fourteen diverse domains. CompBench evaluates recent MLLMs, offering detailed analyses and insights for future advancements.

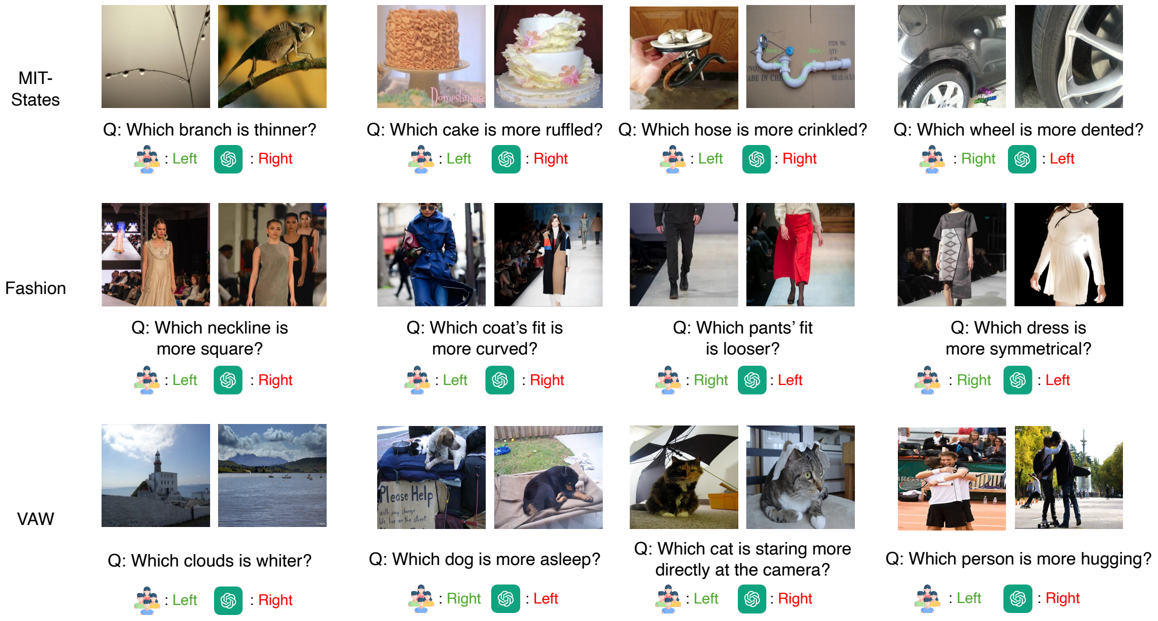

MIT-States
Q: Which branch is thinner?
: Left : Right
Q: Which cake is more ruffled?
: Left : Right
Q: Which hose is more crinkled?
: Left : Right
Q: Which wheel is more dented?
: Right : Left
Fashion
Q: Which neckline is more square?
: Left : Right
Q: Which coat's fit is more curved?
: Left : Right
Q: Which pants' fit is looser?
: Right : Left
Q: Which dress is more symmetrical?
: Right : Left
VAW
Q: Which clouds is whiter?
: Left : Right
Q: Which dog is more asleep?
: Right : Left
Q: Which cat is staring more directly at the camera?
: Left : Right
Q: Which person is more hugging?
: Left : Right

Figure 4.4: **Qualtiative examples on MIT-States [111], Fashionpedia [115], and VAW [205].**

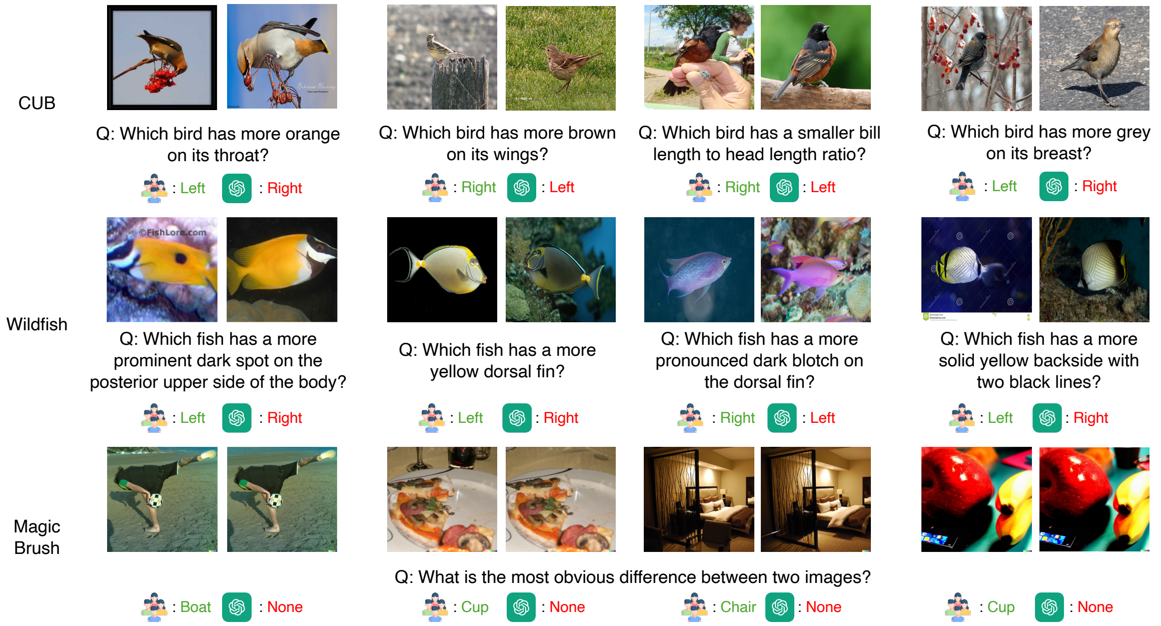

CUB
Q: Which bird has more orange on its throat?
: Left : Right
Q: Which bird has more brown on its wings?
: Right : Left
Q: Which bird has a smaller bill length to head length ratio?
: Right : Left
Q: Which bird has more grey on its breast?
: Left : Right
Wildfish
Q: Which fish has a more prominent dark spot on the posterior upper side of the body?
: Left : Right
Q: Which fish has a more yellow dorsal fin?
: Left : Right
Q: Which fish has a more pronounced dark blotch on the dorsal fin?
: Right : Left
Q: Which fish has a more solid yellow backside with two black lines?
: Left : Right
Magic Brush
Q: What is the most obvious difference between two images?
: Boat : None
: Cup : None
: Chair : None
: Cup : None

Figure 4.5: **Qualtiative examples on CUB-200-2011 [261], Wildfish++ [308], and MagicBrush [296].**

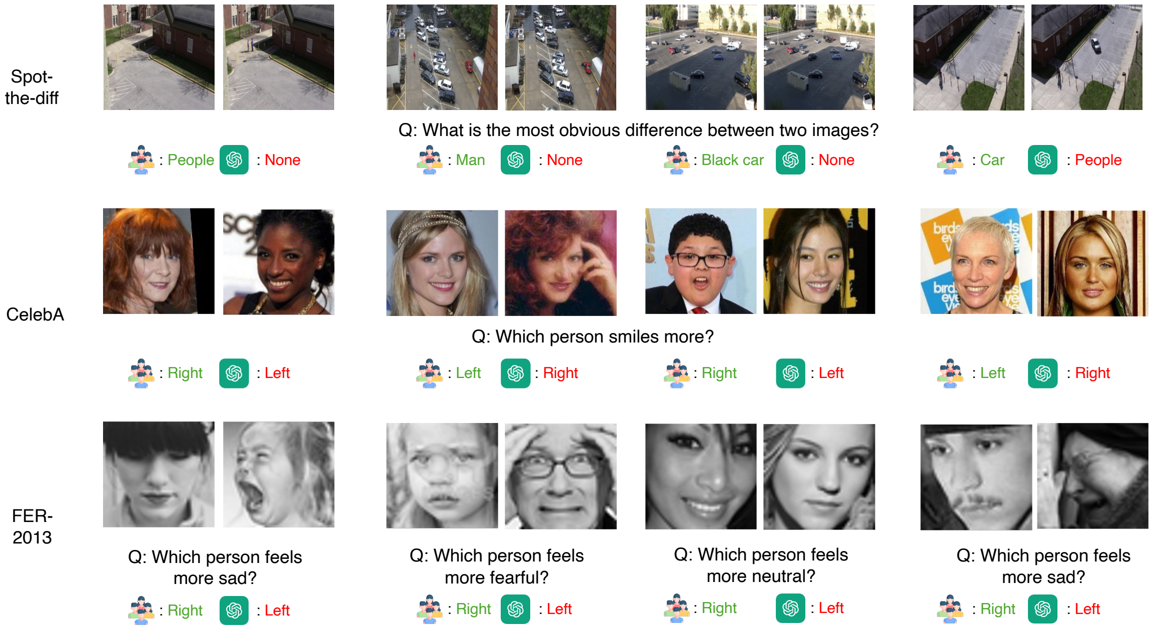

Spot-the-diff
Q: What is the most obvious difference between two images?
: People : None
: Man : None
: Black car : None
: Car : People
CelebA
Q: Which person smiles more?
: Right : Left
: Left : Right
: Right : Left
: Left : Right
FER-2013
Q: Which person feels more sad?
: Right : Left
Q: Which person feels more fearful?
: Right : Left
Q: Which person feels more neutral?
: Right : Left
Q: Which person feels more sad?
: Right : Left

Figure 4.6: **Qualtiative examples on Spot-the-diff [114], CelebA [175], and FER-2013 [80].**

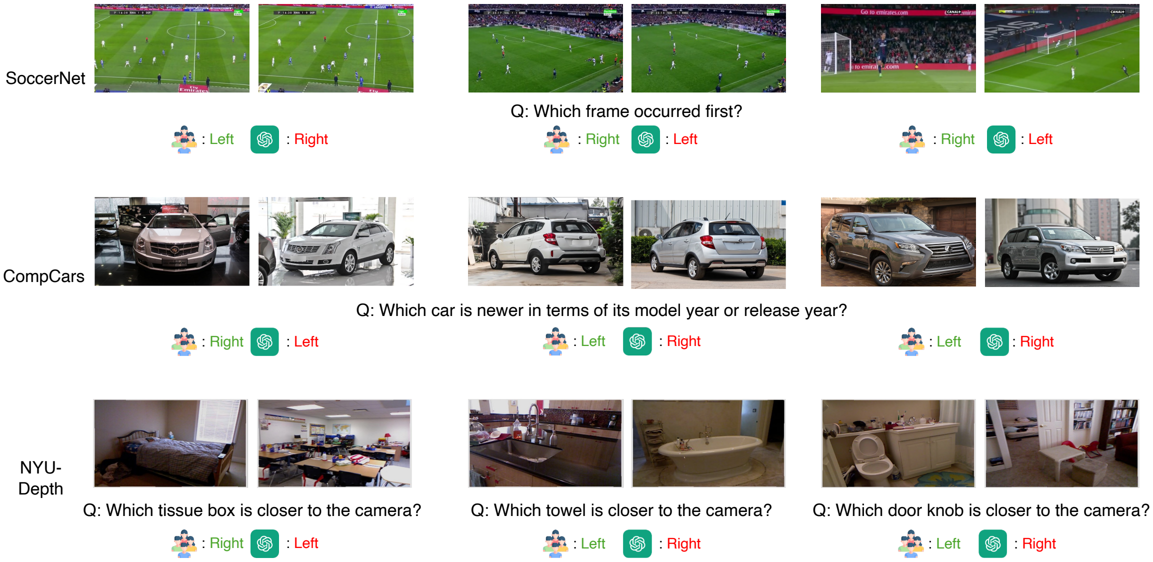

SoccerNet
Q: Which frame occurred first?
: Left : Right
: Right : Left
: Right : Left
CompCars
Q: Which car is newer in terms of its model year or release year?
: Right : Left
: Left : Right
: Left : Right
NYU-Depth
Q: Which tissue box is closer to the camera?
: Right : Left
Q: Which towel is closer to the camera?
: Left : Right
Q: Which door knob is closer to the camera?
: Left : Right

Figure 4.7: **Qualtiative examples on SoccerNet [77], CompCars [283], and NYU-Depth V2 [236].**

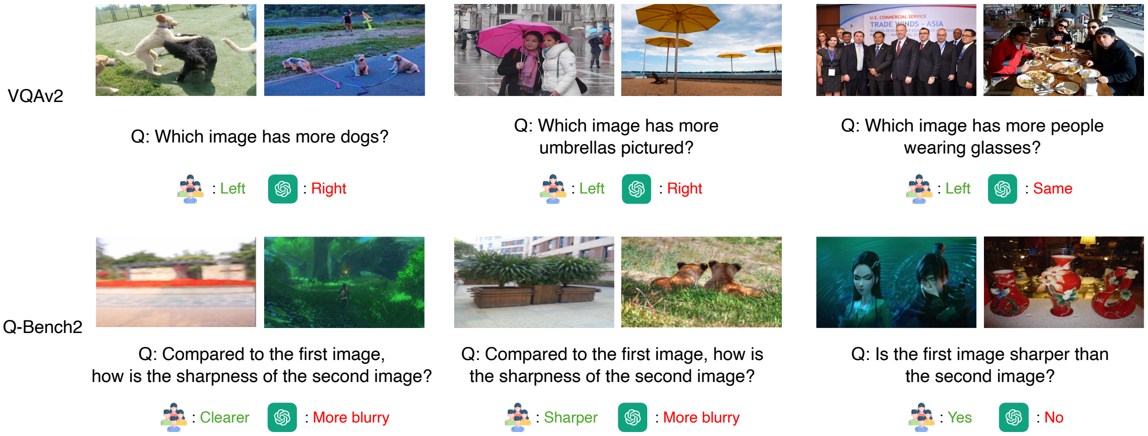

VQAv2
Q: Which image has more dogs?
: Left
: Right
Q: Which image has more umbrellas pictured?
: Left
: Right
Q: Which image has more people wearing glasses?
: Left
: Same
Q-Bench2
Q: Compared to the first image, how is the sharpness of the second image?
: Clearer
: More blurry
Q: Compared to the first image, how is the sharpness of the second image?
: Sharper
: More blurry
Q: Is the first image sharper than the second image?
: Yes
: No

Figure 4.8: **Qualtiative examples on VQAv2 [81] and Q-Bench2 [297].**

Part IV: V&L Data Representations

Chapter 5: Aligning semantic representations with visual features

In Part III, we discussed how to curate data effectively for V&L. Now that we have data, we need to encode it into **representations** (or embeddings) to train the models. In V&L, aligning visual and linguistic representations is crucial for effective model training.

In this chapter, we explore new semantic representations for zero-shot image classification, which align closely with the visual features of images. Concretely, we revisit the use of documents as semantic representations. The documents, such as Wikipedia pages, contain rich visual information, which, however, can easily be buried by the vast amount of non-visual sentences. We thus propose a semi-automatic mechanism for visual sentence extraction that leverages the document section headers and the clustering structure of visual sentences. The extracted visual sentences essentially form semantic representations like visual attributes but need much less human effort. On the ImageNet dataset with over 10,000 unseen classes, our new representations lead to a 64% relative improvement against the commonly used ones, demonstrating their superior alignment with visual features.

5.1 Introduction

Algorithms for visual recognition usually require hundreds of labeled images to learn how to classify an object [94]. In reality, however, the frequency of observing an object follows a long-tailed distribution [305]: many objects do not appear frequently enough

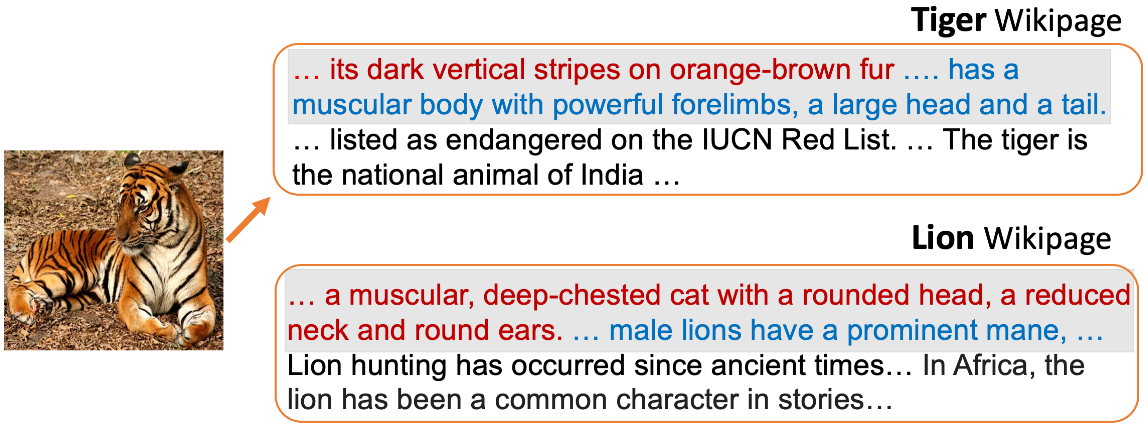

Tiger Wikipage
… its dark vertical stripes on orange-brown fur …. has a muscular body with powerful forelimbs, a large head and a tail.
… listed as endangered on the IUCN Red List. … The tiger is the national animal of India …
Lion Wikipage
… a muscular, deep-chested cat with a rounded head, a reduced neck and round ears. … male lions have a prominent mane, …
Lion hunting has occurred since ancient times… In Africa, the lion has been a common character in stories…

Figure 5.1: An illustration of our ZSL approach, which recognizes the input image by comparing it to the visual sentences of documents. Here we show two documents, one for "Tiger" and one for "Lion". The gray area highlights the extracted visual sentences (red: by section headers; blue: by clustering).

for us to collect sufficient images. Zero-shot learning (ZSL) [142], which aims to build classifiers for unseen object classes using their *semantic representations*, has thus emerged as a promising paradigm for recognizing a large number of classes.

Being the only information of unseen objects, how well the semantic representations describe the visual appearances plays a crucial role in ZSL. One popular choice is *visual attributes* [142, 201, 262] carefully annotated by humans. For example, the bird "Red bellied Woodpecker" has the "capped head pattern" and "pointed wing shape". While strictly tied to visual appearances, visual attributes are laborious to collect, limiting their applicability to small-scale problems with hundreds of classes.

For large-scale problems like ImageNet [59] that has more than $20,000$ classes, existing ZSL algorithms [69, 194] mostly resort to *word vectors* of classes names [188, 203] that are automatically extracted from large corpora like Common Crawl. While almost labor free, word vectors are purely text-driven and barely aligned with visual information. As a result, the state-of-the-art ZSL accuracy on ImageNet falls far behind being practical [37].

Is it possible to develop semantic representations that are as powerful as visual attributes without significant human effort? A feasibility study by representing a class with its Wikipedia page shows some positive signs — Wikipedia pages do capture rich attribute information. For example, the page "Red-bellied Woodpecker" contains phrases "red cap going from the bill to the nape" and "black and white barred patterns on their back, wings and tail" that exactly match the visual attributes mentioned above. In other words, if we can identify *visual* sentences from a document to represent a class, we are likely to attain much higher ZSL accuracy[4].

To this end, we present a simple yet effective semi-automatic approach for *visual sentence extraction*, which leverages two informative semantic cues. First, we leverage the *section structures* of Wikipedia pages: the section header indicates what kind of sentences (visual or not) appear in the section. Concretely, we search Wikipedia pages of common objects following the synsets in ImageNet (*e.g.*, fish, room), and manually identify sections that contain visual information (*e.g.*, characteristics, appearance). We then apply these visual headers to the Wikipedia pages of the remaining ImageNet classes. Second, we observe that visual sentences share some common contextual patterns: for example, they contain commonly used words or phrases of visual attributes (*e.g.*, red color, furry surface). To leverage these patterns, we perform K-means sentence clustering using the BERT features [62] and manually select clusters that contain visual information. We keep sentences in these clusters and combine them with those selected by section headers to represent a document. See Figure 5.1 for an illustration.

[4] Representing a class by a document has been studied in [306, 66, 206], but they use all sentences instead of extracting the visual ones.

To further increase the discriminative ability of the visual sentences between similar object classes (*e.g.*, breeds of dogs), we introduce a novel scheme to assign weights to sentences, emphasizing those that are more representative for each class.

We validate our approach on three datasets: ImageNet Fall 2011 dataset [59], which contains $14,840$ unseen classes with Wikipedia pages; Animals with Attributes 2 (AwA2) [277], which has 50 animal classes; Attribute Pascal and Yahoo (aPY) [67], which has 32 classes. Our results are promising: compared to word vectors on ImageNet, we improve by 64% using visual sentences. On AwA2 and aPY, compared to visual attributes annotated by humans, we improve by 8% and 5%, respectively.

5.2 Comparsion to existing works

Semantic representations. Visual attributes are the most popular semantic representations [142, 201, 262, 298]. However, due to the need of human annotation, the largest dataset has only 717 classes. [214, 213] collect visual sentences for each image, which is not scalable. For large-scale recognition, word vectors [188] have been widely used. [182, 124, 268] explore the use of WordNet hierarchy [189], which may not be available in other applications.

Similar to ours, [9, 66, 206, 306] represent classes by documents, by counting word frequencies but not extracting visual sentences. [11] extract *single* word attributes, which are not discriminative enough (*e.g.*, “red cap” becomes “red”, “cap”). None of them works on ZSL with over 1,000 classes.

[98, 144] collect images and tags of a class and derives its semantic representation from tags, which is not feasible for unseen classes on ZSL.

Zero-shot learning algorithms. The most popular way is to learn an embedding space in which visual features and semantic representations are aligned and nearest neighbor classifiers can be applied [38, 221, 8, 137, 223, 302, 280, 238]. These algorithms consistently improve accuracy on datasets with attributes. Their accuracy on ImageNet, however, is saturated, mainly due to the poor quality of semantic representations [37].

5.3 *Visual* Sentence Extraction

5.3.1 Background and notation

ZSL algorithms learn to align visual features and semantic representations using a set of *seen* classes S. The alignment is then applied to the test images of unseen classes U. We denote by $D = \{(\boldsymbol{x}_n, y_n \in S)\}_{n=1}^{N}$ the training data (*i.e.*, image feature and label pairs) with the labels coming from S.

Suppose that we have access to a semantic representation $\boldsymbol{a}_c$ (*e.g.*, word vectors) for each class $c \in S \cup U$, one popular algorithm DeViSE [69] proposes the learning objective

$$\begin{aligned}\sum_n \sum_{c \neq y_n} \max\{0, \Delta - f_{\boldsymbol{\theta}}^{\top}(\boldsymbol{x}_n) \boldsymbol{M} g_{\boldsymbol{\phi}}(\boldsymbol{a}_{y_n}) \\ + f_{\boldsymbol{\theta}}^{\top}(\boldsymbol{x}_n) \boldsymbol{M} g_{\boldsymbol{\phi}}(\boldsymbol{a}_c)\},\end{aligned} \tag{5.1}$$

where $\Delta \geq 0$ is a margin. That is, DeViSE tries to learn transformations $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\phi}}$ and a matrix $\boldsymbol{M}$ to maximize the visual and semantic alignment of the same classes while minimizing that between classes. We can then classify a test image $\boldsymbol{x}$ by

$$\arg\max_{c \in U} f_{\boldsymbol{\theta}}^{\top}(\boldsymbol{x}) \boldsymbol{M} g_{\boldsymbol{\phi}}(\boldsymbol{a}_c). \tag{5.2}$$

Here, we consider that every class $c \in S \cup U$ is provided with a document $H_c = \{\boldsymbol{h}_1^{(c)}, \cdots, \boldsymbol{h}_{|H_c|}^{(c)}\}$ rather than $\boldsymbol{a}_c$, where $|H_c|$ is the amount of sentences in document H_c and $\boldsymbol{h}_j^{(c)}$ is the jth sentence, encoded by BERT [62]. We mainly study DeViSE, but our approach can easily be applied to other ZSL algorithms.

Section headers
Characteristics, Description, Appearance, Habitat, Diet, Construction and Mechanics, Materials for utensil, Design for appliance, Furnishings for room, Fabrication, Feature for geological formation, Design, Equipment for sport
History, Health, Terminology, Mythology, Conservation, Culture, References, External links, Further reading

Table 5.1: Visual (top) & Non-Visual (bottom) sections.

5.3.2 Visual section selection

We aim to filter out sentences in H_c that are not describing visual information. We first leverage the section headers in Wikipedia pages, which indicate what types of sentences (visual or not) are in the sections. For example, the page “Lion” has sections “Description” and “Colour variation” that are likely for visual information, and “Health” and “Cultural significance” that are for non-visual information.

To efficiently identify these section headers, we use ImageNet synsets [59], which group objects into 16 broad categories. We randomly sample $30 \sim 35$ classes per group, resulting in a set of 500 classes. We then retrieve the corresponding Wikipedia pages by their names and manually identify section headers related to visual sentences. By sub-sampling classes in this way, we can quickly find section headers that are applicable to other classes within the same groups. Table 5.1 shows some visual/non-visual sections gathered from the 500 classes. For example, “Characteristics” frequently appears in pages of animals to describe their appearances. In contrast, sections like “History” or “Mythology” do not contain visual information. Investigating all the 500 Wikipedia pages carefully, we find 40 distinct visual sections. We also include the first paragraph of a Wikipedia page, which often contains visual information.

Sentence clusters
It has large ears that help the fox lower its body temperature.
It usually has a gray coat, with rusty tones, and a black tip to its tail.
It has distinct dark patches around the nose.
It is most recognisable for its dark vertical stripes on orangish-brown fur.
$\cdots$ muscular body with powerful forelimbs, a large head and a tail.
They have a mane-like heavy growth of fur around the neck and jaws $\cdots$
The kit fox is a socially monogamous species.
Male and female kit foxes usually establish monogamous mating $\cdots$
The average lifespan of a wild kit fox is 5.5 years.
Tiger mates all year round, but most cubs are born between March $\cdots$
The father generally takes no part in rearing.
The mortality rate of tiger cubs is about 50% in the first two years.

Table 5.2: Sentence clusters. The top cluster is *visual* and the bottom one is *non-visual*. The sentences from a class *kit-fox* are in red and those from a class *tiger* are in blue.

5.3.3 Visual cluster selection

Our second approach uses K-means for sentence clustering: visual sentences often share common words and phrases of visual attributes, naturally forming clusters. We represent each sentence using the BERT features [62], and perform K-means (with $K = 100$) over all the sentences from Wikipedia pages of ImageNet classes. We then manually check the 100 clusters and identify 40 visual clusters. Table 5.2 shows a visual (top) and a non-visual (bottom) cluster. We highlight sentences related to two classes: “kit-fox” (red) and “tiger” (blue). The visual cluster describes the animals’ general appearances, especially about visual attributes “dark”, “black”, “tail”, “large”, etc. In contrast, the non-visual cluster describes mating and lifespan that are not related to visual aspects.

5.3.4 Semantic representations of documents

After we obtain a filtered document $\hat{H}_c$, which contains sentences of the *visual* sections and clusters, the next step is to represent $\hat{H}_c$ by a vector $\boldsymbol{a}_c$ so that nearly all the ZSL algorithms can leverage it.

A simple way is **average**, $\bar{\boldsymbol{a}}_c = \frac{1}{|\hat{H}_c|}\sum_{\boldsymbol{h}\in\hat{H}_c}\boldsymbol{h}$, where $\boldsymbol{h}$ is the BERT feature. This, however, may not be discriminative enough to differentiate similar classes that share many common descriptions (e.g., dog classes share common phrase like "a breed of dogs" and "having a coat or a tail").

We therefore propose to identify informative sentences that can enlarge the difference of $\boldsymbol{a}_c$ between classes. Concretely, we learn to assign each sentence a weight λ, such that the resulting **weighted average** $\boldsymbol{a}_c = \frac{1}{|\hat{H}_c|}\sum_{\boldsymbol{h}\in\hat{H}_c}\lambda(\boldsymbol{h})\times\boldsymbol{h}$ can be more distinctive. We model $\lambda(\cdot)\in\mathbb{R}$ by a multi-layer perceptron (MLP) b_ψ

$$\lambda(\boldsymbol{h}) = \frac{\exp(b_\psi(\boldsymbol{h}))}{\sum_{\boldsymbol{h}'\in\hat{H}_c}\exp(b_\psi(\boldsymbol{h}'))}. \tag{5.3}$$

We learn b_ψ to meet two criteria. On the one hand, for very similar classes c and c' whose similarity $\cos(\boldsymbol{a}_c, \boldsymbol{a}_{c'})$ is larger than a threshold τ, we want $\cos(\boldsymbol{a}_c, \boldsymbol{a}_{c'})$ to be smaller than τ so they can be discriminable. On the other hand, for other pair of less similar classes, we want their similarity to follow the **average** semantic representation $\bar{\boldsymbol{a}}_c$[5].

To this end, we initialize b_ψ such that the initial $\boldsymbol{a}_c$ is close to $\bar{\boldsymbol{a}}_c$. We do so by first learning b_ψ to minimize the following objective

$$\sum_{c\in S\cup U}\max\{0, \varepsilon - \cos(\boldsymbol{a}_c, \bar{\boldsymbol{a}}_c)\}. \tag{5.4}$$

We set $\varepsilon = 0.9$, forcing $\boldsymbol{a}_c$ and $\bar{\boldsymbol{a}}_c$ of the same class to have $\cos(\boldsymbol{a}_c, \bar{\boldsymbol{a}}_c) > 0.9$. We then fine-tune b_ψ by minimizing the following objective

$$\sum_{c}^{S\cup U}\sum_{c\neq c'}^{S\cup U}\max\{0, \cos(\boldsymbol{a}_c, \boldsymbol{a}_{c'}) - \tau\}. \tag{5.5}$$

We assign τ a high value (*e.g.*, 0.95) to only penalize overly similar semantic representations.

[5] The purpose of introducing $\lambda(\cdot)$ is to improve $\boldsymbol{a}_c$ from the average representation $\bar{\boldsymbol{a}}_c$ to differentiate similar classes.

5.4 Experiments

5.4.1 Dataset and splits: ImageNet

We use the ImageNet Fall 2011 dataset [59] with $21,842$ classes. We use the 1K classes in ILSVRC 2012 [222] for DeViSE training and validation (cf. Equation 5.1), leaving the remaining $20,842$ classes as unseen classes for testing. We follow [36] to consider three tasks, 2-Hop, 3-Hop, and ALL, corresponding to **1,290**, **5,984**, and **14,840** unseen classes *that have Wikipedia pages and word vectors* and are within two, three, and arbitrary tree hop distances (w.r.t. the ImageNet hierarchy) to the 1K classes. On average, each page contains **80** sentences. For images, we use the $2,048$-dimensional ResNet visual features [94] provided by [277]. For sentences, we use a 12-layer pre-trained BERT model [62]. We denote by BERT_p the pre-trained BERT and BERT_f the one fine-tuned with DeViSE.

5.4.2 Baselines, variants, and metrics

Word vectors of class names are the standard semantic representations for ImageNet. Here we compare to the state-of-the-art **w2v-v2** provided by [37], corresponding to a skip-gram model [188] trained with ten passes of the Wikipedia dump corpus. For ours, we compare using all sentences **(NO)**, visual sections **($\textbf{Vis}_\textbf{sec}$)** or visual clusters **($\textbf{Vis}_\textbf{clu}$)**, and both **($\textbf{Vis}_\textbf{sec-clu}$)**. On average, **$\textbf{Vis}_\textbf{sec-clu}$** filters out **57**% of the sentences per class. We denote **weighted average** (Section 5.3.4) by $\text{BERT}_\textbf{p-w}$ and $\text{BERT}_\textbf{f-w}$.

The original DeViSE [69] has f_θ and g_ϕ as identity functions. Here, we consider a stronger version, DeViSE*, in which we model f_θ and g_ϕ each by a two-hidden layers multi-layer perceptron (MLP). We also experiment with two state-of-the-art ZSL algorithms, EXEM [37] and HVE [171].

We use the average *per-class* Top-1 classification accuracy as the metric [277].

Model	Type	Filter	2-Hop	3-Hop	ALL
Random	-	-	0.078	0.017	0.007
DeViSE	w2v-v2	-	6.45	1.99	0.78
	BERT$_{\mathrm{p}}$	No	6.73	2.23	0.83
DeViSE*	w2v-v2	-	11.55	3.07	1.48
	BERT$_{\mathrm{p}}$	No	13.84	4.05	1.75
		Vis$_{\mathrm{sec}}$	15.56	4.41	1.82
		Vis$_{\mathrm{clu}}$	15.72	4.49	2.01
		Vis$_{\mathrm{sec\text{-}clu}}$	15.86	4.65	2.05
	BERT$_{\mathrm{p\text{-}w}}$	Vis$_{\mathrm{sec\text{-}clu}}$	16.32	4.73	2.10
	BERT$_{\mathrm{f}}$	No	17.70	5.17	2.29
		Vis$_{\mathrm{sec}}$	19.52	5.20	2.32
		Vis$_{\mathrm{clu}}$	19.74	5.37	2.36
		Vis$_{\mathrm{sec\text{-}clu}}$	19.82	5.39	2.39
	BERT$_{\mathrm{f\text{-}w}}$	Vis$_{\mathrm{sec\text{-}clu}}$	20.47	5.53	2.42
EXEM	w2v-v2	-	16.04	4.54	1.99
	BERT$_{\mathrm{f}}$	Vis$_{\mathrm{sec\text{-}clu}}$	21.22	5.42	2.37
HVE	w2v-v2	-	8.63	2.38	1.09
	BERT$_{\mathrm{f\text{-}w}}$	Vis$_{\mathrm{sec\text{-}clu}}$	18.42	5.12	2.07

Table 5.3: Comparison of different semantic representations on ImageNet. We use *per-class* Top-1 accuracy(%). The best is in red and the second best in blue.

Model	Type	AwA2				aPY			
		ZSL	GZSL			ZSL	GZSL		
			U	S	H		U	S	H
DeViSE	Visual attributes	59.70	17.10	74.70	27.80	37.02	3.54	78.41	6.73
	w2v-v2	39.56	2.18	69.29	4.22	27.67	1.68	85.53	3.22
	BERT$_{\text{p}}$ + Vis$_{\text{sec-clu}}$	64.32	19.79	72.46	31.09	38.79	3.94	71.60	7.51

Table 5.4: Results on AwA2 and aPY. We compare different semantic representations. Visual attributes are annotated by humans. **GZSL** is the generalized ZSL setting [277]. In GZSL, **U**, **S**, **H** denote unseen class accuracy, seen class accuracy, and their harmonic mean, respectively. We use *per-class* Top-1 accuracy (%).

5.4.3 Main results

Table 5.3 summarizes the results on ImageNet. In combining with each ZSL algorithm, our semantic representations **Vis$_{\text{sec-clu}}$** that uses visual sections and visual clusters for sentence extraction outperforms **w2v-v2**. More discussions are as follows.

BERT vs. w2v-v2. For both DeViSE* and DeViSE, BERT$_{\text{p}}$ by averaging all the sentences in a Wikipedia page outperforms w2v-v2, suggesting that representing a class by its document is more powerful than its word vector.

DeViSE* vs. DeViSE. Adding MLPs to DeViSE largely improves its accuracy: from 0.78% (DeViSE + w2v-v2) to 1.48% (DeViSE* + w2v-v2) at ALL. In the following, we then focus on DeViSE*.

Visual sentence extraction. Comparing different strategies for BERT$_{\text{p}}$, we see both **Vis$_{\text{clu}}$** and **Vis$_{\text{sec}}$** largely improves **NO**, demonstrating the effectiveness of sentence selection. Combining the two sets of sentences (**Vis$_{\text{sec-clu}}$**) leads to a further boost.

Fine-tuning BERT. BERT can be fine-tuned together with DeViSE*. The resulting BERT$_{\text{f}}$ has a notable gain over BERT$_{\text{p}}$ (*e.g.*, 2.39% vs. 2.05%).

Weighted average. With the weighted average ($\text{BERT}_{\text{p-}\mathbf{w}}$, $\text{BERT}_{\text{f-}\mathbf{w}}$), we obtain the best accuracy.

ZSL algorithms. EXEM + w2v-v2 outperforms DeViSE* + w2v-v2, but falls behind DeViSE* + $\text{BERT}_{\text{p-}\mathbf{w}}$ (or BERT_{f}, $\text{BERT}_{\text{f-}\mathbf{w}}$). This suggests that algorithm design and semantic representations are both crucial. Importantly, EXEM and HVE can be improved using our proposed semantic representations, demonstrating the applicability and generalizability of our approach.

5.4.4 Results on other datasets

Table 5.4 summarizes the results on AwA2 [277] and aPY [67]. The former has 40 seen and 10 unseen classes; the latter has 20 seen and 12 unseen classes. We apply DeViSE together with the $2,048$-dimensional ResNet features [94] provided by [277]. Our proposed semantic representations (*i.e.*, $\mathbf{BERT_p}$ + $\text{Vis}_{\text{sec-clu}}$) outperform **w2-v2** and the manually annotated visual attributes on both the ZSL and generalized ZSL (GZSL) settings. These improved results on ImageNet, AwA2, and aPY demonstrate our proposed method's applicability to multiple datasets.

5.4.5 Analysis on ImageNet

To further justify the effectiveness of our approach, we compare to additional baselines in Table 5.5.

- $\mathbf{BERT_{p\text{-}w\text{-}direct}}$: it directly learns b_ψ (Equation 5.3) as part of the DeViSE objective. Namely, we directly learn b_ψ to identify visual sentences, without our proposed selection mechanisms, such that the resulting $\boldsymbol{a}_c$ optimizes Equation 5.1.
- $\mathbf{Par_{1st}}$: it uses the first paragraph of a document.
- $\mathbf{Cls_{name}}$: it uses the sentences of a Wikipedia page that contain the class name.

Model	Type	Filter	2-Hop	3-Hop	ALL
DeViSE*	BERT$_{\text{p}}$	No	13.84	4.05	1.75
	BERT$_{\text{p-}\textbf{w-direct}}$	No	14.85	4.25	1.79
	BERT$_{\text{p}}$	Par$_{\text{1st}}$	13.48	4.10	1.78
		Cls$_{\text{name}}$	14.82	3.31	1.40
		Vis$_{\text{sec}}$	15.56	4.41	1.82
		Vis$_{\text{clu}}$	15.72	4.49	2.01
		Vis$_{\text{sec-clu}}$	15.86	4.65	2.05
	BERT$_{\text{p-}\textbf{w}}$	Vis$_{\text{sec-clu}}$	16.32	4.73	2.10

Table 5.5: The effectiveness of our visual sentence extraction. **BERT$_{\textbf{p-w-direct}}$** directly learns visual sentences without our sentence selection. **Par$_{\textbf{1st}}$** and **Cls$_{\textbf{name}}$** use the first paragraph and sentences containing the class name, respectively.

As shown in Table 5.5, our proposed sentence selection mechanisms (*i.e.*, Vis$_{\text{sec}}$, Vis$_{\text{clu}}$, and Vis$_{\text{sec-clu}}$) outperform all the three baselines.

5.5 Summary

ZSL relies heavily on the quality of semantic representations. Most recent work, however, focuses solely on algorithm design, trying to squeeze out the last bit of information from the pre-define, likely poor semantic representations. [37] has shown that existing algorithms are trapped in the plateau of inferior semantic representations. Improving the representations is thus more crucial for ZSL. We investigate this direction and show promising results by extracting *distinctive visual* sentences from documents for representations, which can be easily used by any ZSL algorithms.

Image / Label		Semantic Type	Top 5 prediction
	Tiger	BERT_{f-w}	Tiger, Tiger cat, Tabby, leopard, Jaguar
		w2v-v2	Tiger cat, Tiger, Cougar, Madagascar cat, Standard poodle
	Scooter	BERT_{f-w}	Scooter, Tandem bicycle, mountain bike, forklift, police van
		w2v-v2	Tandem bicycle, Scooter, forklift, mountain bike, police van
	Grey whale	BERT_{f-w}	Grey whale, Killer whale, Pelican, Sea lion, Sturgeon
		w2v-v2	Killer whale, Grey whale, Sea lion, Ice bear, Cocker spaniel
	Sports car	BERT_{f-w}	Sports car, Race car, Garden cart, Minivan, Limousine
		w2v-v2	Jeep, Sports car, ambulance, fire truck, taxi
	Printer	BERT_{f-w}	Printer, Hard disc, Polaroid camera, Slot machine, Chocolate sauce
		w2v-v2	Hard disc, Cannon, Printer, Thimble, Stethoscope

Figure 5.2: Qualitative results between **$\text{BERT}_{\text{f-w}}$** and **w2v-v2** on ImageNet. For each image, we report Top 5 prediction. While **w2v-v2** is not able to distinguish similar classes (e.g. Predicting "Scooter" as "Tandem bicycle"), our **$\text{BERT}_{\text{f-w}}$** differentiates them.

Chapter 6: HTML representations with visual contextualization

In this chapter, we explore another study to improve representations for V&L tasks. Concretely, we focus on **HTML representations** for web navigation. Automatic web navigation aims to build a web agent that can follow language instructions to execute complex and diverse tasks on real-world websites. Existing work primarily takes HTML documents as input, which define the contents and action spaces (*i.e.*, actionable elements and operations) of webpages. Nevertheless, HTML documents may not provide a clear task-related context for each element, making it hard to select the right (sequence of) actions. We thus propose to contextualize HTML elements through their "dual views" in webpage screenshots: each HTML element has its corresponding bounding box and visual content in the screenshot. We build upon the insight—*web developers tend to arrange task-related elements nearby on webpages to enhance user experiences*—and propose to contextualize each element with its neighbor elements in the screenshot, using both textual and visual features. The resulting representations of HTML elements are more informative for the agent to take action during web navigation.

6.1 Introduction

We study automatic web navigation with natural language instructions [60, 285]. This problem is crucial as it can potentially streamline and automate a wide range of tasks

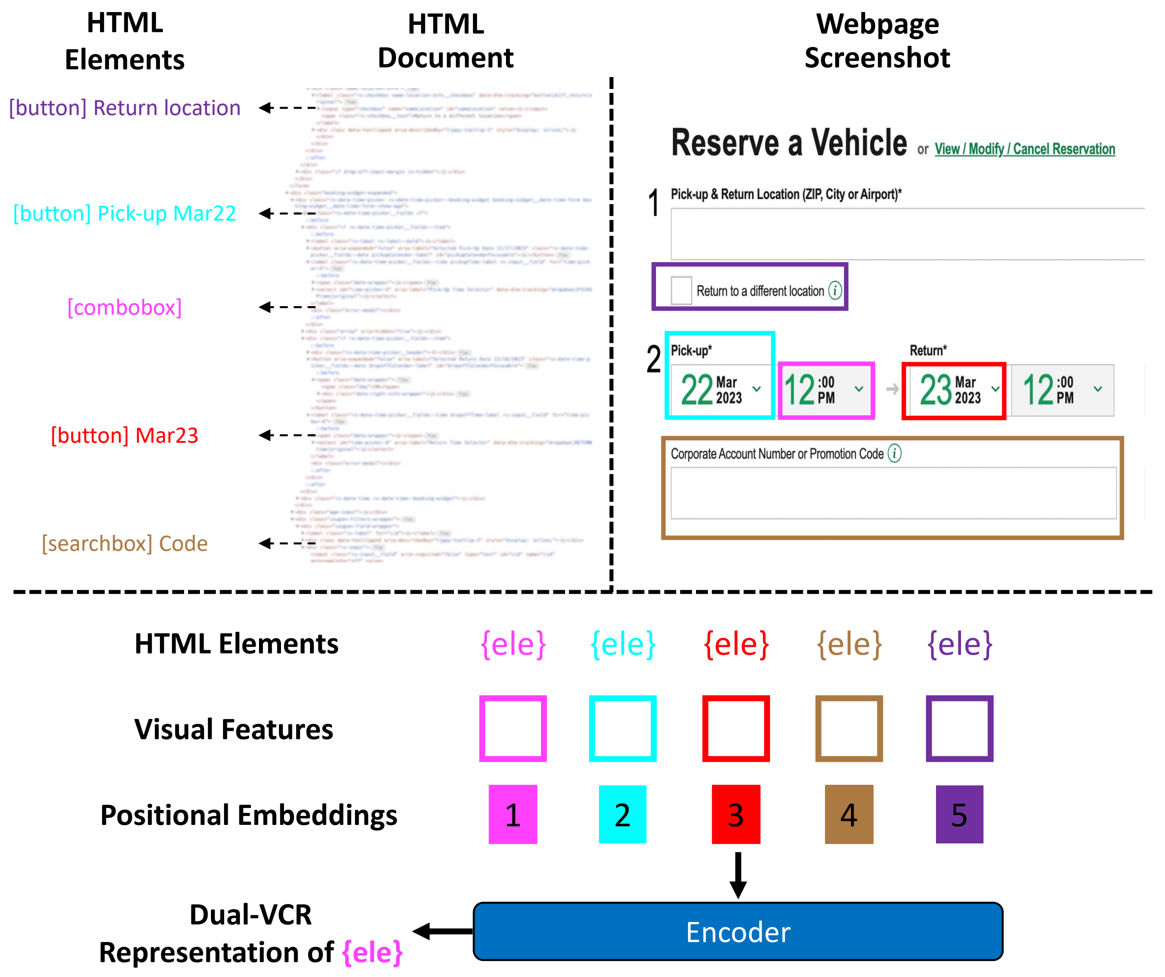


Figure 6.1: **Overview of our proposed Dual-View Contextualized Representation (DUAL-VCR).** HTML elements (*e.g.*, "[combobox]") may not have clear contexts for solving web navigation tasks (*e.g.*, "Find the lowest rent truck with a pick-up time at 11 am on March 27."). DUAL-VCR contextualizes each element with its neighbors in the screenshot (*e.g.*, "[button] Pick-up Mar22") to obtain more informative representations for decision-making.

in our increasingly web-centric world, from online shopping to accessing information. Successfully solving this problem can also broadly advance artificial intelligence as it requires understanding and executing various tasks by interacting with dynamic and complex real-world (web) environments.

Existing work primarily takes HTML documents as the web agent's input [86, 60, 240], which define the meaning and layout of webpage content. Written partially in natural language, HTML documents enable the use of large language models (LLMs) [52, 195, 30, 102, 257, 50, 49, 249] to ground language instructions (*e.g.*, "Find one-way flights from New York to Toronto.") in web environments. Moreover, elements in HTML documents directly define the space of actions (*e.g.*, element "[button] Search" with operation "click"), preventing the agent from hallucinating infeasible actions.

With that being said, HTML documents may lack a clear task-related context for each element, impeding the agent from selecting the right (sequence of) actions to complete a task. HTML is quite flexible for web developers to arrange their code. Even semantically related elements, such as an actionable element (*e.g.*, "drop-down box") and its label element (*e.g.*, "Number of Passengers"), may not be located nearby in the document or the DOM tree. This problem also applies to elements relevant to solving a task. While LLMs may learn to capture the context, a raw HTML document of real-world webpages is often quite huge, consisting of tens of thousands of tokens, making it either infeasible or cost-prohibitive to be directly fed into LLMs [86, 60, 240].

In this paper, we propose to enhance the context of each HTML element by leveraging its "dual view" in the screenshot of the rendered webpage: many of the HTML elements (including the actionable ones) are visible in the screenshot and have their corresponding bounding boxes[6]. Taking the insight—*semantically related and task-related HTML elements are often located nearby on the webpage* to facilitate user experiences—we propose to contextualize each HTML element with its neighbors in the screenshot. Concretely, when encoding each HTML element, we 1) append its spatially adjacent elements with positional embeddings and 2) incorporate both the visual and textual features (Figure 6.1).

While simple, our method, which we name **Dual-View Contextualized Representation (DUAL-VCR)**, has several compelling properties that benefit web navigation fundamentally. First, **DUAL-VCR** uses the built-in feature of HTML documents to align textual and visual content, making it robust to complex and diverse websites. Second, **DUAL-VCR** effectively leverages visual cues on the webpages, which are designed to ease users' efforts in understanding and completing tasks. Specifically, **DUAL-VCR** connects *visually proximate elements that are often semantically related and task-related*, providing the agent with more explicit contexts to take not only individual actions but also the sequence of actions. Last but not least, **DUAL-VCR** can potentially be integrated into any web navigation algorithms that take HTML documents as input.

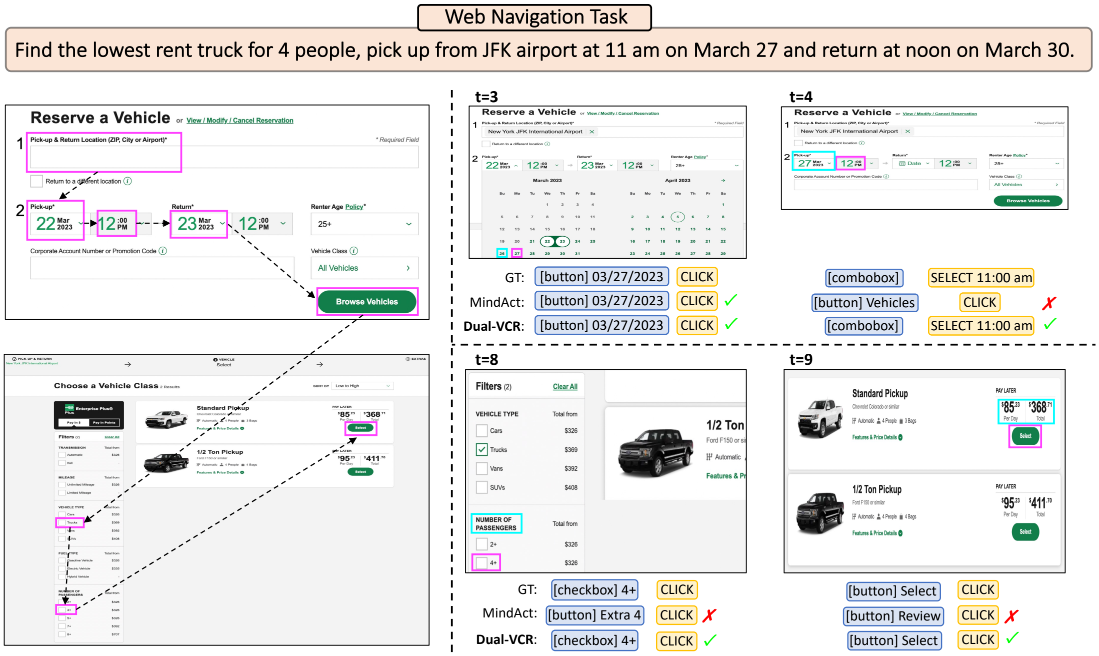
We validate DUAL-VCR on the Mind2Web dataset [60], the largest web navigation benchmark with over 2,000 tasks curated from 137 real-world websites across 31 domains, including restaurants, airlines, public services, etc. Concretely, we implement **DUAL-VCR** on top of the **MindAct** algorithm [60], which was proposed to tackle huge HTML documents. In short, at each action, MindAct first applies a small LM to rank each HTML element to shrink the document; it then uses an LLM to predict the action. We integrate DUAL-VCR into both steps to enhance the context for element ranking and decision-making. DUAL-VCR consistently improves MindAct across all three scenarios (cross-task, cross-website,

[6] These bounding boxes can be directly inferred from the HTML document without the need to detect them.

and cross-domain), leading to a **3.7**% absolute gain on average over nine evaluation metrics. Moreover, DUAL-VCR notably outperforms baselines that use entire HTML documents or screenshots as input, offering significant advantages in computation and accuracy.

Our contributions are three-folded:

- We propose DUAL-VCR, a simple and effective dual-view representation of HTML elements for web navigation.
- DUAL-VCR consistently outperforms baselines on the real-world web navigation benchmark Mind2Web [60].
- We conduct comprehensive analyses to understand the effect of our design choices on web navigation performance.



Find the lowest rent truck for 4 people, pick up from JFK airport at 11 am on March 27 and return at noon on March 30.

	t=3			t=4		
GT:	[button] 03/27/2023	CLICK		[combobox]	SELECT 11:00 am	
MindAct:	[button] 03/27/2023	CLICK	✓	[button] Vehicles	CLICK	✗
Dual-VCR:	[button] 03/27/2023	CLICK	✓	[combobox]	SELECT 11:00 am	✓

	t=8			t=9		
GT:	[checkbox] 4+	CLICK		[button] Select	CLICK	
MindAct:	[button] Extra 4	CLICK	✗	[button] Review	CLICK	✗
Dual-VCR:	[checkbox] 4+	CLICK	✓	[button] Select	CLICK	✓

Figure 6.2: **Example of real-world web navigation. Top**: the web navigation task described in natural language. **Left**: the sequence of HTML elements (visualized on webpages, not HTML documents) to interact with to complete the task. We superimpose bounding boxes and arrows to locate the target elements and indicate their order. **Right**: the detail at each time step (we showed $t = \{3,4,8,9\}$ for brevity). GT: ground-truth action (Element with Operation). We compare the predicted actions by MindAct [60] and our **Dual-VCR**. The bounding box and bounding box indicate the target element and one of its neighbors encoded by **Dual-VCR**. As shown, **Dual-VCR** correctly predicts the elements and operations at "all" time steps, taking advantage of the much richer task-related dual-view context it encodes.

6.2 Literature survey on web navigation

Web navigation datasets. Several prior studies [108, 285, 157, 244, 31] have introduced promising benchmarks for assessing agents in web navigation tasks. However, these benchmarks are often limited to a narrow range of website domains or confined to simplified simulated environments. For instance, MiniWob++ [108] and WebShop [285] collected a set of websites including daily tasks (*e.g.*, shopping), but each website only has fewer than fifty HTML elements on average. Some other studies [157, 244, 31] instead explored other domains, including mobile applications, but their action spaces are often simpler than web navigation. Recently, Mind2Web [60] released the first large-scale web navigation benchmark consisting of over 2K tasks from various real-world websites. This enables a comprehensive understanding of web agent's behaviors in "real-world" scenarios.

The use of HTML documents. Most earlier work [108, 285, 166, 116] focused on simple navigation scenarios like MiniWob++ [108]. Due to the brevity of its HTML documents, they input whole HTML documents into LLMs to complete the web navigation tasks. A few studies represented HTML documents in a more dense format. For instance, ASH [240] summarized the HTML document using LLMs with hierarchical prompting. DOM-Q-NET [116] leveraged a graph neural network to represent a document as a graph. For real-world web navigation (*e.g.*, Mind2Web), HTML documents are often overly lengthy and complex. Thus, recent studies [60, 71, 86] applied text-based filtering to first identify key HTML elements within the document and only used the selected elements to complete the task. While all these prior methods are promising, the HTML document alone may not provide a clear task-related context for each element, making it challenging to select the right actions. Our approach instead enhances the context of each HTML element based on their dual view in the screenshot.

The use of webpage screenshots. Beyond using HTML documents, several studies [108, 285, 230, 148, 71, 109, 299, 91, 99] have explored the incorporation of screenshots for web navigation. Some of them [108, 71, 109, 91, 99, 299] utilized both screenshots and HTML documents to learn their joint representations during decision-making. Some others [230, 148, 48] solely relied on screenshots, bypassing the use of HTML documents. We note that all prior methods primarily focused on utilizing "whole" screenshots. In contrast, we shift the focus to neighboring elements within the screenshot, providing significant benefits in computation and accuracy.

6.3 Approach: DUAL-VCR

We introduce **Dual-View Contextualized Representation (DUAL-VCR)** for enhanced web navigation. To begin with, we provide a brief background about web navigation.

6.3.1 Background: web navigation

A web navigation task consists of a website S (*e.g.*, an airline website) and an instruction q ("Find one-way flights from New York to Toronto."). Given (S,q), a web agent f needs to decide and perform a sequence of actions $a = \{a_1, a_2, \cdots, a_t, \cdots\}$ on the website to complete the task. Figure 6.2 (left) gives an illustration.

At time step t, the website has an HTML document H_t, composed of a list of elements $H_t = \{e_{t,1}, e_{t,2}, \cdots, e_{t,N}\}$. These HTML elements jointly define 1) the layout and content on the rendered webpage I_t, and 2) the action space at time t: each candidate action is a pair of an actionable element (*e.g.*, "[textbox] To") and an operation (*e.g.*, "Type Toronto"). After taking action a_t, both the HTML document and webpage will be updated into (H_{t+1}, I_{t+1}). For example, clicking the "[checkbox] One way" on the airline webpage removes the "[textbox] Return date" from the webpage. Namely, the web environment is dynamic, and the agent must take this into account to decide its actions.

Because of the rich content in the HTML document H_t, existing work primarily takes it, together with the instruction q and the action history (*e.g.*, *Type New York in the From box*), as the agent's input at time t to decide the next action (*e.g.*, *Type Toronto in the To box*),

$$a_{t+1} = f(q, H_t, \{a_1, a_2, \cdots, a_t\}). \tag{6.1}$$

One excellent candidate for f is LLMs [52, 195, 30, 102, 257, 50, 49, 249], which have shown straggering sucesses in question answering [270] and logical reasoning [55]. For example, [108, 133] applied LLMs to simplified web navigation.

However, for real-world webpages that easily contain thousands of HTML elements (amounting to tens of thousands of tokens), directly applying LLMs is neither efficient nor effective. As such, recent work [86, 60, 240] employed a two-stage framework: first summarizing the HTML document and then predicting the action. For instance, given the instruction q and the action history at time t, the MindAct algorithm [60] first ranks each HTML element using a small LM. Only the top-K HTML elements are fed into an LLM to predict the next action. (See Figure 6.3 for an illustration.)

6.3.2 Context enhancement

We identify one critical pitfall in the two-stage framework. *Since HTML documents may not provide a clear context for each element, the element ranker and the subsequent action predictor may not perform as effectively as expected.* Figure 6.1 illustrates one such issue: the element "[combobox]" should be paired with "[button] Pick-up Mar22" to fully describe its role, *i.e.*, time for pick-up. However, these two elements are not necessarily nearby in the HTML document.

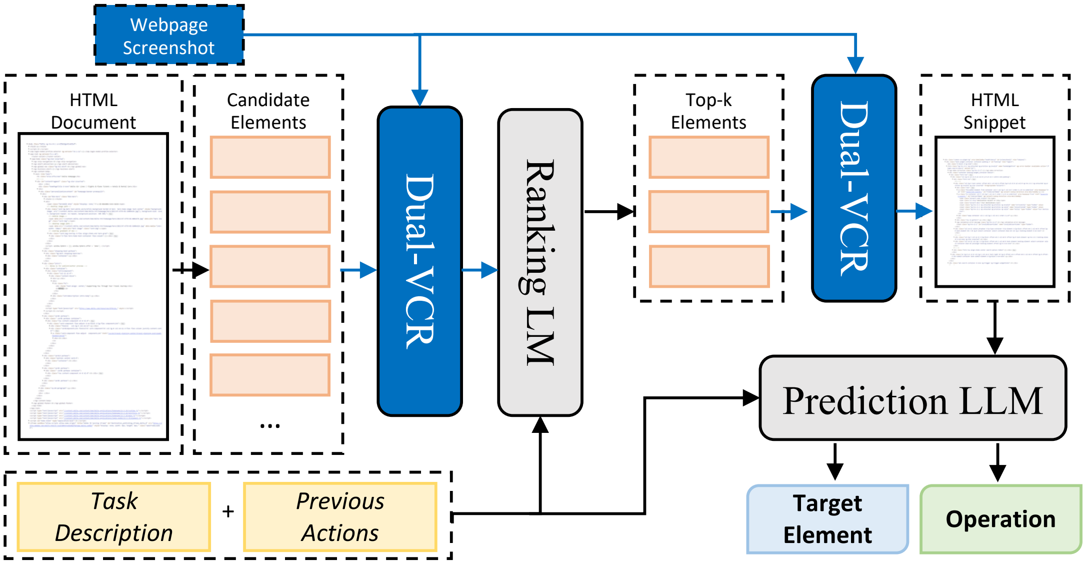


Figure 6.3: **The web navigation pipeline with DUAL-VCR,** built on top of the MindAct algorithm [60]. MindAct uses a small ranking LM to select candidate HTML elements and a prediction LLM to decide actions. Blocks and arrows in NavyBlue indicate the insertion of **DUAL-VCR** for enhanced element representations.

To resolve this issue, we propose to leverage the "dual view" of each HTML element $e_{t,n} \in H_t$ in the rendered webpage I_t to enhance its context. In essence, many HTML elements (including the actionable ones) are visible in I_t. Further, their visual location (*e.g.*, bounding boxes) can be inferred from HTML documents. Since a webpage (specifically, its screenshot) is designed for users to interact with the website visually, we hypothesize that incorporating the visual cues into HTML element representations would benefit the web agent in understanding and completing tasks.

To this end, we propose **Dual-View Contextualized Representation (DUAL-VCR)**. In the screenshot view, we identify the bounding box of each HTML element using a web automation testing tool[7]. Taking the insight—web developers tend to arrange semantically relevant and task-related elements in proximity to each other on the screenshot to enhance

[7] https://playwright.dev/

user experiences—we contextualize each element with its "visual" neighbors. Concretely, we calculate the center points of all elements using their bounding boxes and measure their pairwise distances. For each *candidate* element to be ranked by MindAct, we search for the closest M elements to form its context jointly.

We consider both the visual and textual information to encode the candidate element and its visual neighbors. We extract each element's visual feature using the Pix2Struct Vision Transformer (ViT) [145], which is pre-trained on webpage screenshots. Specifically, we input the whole screenshot I_t into the ViT and apply ROI Align [93, 159] on top of the output embeddings to obtain the feature vector corresponding to each element's bounding box. In the HTML document view, we extract each element's corresponding "HTML text" following MindAct [60].

6.3.3 DUAL-VCR-enhanced element ranker

In MindAct, a small ranking LM is built to predict each element's importance for action prediction. At each time step, the ranking LM takes the element's HTML text tokens, the task description q, and the previous actions as input.

We propose to expand the ranking LM to integrate 1) both visual features and textual features and 2) both the candidate element and its neighbor elements. (See Figure 6.4 for an illustration.) We make the following design choices. To align the visual embedding and textual embedding, we follow the recent practice of vision-and-language models (*e.g.*, BLIP-2 [150], LLaVA [169], LLaVA-1.5 [168]) to learn a linear projection layer to project ViT visual features into the same dimensionality as the token embeddings in the ranking LM. To pair each of the projected visual vectors with its corresponding text tokens and specify each neighbor element in the context, we add positional encoding. Concretely, we

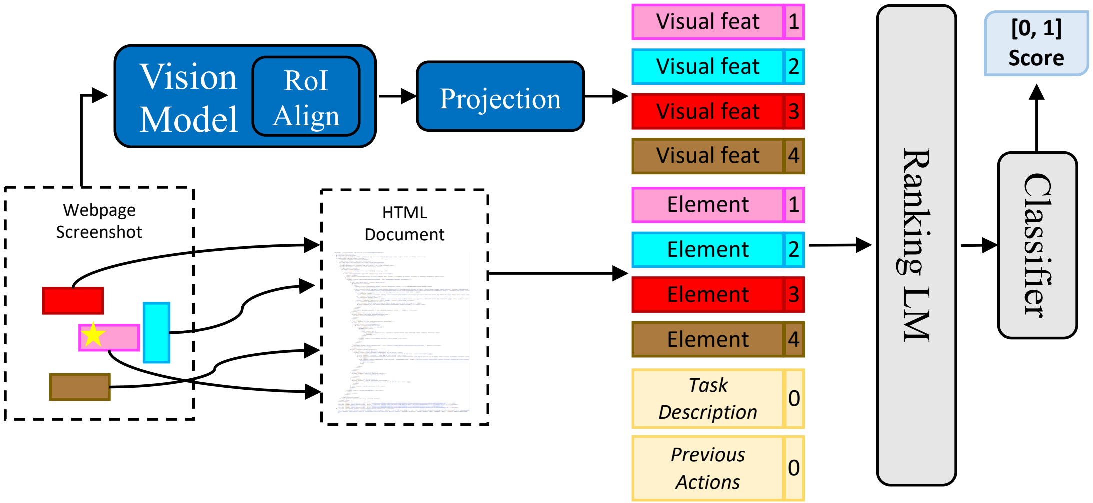


Figure 6.4: **DUAL-VCR-enhanced element ranker**. We contextualize the candidate element (denoted by ⋆) with its neighbors in the screenshot, using both the visual features (by [145]) and textual features (extracted from the HTML document). Positional embeddings are added to specify neighbor elements, learning their spatial relationships and pairing the textual features with visual features. This dual-view contextualized representation is used to rank the candidate element, measuring its relevance to the current task.

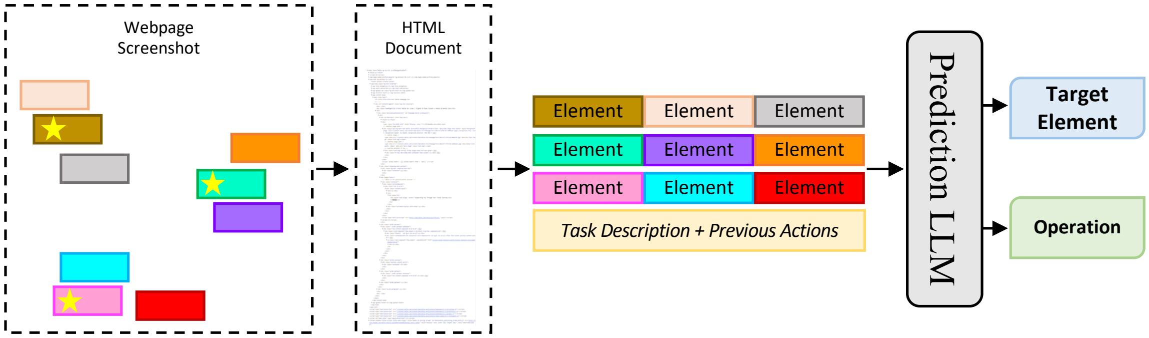


Figure 6.5: **DUAL-VCR-enhanced action predictor**. Given the top-*K* candidate elements (three in the figure, marked with ⋆), DUAL-VCR appends each with its neighbor elements. The resulting HTML snippet, together with the task description and previous actions, is then fed into an LLM for predicting the next action.

sort the neighbors based on their spatial distances from the candidate element and add a learnable positional embedding (unique for each rank) to the neighbor element's visual and text token embeddings. These positionally encoded visual and text token embeddings (of the candidate and the neighbor elements) are fed into the ranking LM; the projected visual features are prepended to the text embeddings, serving as soft visual prompts. In training, we only learn the linear projection layer, the positional embeddings, and the LM while keeping the ViT frozen. This training scheme has been shown to effectively enhance the alignment between vision and language components and improve the pre-trained LM's adaptability to downstream tasks. Please see more details in the supplementary materials.

6.3.4 DUAL-VCR-enhanced action predictor

After obtaining the top-*K* elements from the ranker (§6.3.3), MindAct combines them into an HTML snippet as the input to LLMs. The objective is to predict the action for the current time step, including the target element (*e.g.*, “[textbox] To”) and its associated operation (*e.g.*, “Type Toronto”). Specifically, MindAct converts the target element prediction problem into multiple-choice question-answering.

We apply DUAL-VCR to contextualize each of the answer candidates. Similarly to §6.3.3, we find the M closest neighbors for each candidate element on the screenshot. We then append the HTML text tokens of these M neighbors to the candidate element; we add specific tokens to separate between elements. Figure 6.5 gives an illustration. Please see the supplementary material for more details.

6.3.5 Why DUAL-VCR?

DUAL-VCR leverages and encodes visual cues on the webpage, offering valuable contexts for the HTML elements in element ranking and action prediction. We show two cases.

First, as shown in Figure 6.1, some HTML elements (*e.g.*, “[combobox]”) are quite generic and must be paired with spatially nearby elements (*e.g.*, “[button] Pick-up Mar22”) to specify their meanings (*i.e.*, time for pick-up). Similar examples can be found in Figure 6.2. At $t = 8$, there are two seemingly similar candidates “[checkbox] 4+” and “[button] Extra 4”. Nevertheless, the former is spatially closer to the element “Number of passengers”, indicating its relatedness to the task “... truck for 4 people ...” (see the top of Figure 6.2). At $t = 9$, two identical “[button] Select” elements exist. The only way to differentiate them is through their visual neighbors: one is associated with a lower price than the other. Our DUAL-VCR offers an explicit way to enforce these spatial contexts in the screenshots.

Second, as shown in the left panel of Figure 6.2, consecutive steps to solve a task often involve spatially nearby elements. Completing one step thus introduces a prior that its nearby elements may be the next to take action upon. As both the ranking LM and prediction LLM take the task description q, *past actions*, and our DUAL-VCR representation as input, the models could potentially capture such prior information to increase the success rate for the following action. For example, at $t = 4$, DUAL-VCR successfully takes the action “Select 11:30 am”, likely attributing to its capability to recognize that the previously completed task was the spatially nearby “Select 03/27/2023”.

6.4 Experimental Results

Dataset. We validate DUAL-VCR on Mind2Web [60], a comprehensive benchmark for real-world web navigation. Unlike other benchmarks based on simulated websites with only a few HTML elements, Mind2Web uses over 100 real-world websites with thousands of HTML elements. Concretely, they provide over 2K open-ended tasks collected from 137 real-world websites across 31 different domains, including travel, shopping, public service, etc (Table 6.1). Please see more details in the supplementary material.

Evaluation Tasks. Followed by Mind2Web [60], we evaluate models at three different test splits. In **Cross-Domain**, we evaluate the model's generalizability to a new domain where it has not seen any websites or tasks associated with that domain during training. This split contains 912 tasks in total. In **Cross-Website** (177 tasks), while the model is not exposed to test websites, it is trained on websites from the same domain and potentially with similar tasks. This configuration enables us to evaluate the model's capacity to adapt to entirely new websites within familiar domains and tasks. Similar to the conventional training/test split, **Cross-Task** (252 tasks) randomly splits 20% of the data as a test set, regardless of the domains and the websites. Please see the supplementary material for more details.

Evaluation Metrics. We use the Mind2Web's official metrics. The ranker performance is measured by **Recall@K**, where K is the number of top HTML candidate elements. **Element Accuracy** (Ele. Acc) compares the selected element with the ground-truth elements. **Operation F1** (Op. F1) calculates the token-level F1 score for the predicted operation. **Step Success Rate** (Step SR) measures the success of each step; A step is considered successful only if both the selected element and the predicted operation are correct. For each step, they provide previous "ground-truth" actions with the assumption that the model successfully completes all previous steps.

Baselines. DUAL-VCR is based on MindAct [60], which has a ranking LM and a prediction LLM. Our main baselines are thus its ranker and action predictor, denoted by **MINDACT$_{\textbf{RANK}}$** and **MINDACT$_{\textbf{PRED}}$**. MINDACT$_{\text{RANK}}$ uses DeBERTa$_{\text{base}}$ [95], a small encoder-only LM to rank elements. For action prediction, MINDACT$_{\text{PRED}}$ uses Flan-T5$_{\text{base}}$ [52], an instruction fine-tuned LLM.

Our Models. Aligned with MindAct, we use the same DeBERTa$_{\text{base}}$ [95] / Flan-T5$_{\text{base}}$ [52] for our ranker / action predictor, repsectively. For visual features extraction, we utilize Pix2Struct [145]'s ViT (pre-trained on screenshots) as the visual backbone and apply ROI Align [93] on the element's region. We use two linear layers to project visual features into textual embedding space. Please see the supplementary materials for details on the model training.

Notation of DUAL-VCR. DUAL-VCR has several variations to understand the effect of each of its components in detail. We denote them as follows:

- **DUAL-VCR$_{\textbf{VIS}}$**: Ranker w/ candidate's visual features.
- **DUAL-VCR$_{\textbf{VNEI-TXT}}$**: Ranker w/ neighbors' HTML text.
- **DUAL-VCR$_{\textbf{VNEI-TXT+VIS}}$**: Ranker w/ candidate's visual features and its neighbors' visual features and HTML text.
- **DUAL-VCR$_{\textbf{PRED}}$**: Action predictor w/ neighbors' HTML text.

Dataset	# Websites	Website	# Tasks	Avg # HTML	
		Type		Elements	Tokens
MiniWoB++ [108]	100	Simplified	100	28	500
Mind2Web [60]	137	Real-world	2,350	1,135	44,402

Table 6.1: **Statistics of Mind2Web [60].** Min2Web, the largest web navigation benchmark, collects real-world websites across various domains. The significant volume of content on the webpage (*e.g.*, an average of 1K/44K HTML elements/tokens) poses challenges for LLMs in both computational and learning aspects.

Ranker	Recall			
	@1	@5	@10	@50
MindAct$_{\text{Rank}}$	25.4	61.0	73.5	88.9
Dual-VCR$_{\text{vnei-txt}}$	37.3	70.8	79.3	89.2
Dual-VCR$_{\text{vis}}$	37.1	70.2	79.2	89.1
Dual-VCR$_{\text{vnei-txt+vis}}$	**38.4**	**71.6**	**79.7**	**90.1**

Table 6.2: **Ranking performance.** Visual neighbors' HTML text (Dual-VCR$_{\text{vnei-txt}}$) consistently outperforms MindAct$_{\text{Rank}}$. Moreover, Dual-VCR$_{\text{vnei-txt+vis}}$, using both visual neighbors' HTML text and visual features, performs best, showing the strength of dual-view contextualization in element ranking.

6.4.1 Effectinvess of Dual-VCR

The main goal of our experiments is to show that our dual-view contexutalization is beneficial in (i) finding promising top-K candidates from entire HTML documents (*i.e.*, ranking peformance), and (ii) predicting the action, including both element selection and operation prediction.

Ranking performance. Table 6.2 summarizes the ranking results across different top-K candidate elements. First, we see that incorporating the visual neighbor elements' HTML text (Dual-VCR$_{\text{vnei-txt}}$) consistently and significantly outperforms MindAct$_{\text{Rank}}$ on all Recall@Ks (*e.g.*, 37.3% vs. 25.4% on Recall@1, 79.3% vs. 73.5% on Recall@10),

Ranker	Action Predictor	Cross-Task			Cross-Website			Cross-Domain		
		Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR
MindAct$_{\text{RANK}}$	MindAct$_{\text{PRED}}$	42.0	74.9	41.1	30.7	67.0	30.0	31.5	66.6	31.0
Dual-VCR$_{\text{VNEI-TXT}}$	Dual-VCR$_{\text{PRED}}$	45.3	78.4	44.5	32.0	71.5	31.5	32.4	72.9	32.0
Dual-VCR$_{\text{VNEI-TXT+VIS}}$		**47.0**	**78.7**	**46.0**	**32.7**	**72.0**	**32.5**	**33.2**	**73.3**	**32.5**

Table 6.3: **Results of action prediction.** Our Dual-VCR$_{\text{VNEI-TXT}}$ → Dual-VCR$_{\text{PRED}}$, leveraging visual neighbors' HTML text information, notably improves over the baseline (MindAct$_{\text{RANK}}$ →MindAct$_{\text{PRED}}$) on all nine metrics. Adding visual neighbors' visual features (Dual-VCR$_{\text{VNEI-TXT+VIS}}$) leads to further improvements, highlighting the benefit of dual-view context on real-world web navigation.

suggesting that contextualizing the element with its neighbors indeed helps find the target element. Second, the candidate element's visual features (Dual-VCR$_{\text{VIS}}$) lead to notable improvements over MindAct$_{\text{RANK}}$ (*e.g.*, 70.2% vs. 61.0% on Recall@5). This implies that the visual features offer additional context in differentiating HTML elements, compared to using only its HTML text. Lastly, Dual-VCR$_{\text{VNEI-TXT+VIS}}$ achieves a further boost by leveraging both visual neighbors' HTML text and visual features (*e.g.*, 38.4%/90.1% on Recall@1/@50).

Action prediction performance. Table 6.3 shows the results of action prediction. Compared to the baseline (the combination of MindAct$_{\text{RANK}}$ and MindAct$_{\text{PRED}}$), using the visual neighbors' HTML texts (Dual-VCR$_{\text{VNEI-TXT}}$ → Dual-VCR$_{\text{PRED}}$) notably improves across all metrics. For instance, we achieve gains of 3.4% on Step SR in Cross-Task, 1.3% on Ele. Acc in Cross-Webiste, and 6.3% on Op. F1 in Cross-Domain. These consistent improvements demonstrate the advantages of incorporating visual neighbor information during the model's decision-making process. Moreover, aligning with the ranking result, integrating the visual neighbors' visual features into the ranker (Dual-VCR$_{\text{VNEI-TXT+VIS}}$)

Ranker	Action Predictor	Cross-Task		
		Ele. Acc	Op. F1	Step SR
MindAct$_{\text{RANK}}$	MindAct$_{\text{PRED}}$	42.0	74.9	41.1
Dual-VCR$_{\text{VIS}}$		42.5	75.1	41.5
Dual-VCR$_{\text{VNEI-TXT}}$		44.6	75.7	43.2
Dual-VCR$_{\text{VNEI-TXT+VIS}}$		46.0	78.6	44.8
MindAct$_{\text{RANK}}$	Dual-VCR$_{\text{PRED}}$	44.4	75.2	43.1
Dual-VCR$_{\text{VIS}}$		44.6	76.8	43.8
Dual-VCR$_{\text{VNEI-TXT}}$		45.3	78.4	44.5
Dual-VCR$_{\text{VNEI-TXT+VIS}}$		**47.0**	**78.7**	**46.0**

Table 6.4: **Ablation studies** for validating the importance of each component in Dual-VCR. See §6.4.2 for a detailed discussion.

shows its effectiveness in action prediction as well. Concretely, it achieves the best performance on all nine metrics, along with a 5% maximum gain on each type of metric against the baseline (*e.g.*, Ele. Acc: 47.0% vs. 42.0% on Cross-Task, Op. F1: 72.0% vs. 67.0% on Cross-Website, Step SR: 46.0% vs. 41.1% on Cross-Task).

6.4.2 Analysis

We aim to understand Dual-VCR in detail. We show a) a more in-depth analysis of the main table, b) the interaction between the ranker and the action predictor, c) its effectiveness compared to whole input data and random elements, and d) the effect of different sizes of visual neighbors.

Detailed ablation. Table 6.4 provides more details about the main table to better understand the impact of each component in Dual-VCR. First, we keep the action predictor as MindAct$_{\text{PRED}}$ and focus on the pure effects of our rankers on the action prediction task (*i.e.*, 1st to 4th rows). We see that incorporating the candidate element's visual features (Dual-VCR$_{\text{VIS}}$) achieves a slight but significant improvement over MindAct$_{\text{RANK}}$ across

Ranker	Action Predictor	Top-1			Top-5			Top-10			Top-50		
		Re-call	Ele. Acc	Op. F1	Re-call	Ele. Acc	Op. F1	Re-call	Ele. Acc	Op. F1	Re-call	Ele. Acc	Op. F1
MindAct$_{\text{RANK}}$	MindAct$_{\text{PRED}}$	25.4	24.0	23.7	61.0	39.2	52.1	73.5	41.4	62.8	88.9	42.0	74.9
Dual-VCR$_{\text{VNEI-TXT}}$		**37.3**	**35.5**	**33.5**	**70.8**	**43.1**	**54.1**	**79.3**	**43.9**	**63.0**	**89.2**	**44.6**	**75.7**

Table 6.5: **Relationship between ranker and action predictor on Cross-Task.** The ranker has a linear correlation with the action predictor, suggesting the importance of improving its ranking capabilities for decision-making.

all metrics (*e.g.*, 42.5% vs. 42.0% on Ele. Acc). Furthermore, our ranker with the visual neighbors' HTML text (Dual-VCR$_{\text{VNEI-TXT}}$) outperforms MindAct$_{\text{RANK}}$ by a notable margin of +2.6%/+0.8%/+2.1% on Ele. Acc/Op. F1/Step SR, respectively. Besides, Dual-VCR$_{\text{VNEI-TXT+VIS}}$, which encodes the visual neighbors' visual features, further improves the model's decision-making ability (*e.g.*, 46.0% vs. 44.6% on Ele. Acc). In short, we consistently demonstrate the effectiveness of each component in our ranker.

Second, conversely, we fix the ranker and examine the benefit of encoding visual neighbors' HTML text features into the action predictor (Dual-VCR$_{\text{PRED}}$). Compared to MindAct$_{\text{PRED}}$, Dual-VCR$_{\text{PRED}}$ achieves consistent gains across all rankers. For instance, MindAct$_{\text{RANK}}$ → Dual-VCR$_{\text{PRED}}$ outperforms MindAct$_{\text{RANK}}$ → MindAct$_{\text{PRED}}$ (*e.g.*, 44.4% vs. 42.0% on Ele. Acc). Similarly, when fixing the ranker with Dual-VCR$_{\text{VNEI-TXT+VIS}}$, Dual-VCR$_{\text{PRED}}$ improves over MindAct$_{\text{PRED}}$ (*e.g.*, 46.0% vs. 44.8% on Step SR). This shows directly encoding the visual neighbor's HTML text into the action predictor is beneficial.

Finally, Dual-VCR$_{\text{VNEI-TXT+VIS}}$ and Dual-VCR$_{\text{PRED}}$ are complementary; we achieve the best performance across all metrics when leveraging both (*e.g.*, 47.0%/78.7%/46.0% on Ele. Acc/Op. F1/Step SR). Please see more ablation studies in the supplementary materials.

Ranker-action predictor relationship. We analyze the relationship between the ranker and the action predictor in Table 6.5. We observe a linear connection between the two. Concertely, improving the ranker (*e.g.*, 25.4% vs. 37.3% on Recall@1) correlates with improved action prediction results (*e.g.*, 24.0% vs. 35.5% on Ele. Acc). Aligned with results in §6.4.2, this again highlights the importance of improving the model's ranking ability in web navigation.

Comparison to whole input data. Since HTML documents contain a significant amount of content, such as thousands of HTML elements, conducting experiments with whole data is computationally challenging. Nevertheless, we do our best to report the associated results on Table 6.6 to give more context on the effect of Dual-VCR. First, instead of asking the ranker to prune HTML documents, we directly pass the whole HTML documents into the action predictor (WholeHTML$_{\text{PRED}}$). We see that WholeHTML$_{\text{PRED}}$ performs notably less against the baseline (MindAct$_{\text{PRED}}$) (*i.e.*, 38.6% vs. 42.0% on Ele. Acc). We attribute this to the difficulty of finding the target element among *all thousands* of elements. In contrast, our Dual-VCR$_{\text{PRED}}$ achieves a much better result (*i.e.*, 44.4%) with significantly less amount of input elements.

Second, Dual-VCR outperforms the utilization of whole images. We first use the entire image for the ranker (WholeImage$_{\text{RANK}}$). To extract the image features, we use the same procedure mentioned in §6.3.2, except for providing the region of the whole image instead of that of specific elements. We then use these whole image features, along with the same HTML text input used in MindAct$_{\text{PRED}}$, to train WholeImage$_{\text{RANK}}$. Although the entire image features are shown effective over the baseline (*i.e.*, 43.9% vs. 42.0%), it performs notably less than our approach using the *visual neigbhor*'s visual information (*i.e.*, 46.0% of Dual-VCR$_{\text{VNEI-TXT+VIS}}$). In addition, we conducted a study applying the

Ranker	Action	Cross-Task
	Predictor	Ele. Acc
MindAct$_{\text{RANK}}$	MindAct$_{\text{PRED}}$	42.0
	WholeImage$_{\text{PRED}}$	43.6
	Dual-VCR$_{\text{PRED}}$	44.4
WholeImage$_{\text{RANK}}$	MindAct$_{\text{PRED}}$	43.9
Dual-VCR$_{\text{VNEI-TXT}}$		44.6
Dual-VCR$_{\text{VNEI-TXT+VIS}}$		**46.0**
-	WholeHTML$_{\text{PRED}}$	38.6

Table 6.6: **Visual neighbor vs. whole input data.** Using visual neighbors notably outperforms the use of whole data, offering advantages regarding computational efficiency and performance.

whole image to the action predictor. Specifically, similar to recent vision-and-language models [150, 169, 168], we extract whole image features using fine-tuned ViT [145] and prepend them to the top-50 candidate elements extracted from MindAct$_{\text{RANK}}$ as the input to the LLM (Flan-T5$_{\text{base}}$ [52]). Similar to the result of WholeImage$_{\text{RANK}}$, this action predictor (WholeImage$_{\text{PRED}}$) performs worse than Dual-VCR$_{\text{PRED}}$, which only uses *visual neighbors*' HTML text. Overall, this highlights the advantages of our approach in terms of computational efficiency and performance. See additional results in the supplementary materials.

Visual neighbors offer meaningful contexts. We examine whether visual neighbors provide meaningful context for element ranking and action prediction. To assess this, we compare visual neighboring elements with random elements (Table 6.7). Specifically, We randomly select (five) elements from HTML documents and use them to train either the ranker or the action predictor. While our ranker (*e.g.*, Dual-VCR$_{\text{VNEI-TXT}}$) notably improves the ranking performance over MindAct$_{\text{RANK}}$ (*e.g.*, 89.2% vs. 88.9%), the "random" ranker performs less than MindAct$_{\text{RANK}}$ (*e.g.*, 86.7% vs. 88.9%). This, in turn, leads to a significant

Ranker	Recall	Action	Cross-Task	
	@50	Predictor	Ele. Acc	Op. F1
MindAct$_{\text{RANK}}$	88.9	MindAct$_{\text{PRED}}$	42.0	74.9
		Random$_{\text{PRED}}$	41.5	73.6
		Dual-VCR$_{\text{PRED}}$	44.4	75.2
Random$_{\text{RANK}}$	86.7	MindAct$_{\text{PRED}}$	40.6	72.0
Dual-VCR$_{\text{VNEI-TXT}}$	**89.2**		**44.6**	**75.7**

Table 6.7: **Visual neighbors vs. random elements.** Visual neighbors provide meaningful contexts for web navigation, notably outperforming elements randomly extracted from HTML documents.

Ranker			Cross-Task	
Method	# neighbors	Recall@50	Ele. Acc	Op. F1
Dual-VCR$_{\text{VIS}}$	0	89.1	42.5	75.1
Dual-VCR$_{\text{VNEI-TXT+VIS}}$	3	89.7	45.5	77.3
	5	**90.1**	**46.0**	**78.6**
	10	89.5	45.2	77.0

Table 6.8: **Effects of the number of neighbors on ranker.** Choosing the right size of visual neighbors is important for element ranking, and the size of five is found to be most effective for Mind2Web [60]. We fix the action predictor with MindAct$_{\text{PRED}}$.

performance drop in the action prediction (*e.g.*, 42.0% vs. 40.6% on Ele. Acc). Similarly, compared to the MindAct$_{\text{PRED}}$, including random elements in the action predictor hurts the action prediction performance (*e.g.*, 74.9% vs. 73.6 on Op. F1) while visual neighbors are beneficial (*e.g.*, 75.2%). In sum, we empirically demonstrate the benefits of context in visual neighbors for web navigation.

Effects of the number of visual neighbors. We ablate the impact of varying sizes of visual neighbors, starting with Table 6.8, which shows its effect on the ranker while maintaining the same action predictor (MindAct$_{\text{PRED}}$). We observe a linear correlation between the size of visual neighbors and their ranking/action prediction performance. For instance, increasing

Action Predictor		Cross-Task	
Method	# neighbors	Ele. Acc	Op. F1
MINDACT$_{\text{PRED}}$	0	46.0	78.6
DUAL-VCR$_{\text{PRED}}$	3	46.4	78.7
	5	**47.0**	**78.7**
	10	46.2	78.6

Table 6.9: **Effects of the number of neighbors on action predictor.** Similar to Table 6.8, the size of five is most appropriate for the action prediction. We use DUAL-VCR$_{\text{VNEI-TXT+VIS}}$ for the ranker.

the size of neighbors up to five shows consistent improvements (*e.g.*, 89.1%→90.1% on Recall@50 and 75.1%→78.6% on Op. F1). However, considering too many neighbors (*e.g.*, the size of ten) hurts the performance. For example, increasing the size from five to ten decreases the element accuracy from 46.0% to 45.2%. We also see a similar pattern when ablating the effect of the visual neighbor size on the action predictor (Table 6.9). Concretely, while keeping the same ranker (DUAL-VCR$_{\text{VNEI-TXT+VIS}}$), the action performance increases up to the size of five (*e.g.*, 46.0%→47.0% on Ele. Acc) but decreases when the size becomes ten (*e.g.*, 46.2% on Ele. Acc). Overall, this suggests that choosing an appropriate number of neighbors is necessary for both element ranking and action prediction.

6.5 Summary

We introduce DUAL-VCR to effectively represent HTML elements for web navigation. DUAL-VCR contextualizes each element with its visual neighbor elements, leveraging both textual and visual features. DUAL-VCR consistently improves real-world web navigation in the Mind2Web benchmark, supported by comprehensive analyses.

Part V: V&L Data Learning

Chapter 7: Learning from data with appropriate learning objectives

In Parts II and III, we have discussed data curation and representation, respectively, as means to advance vision-language (V&L) systems. This chapter delves into **data learning**, enabling models to acquire V&L knowledge or capabilities from data.

7.1 Introduction

Understanding the role of text as it appears in the context of a visual scene is important in various real-world applications, *e.g.*, from automatically organizing images of receipts, to assisting visually-impaired users in overcoming challenges related to comprehension of non-Braille writing in their surroundings, to enabling autonomous robots to make safe decisions in environments designed for humans. As a result, scene-text understanding (STU) has received increased attention in vision-and-language (V&L) understanding tasks, such as visual question answering (VQA) [237, 23, 191, 269, 186, 185, 184] or image captioning [235, 89, 158]. Please see Figure 7.1 for an illustration.

We identify two distinct capabilities that models targeting STU must address: (i) *recognizing* text in a visual scene and (ii) *connecting* the text to its context in the scene. Previous solutions that target STU tasks [237, 235, 104, 284] often delegate scene-text recognition to off-the-shelf OCR (Optical Character Recognition) systems [235, 26] and model the visual context using pre-computed object-detection features. These two streams of information

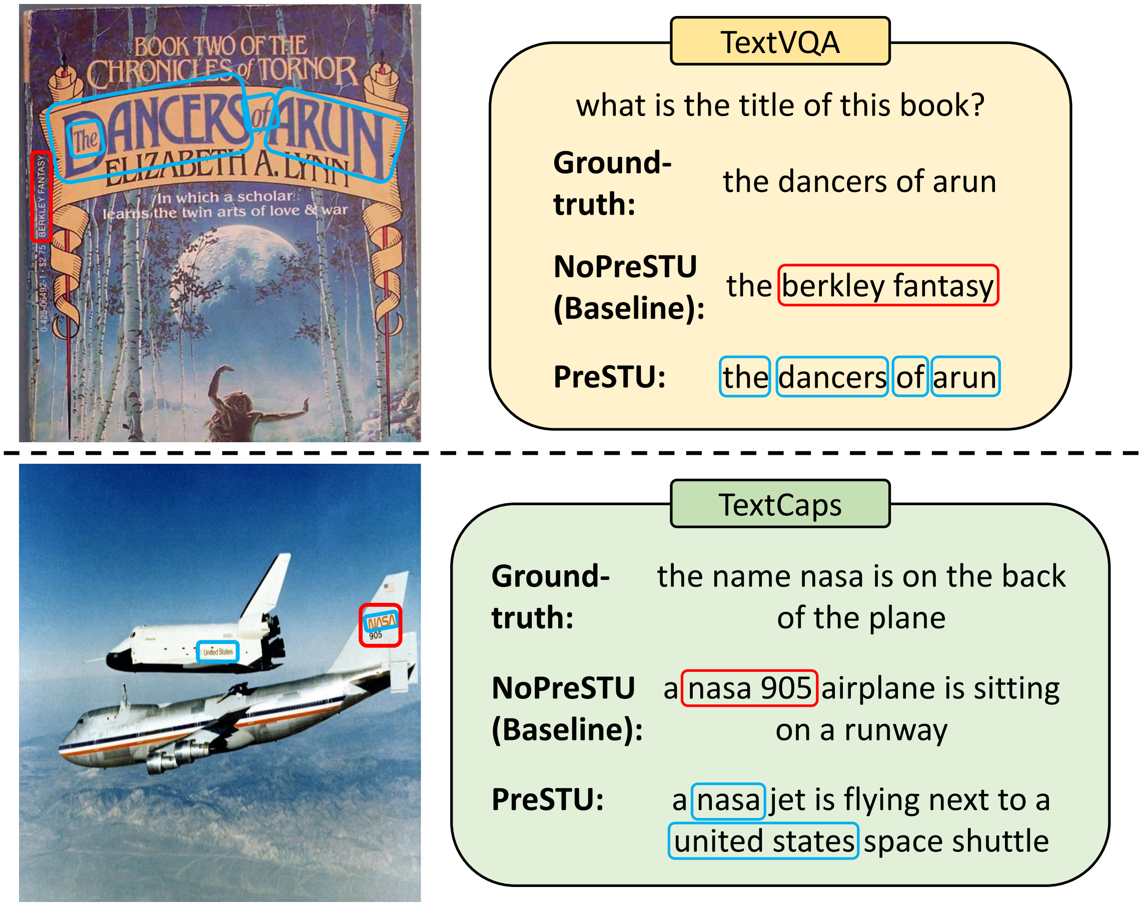


Figure 7.1: **Example of scene-text understanding (STU) tasks.** NoPreSTU (baseline) and PreSTU share the same V&L model, but PreSTU is pre-trained on our proposed pre-training objectives. Scene texts are highlighted by bounding boxes. Unlike the baseline, PreSTU correctly predicts the title of the book on scene-text VQA (TextVQA [237]) and even generates a more detailed scene-text caption (*e.g.*, “united states space shuttle”) than the ground-truth annotated by humans (TextCaps [235]).

(noisy OCR strings and visual features on detected objects) are used as input into a V&L model. While achieving decent results, these methods heavily rely on the quality of the upstream OCR system and lack a direct connection between the text being recognized and a high-fidelity representation of its context.

More concretely, previous methods have not fully explored pre-training objectives that specifically target STU. In general, V&L pre-training objectives (*e.g.*, masked language modeling, image-text matching [176], etc.) have been proven effective for learning and became the go-to approach in V&L research. However, these objectives typically do not require a model to understand the role of text embedded in a visual context. For instance, LaTr [22] ignores the visual context during pre-training and instead focuses on modeling the co-occurrence statistics of layout-aware text-only OCR tokens. Even in systems that do perform STU pre-training, such as TAP [284], their models are built upon the aforementioned pipeline. Specifically, TAP represents the visual input by a set of object features detected and extracted by FRCNN [217]. As a result, it may lose some visual contexts that cannot be captured by objectness (*e.g.*, activities) but are relevant to understand the role of recognized text.

In this paper, we address such a challenge by incorporating an OCR-aware learning objective in the context of a high-fidelity representation of the image context. We adopt a Transformer-based [258] encoder-decoder V&L architecture, using a T5 [209] backbone. The model takes both image and text inputs. For the former, we extract fine-tunable visual features directly from image *pixels* using a ViT [64] encoder, rather than adopting frozen visual features from pre-detected objects [217]. For the latter, we concatenate task-specific text tokens (*e.g.*, task prompts) with tokens extracted from an off-the-shelf OCR system, in a manner that allows the model to interpret (via the prompt) the OCR tokens in the context of the image.

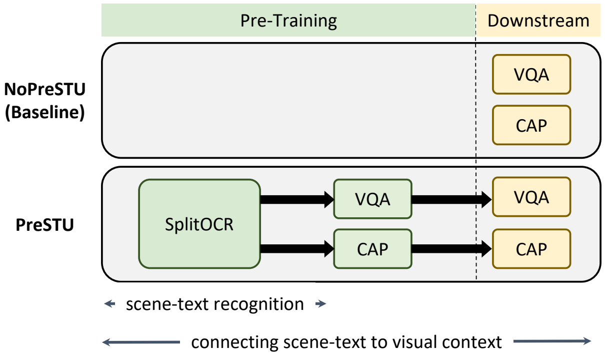
Building upon this model, we propose PreSTU, a novel recipe for **Pre**-training for **S**cene-**T**ext **U**nderstanding (Figure 7.2). PreSTU consists of two main steps. First, it teaches the model to recognize scene text from image pixels[8] and at the same time connect scene text to the visual context. Specifically, given an image and the "part" of the scene texts in the image, the model is pre-trained to predict the "rest" of the scene texts. We call this step SPLITOCR. Second, it teaches the model to further strengthen the connection between scene text and visual context by pre-training with OCR-aware downstream tasks (*e.g.*, VQA and CAP). For pre-training, we leverage large-scale image-text resources [229, 39, 23], with the (noisy) scene text extracted by the off-the-shelf OCR system (Google Cloud OCR[9]).

We validate PreSTU on eight VQA (ST-VQA [23], TextVQA [237], VizWiz-VQA [88], VQAv2 [83], OCR-VQA [191], DocVQA [186], ChartQA [184], AI2D [131]) and four image captioning (TextCaps [235], VizWiz-Captions [89], WidgetCap [158], Screen2Words [263]) benchmarks. Our OCR-aware objectives SPLITOCR, VQA, and CAP are significantly beneficial. For instance, compared with strong baselines which take OCR signals as input, we observe more than 10% absolute gain on TextVQA and 42 CIDEr point gains on TextCaps (Figure 7.1). Finally, we conduct comprehensive experiments to understand which factors contribute to effective STU pre-training. In summary, our contributions are as follows:

- We propose PreSTU, a simple and effective pre-training recipe with OCR-aware objectives designed for scene-text understanding (§7.3).

[8] This makes our model more robust to the quality of OCR systems.

[9] https://cloud.google.com/vision/docs/ocr



Objective	Text Input	Output
SplitOCR	Generate ocr_text in en: <OCR$_1$> <OCR$_2$>...<OCR$_m$>	<OCR$_{m+1}$>...<OCR$_N$>
VQA	Answer in en: <Question> <OCR$_1$> <OCR$_2$>...<OCR$_N$>	<Answer>
CAP	Generate alt_text in en: <OCR$_1$> <OCR$_2$>...<OCR$_N$>	<Caption>

Figure 7.2: **Our proposed pipeline.** Left: Comparison between PreSTU and NoPreSTU (baseline) we want to compare against. Green denotes the PreSTU pre-training phase and yellow the downstream/fine-tuning phase. SplitOCR encourages scene-text recognition as well as the learning of the connection between scene text and its visual context; VQA and CAP further strengthen that connection. Right: The text input and output for each objective. All objectives utilize OCR signals. See Figure 7.3 for the architecture of PreSTU.

- We show that our objectives consistently lead to improved scene-text understanding on twelve diverse downstream VQA / image captioning tasks (§7.4.1) and even on cases when OCR signals are absent during downstream tasks (§7.4.2).
- We perform detailed analyses to understand the effect of our design choices on STU performance (§7.4.2).

7.2 Prior STU studies

Scene-Text Understanding. Most early STU works [112, 113, 149, 26, 96, 173] have merely focused on Optical Character Recognition (OCR). We instead focus on scene-text understanding (STU) in the context of V&L tasks: VQA [237, 23] and image captioning [235]. The most common approach for these STU tasks is to fuse pre-extracted object detection features with off-the-shelf OCR signals as additional input [237, 104, 235, 22, 90, 125, 265, 281, 183, 147]. These works often focus on specific challenges in downstream STU tasks, including dealing with noisy OCR signals, enabling the generation of rare words, or incorporating geometric information of OCR texts. In contrast, our work focuses on pre-training general-purpose STU models and shows the effectiveness of our objectives on multiple downstream STU tasks (§7.4.1).

V&L Pre-Training for STU. One line of works incorporates OCR signals explicitly for pre-training [284, 22, 181]. TAP proposes an objective to learn the relative spatial position of two OCR texts. LOGOS [181] localizes a region that is most related to a given task and relies on its OCR text to complete the task. LaTr [22] models the co-occurrence statistics of layout-aware OCR tokens. Our pre-training objectives, on the other hand, focus on learning both scene-text recognition and the role of scene-text in its visual context.

The other line of works is OCR-free. Recently, extremely large image-text models have shown promising results on STU tasks, despite having no explicit STU objectives (*e.g.*, GIT2 [264], Flamingo [13]). However, it would require an analysis of their private data and a prohibitive amount of resources to pinpoint what contributes to such strong results. Our study offers a complementary perspective to this OCR-free approach by pushing the limit of the OCR-heavy approach further than before and conducting more thorough experiments at a smaller scale.

7.3 PreSTU: Pre-Training for Scene-Text Understanding

Figure 7.2 provides an overview of PRESTU OCR-aware objectives and their input-output format. In what follows, we first describe our starting point: model architecture and OCR signals (§7.3.1). Then, we describe our recipe for pre-training (§7.3.2), including the objectives, **SPLITOCR**, **VQA**, and **CAP** (§7.3.2.1), and data sources (§7.3.2.2). Finally, we describe the fine-tuning stage and target benchmarks (§7.3.3).

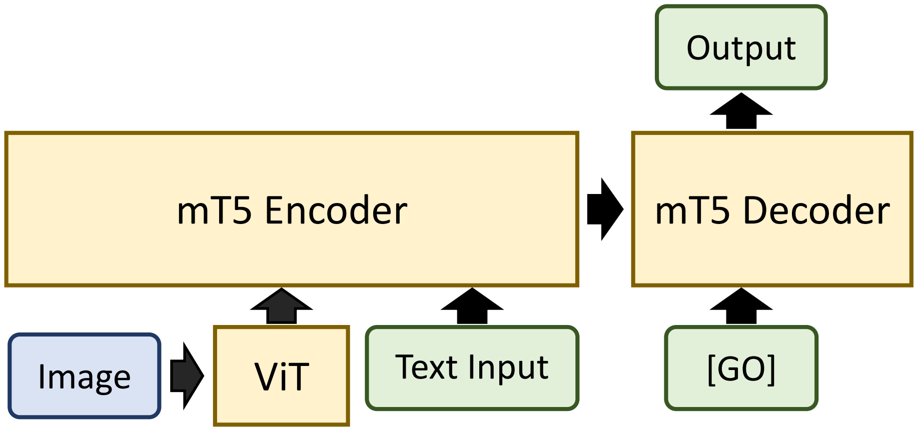


Figure 7.3: **V&L model architecture used in all of our experiments.** We use a simple transformer-based encoder-decoder (pre-trained ViT [64] + mT5 [282]) transforming image and text inputs to the text output. Green box: text input/output. Blue box: visual input. Yellow box: model blocks. See Figure 7.2 for the input-output pairs for different objectives.

7.3.1 Setup

V&L model architecture. Our main architecture is illustrated in Figure 7.3. We start from an encoder-decoder V&L architecture which unifies image-to-text (e.g., image captioning) and image+text-to-text (e.g., VQA) tasks. The pre-trained vision encoder is ViT-B/16 [64], and the pre-trained language encoder-decoder is mT5-Base [282]. Specifically, ViT is a transformer-based encoder that takes a sequence of image patches as input, pre-trained on an image classification task. mT5 is a multilingual variant of text-to-text transformers T5 [209], pre-trained on a massive multilingual text corpus with the span corruption objective. See more details in the supplementary material.

As mentioned in LaTr [22], this starting point leads to modeling advantages over existing model architectures for STU tasks. First, we believe that understanding the role of OCR text in the visual context is much easier from image pixels, making ViT a natural choice. Second, mT5 uses wordpiece vocab to encode and decode text tokens; thus a certain level of robustness to the noise in the input OCR texts comes with it by default. On the other hand, M4C [104] and TAP [284] resort to a more complicated solution of using fastText [24] and Pyramidal Histogram of Characters features [14]. Third, mT5 is an encoder-decoder model which enables to generate the open-ended text. This is suitable for general image captioning and scene-text VQA where the answers tend to be out-of-vocab. In contrast, most prior works [237, 104, 284, 265, 181] treat VQA as answer vocab-based classification. Lastly, our model is built upon well-developed vanilla unimodal building blocks in vision and NLP. We deliberately choose this general encoder-decoder architecture to push for the applicability of our objectives. Such a design choice allows us to develop less model-dependent pre-training objectives.

Image resolution. Unless stated otherwise, we use the image resolution of 640x640 in all of our experiments.

OCR signals. We obtain OCR signals from Google Cloud OCR for all pre-training and downstream datasets in our experiments. They come in the form of a set of texts and their corresponding box coordinates in the image (*i.e.*, object detection-like). We order OCR texts based on their locations, top-left to bottom-right and concatenate them with the T5 separator </s>. This allows models to implicitly learn the scene text's spatial information and standarize the target output sequence during training. Unless stated otherwise, we use these sorted *silver* OCR texts in all of our experiments.

7.3.2 Pre-Training Stage

7.3.2.1 PreSTU Objectives

We consider two sets of OCR-aware pre-training objectives for scene-text understanding.

Task-agnostic objective: SplitOCR. Inspired by the impressive performance of the visual language modeling pre-training objective for image+text-to-text downstream tasks [271], we propose an OCR-aware pre-training objective called SPLITOCR. This objective is designed to be downstream task-agnostic, focusing on teaching the two core capabilities for STU: recognizing scene text and connecting it to the visual context.

We randomly split the OCR texts into two parts and use the first part as additional input and the second part as a target. Recall that we have ordered the OCR texts based on their locations such that the model can recognize them in a consistent manner. Note that if the splitting point is right at the beginning of the OCR sequence, the model performs a simplified version of the traditional Optical Character Recognition task (*i.e.*, predicting the whole OCR tokens). We denote this by OCR in Table 7.6 and also compare it with SPLITOCR in our ablation studies.

Why SPLITOCR? SPLITOCR equips the model with the abilities to recognize scene text and connect it to the visual context in a unified, seamless manner. Specifically, operating SPLITOCR upon the "first part" of OCR tokens and the image pixels (not pre-extracted global or object detection features) and predicting the "second part" of OCR tokens requires the model to (i) identify which scene text in the image *still* needs to be recognized, inherently connecting the input scene text to its visual context; (ii) perform the OCR *task*, inherently acquiring the scene-text recognition skill.

Task-specific objectives: VQA and CAP. We propose OCR-aware downstream-task-specific pre-training objectives on top of SPLITOCR. We consider two objectives based on our downstream tasks: (i) VQA which predicts the target answer from the question prompt, the visual question, and OCR texts and (ii) CAP which predicts the target caption from the caption prompt and OCR texts. This is similar to previous approaches to STU, except that we encode the image pixels, not features from pre-detected regions.

Why VQA or CAP? Task-specific objectives aim to achieve two goals. First, they further encourage the learning of the relationship between scene text and its visual context through direct interaction between input image pixels and input OCR texts. Second, it eases the knowledge transfer from pre-training to fine-tuning since task-specific objectives share the same input format as that of the downstream tasks (§7.3.3). See Figure 7.2 for more details.

7.3.2.2 Pre-Training Data

Our main pre-training data is CC15M, the union of two popular image-text datasets: Conceptual Captions (CC3M) [229] and Conceptual 12M (CC12M) [39].[10] CC3M consists of 3.3M $\langle image, caption \rangle$ pairs, obtained by processing raw alt-text descriptions from the Web. CC12M extends CC3M by relaxing its over-restrictive filtering pipeline. We use CC15M for SPLITOCR and CAP pre-training. Note that the captions of CC15M are not used for SPLITOCR and their images are not necessarily scene text-related. See more details in the supplementary material.

Since CC15M does not have data in the form of visual questions and their answers for us to leverage, we resort to ST-VQA [23]. It is a scene-text VQA dataset whose images are collected from 6 diverse data sources (COCO-Text [260], Visual Genome [141], VizWiz [88], ICDAR [129, 128], ImageNet [59], IIIT-STR [190]). We use its training set for pre-training. We use ST-VQA as pre-training data for other VQA benchmarks as well as a downstream benchmark for testing SPLITOCR (§7.3.3).

[10] Due to expired URLs, only 13M $\langle image, caption \rangle$ pairs are used in our experiments.

7.3.3 Fine-tuning Stage

In all of our downstream scene-text V&L tasks, the input-output pairs follow the same format as either VQA or CAP (with OCR text tokens as input.) The only difference from the task-specific pre-training is the training data.

We validate PreSTU on twelve datasets related to VQA and image captioning tasks. ST-VQA, TextVQA, and TextCaps are the main benchmarks for STU. We also consider other scene-text domains, including book (OCR-VQA), document (DocVQA), illustration (ChartQA), diagram (AI2D), and screenshot domains (WidgetCap and Screen2Words). VizWiz-VQA and VizWiz-Captions are for the blind and heavily involve STU. VQAv2 is a general VQA dataset. See complete details in the supplementary material.

7.3.4 Discussion

We compare PreSTU with two well-known prior STU works TAP [284] and LaTr [22]. In terms of modeling, TAP leverages two conventional V&L objectives: visual-region masked language modeling and image-text matching, as well as the objective of learning the relative spatial position of two OCR text detections. TAP models the image using object-based features [217], which we believe is a suboptimal visual context. Besides, TAP adopts vocab-based classification, less suitable for some STU tasks which are full of out-of-vocab words. LaTr overcomes those weaknesses by adopting a similar V&L architecture to ours (ViT-B/16 / T5_{large}). However, its pre-training objective does not involve the visual component (ViT). Instead, it only pre-trains its language component to learn the co-occurrence statistics of layout-aware OCR tokens. As the visual component is distorted or absent during pre-training, these models do not inherently learn the two essential STU capabilities, and would likely suffer in a case when OCR signals are absent during downstream tasks. In contrast, PRESTU fully embraces the visual component. As shown in §7.4.2, this brings a huge benefit especially when OCR signals are not available. See a more detailed comparison in §7.4.1.4.

In terms of pre-training data, TAP aggregates scene-text *dedicated* downstream data, including ST-VQA, TextVQA, TextCaps, and OCR-CC. Thus, while it aligns well with the corresponding downstream tasks, it is less generalizable to other V&L tasks. In contrast, PRESTU adopts *general* pre-training data (*i.e.*, CC15M), providing a more flexible interface for V&L tasks. Besides, LaTr argues that pre-training on document images is a better choice since acquiring large quantities of natural images with scene text for pre-training is challenging and hard to scale, and the amount of text is often sparse. Our work challenges this assumption and shows that one can pre-train effectively for STU on natural images with minimal preprocessing. (*i.e.*, nothing beyond extracting OCR signals).

Finally, in terms of evaluation as we will show next, our experiments are done on a much wider range of benchmarks than before. This is in stark contrast to existing works which often focus on three benchmarks at most.

Model	Pre-training Objective	Test Benchmark			
		ST-VQA ANLS	TextVQA Acc	VizWiz-VQA Acc	VQAv2 Acc
NOPRESTU	-	56.7	44.8	57.7 / 57.2	74.8 / 75.2
PRESTU	VQA	N/A	48.3	58.3 / 57.6	75.0 / 75.0
	SPLITOCR	**65.5**	55.2	61.9 / 61.3	**76.0 / 76.2**
	SPLITOCR→VQA	N/A	**56.3**	**62.5 / 62.0**	**76.1 / 76.1**

Table 7.1: **Effectiveness of PRESTU objectives on VQA.** Our pre-training objectives (VQA, SPLITOCR, SPLITOCR→VQA) show consistent gains over the baseline on all VQA benchmarks. We use CC15M for SPLITOCR pre-training and ST-VQA for VQA pre-training. Since ST-VQA for VQA pre-training, we mark VQA and SPLITOCR→VQA as "N/A". Results are reported on the test set for ST-VQA, test-std for TextVQA, and test-dev/test-std for VizWiz-VQA and VQAv2.

Model	Pre-training Objective	TextCaps test-std					VizWiz-Captions test-std				
		B	M	R	S	C	B	M	R	S	C
NOPRESTU	-	23.4	21.0	45.0	13.6	96.9	29.4	22.6	49.9	18.5	87.2
PRESTU	CAP	31.6	25.6	51.5	18.7	133.1	33.7	24.5	52.8	20.8	103.1
	SPLITOCR	28.5	23.9	48.9	16.3	126.1	29.8	22.6	50.3	18.6	90.2
	SPLITOCR→CAP	**32.8**	**26.2**	**52.2**	**19.1**	**139.1**	**34.3**	**24.7**	**53.4**	**21.1**	**105.6**

Table 7.2: **Effectiveness of PRESTU objectives on image captioning.** Our pre-training objectives (CAP, SPLITOCR, SPLITOCR→CAP) show significant gains over the baseline on all image captioning benchmarks, with SPLITOCR→CAP performing best. We use CC15M for both SPLITOCR and CAP pre-training. B: BLEU@4, M: METEOR, R: ROUGE-L, S: SPICE, C: CIDEr.

7.4 Experimental Results

Baselines. We denote by NOPRESTU our main baseline. It is the same pre-trained V&L model as PRESTU (*i.e.*, ViT-B/16 / mT5) but *not* pre-trained with any of our pre-training objectives.

Metrics. For VQA tasks, we use standard VQA accuracy following [237, 284, 264]. It is the average score over nine subsets of the ground-truth ten answers, where each score is:

$min(\frac{\#answer\ occurrences}{3}, 1)$. For ST-VQA/DocVQA, we use Average Normalized Levenshtein Similarity (ANLS), *softly* penalizing the model's mistakes on scene-text recognition. For ChartQA, we report its official metric, a relaxed accuracy that allows a minor inaccuracy for numeric answers. For image captioning tasks, we use their standard evaluation metrics, including BLEU [198], METEOR [61], ROUGE-L [162], SPICE [15], and CIDEr [259].

7.4.1 Main Results

The main goal of our experiments is to assess the utility of our pre-training objectives SPLITOCR and VQA/CAP in VQA (§7.4.1.1) and image captioning (§7.4.1.2) tasks.

7.4.1.1 VQA

Table 7.1 summarizes our main results on VQA tasks, including ST-VQA, TextVQA, VizWiz-VQA, and VQAv2. SPLITOCR outperforms the baseline (*i.e.*, without our STU pre-training) by a large margin on scene-text-heavy VQA tasks, more than +8.8 ANLS on ST-VQA, +10.4% on TextVQA, and +4.1% on VizWiz-VQA. With SPLITOCR→VQA, we slightly but significantly improve the performance further on TextVQA and VizWiz-VQA, +1.1% and 0.7%, respectively. These results show the utility and applicability of our pre-training objectives for improving scene-text understanding.

SPLITOCR and VQA are complementary on scene-text-heavy VQA tasks (TextVQA/VizWiz-VQA), where each of them alone underperforms SPLITOCR→VQA. Additionally, we observe the first-stage pre-training via SPLITOCR is more beneficial than the second-stage task-specific pre-training VQA. This could be due to the superiority of SPLITOCR or the lack of large-scale scene-text VQA pre-training data, or both. We identify data development for scene-text VQA as an open research question.

Our results also highlight the importance of STU in general real-world VQA (*i.e.*, not specially designed for STU). We observe a slight but significant improvement over the baseline on VQAv2 and a more significant improvement on VizWiz-VQA for blind people. We attribute this to a subset of questions that require text recognition and reasoning skills [292]. We believe this is an important step since these questions are considered "hard to learn" or even "outliers" that work against VQA algorithms [246, 127].

7.4.1.2 Image Captioning

Table 7.2 summarizes our main results on image captioning tasks, TextCaps and VizWiz-Captions. Aligned with the VQA results, SPLITOCR significantly improves over the baseline across all evaluation metrics, with SPLITOCR→CAP performing best. The gain is notably 42.2 CIDEr points on TextCaps, and 18.4 on VizWiz-Captions. Overall, we highlight the usefulness of SPLITOCR across V&L tasks with different input-output formats.

Similar to the VQA results, SPLITOCR and CAP are complementary. However, CAP alone is more beneficial than SPLITOCR alone. We attribute this to our large-scale web-based image-text data that is already suitable for CAP pre-training. Despite such a strong CAP model, SPLITOCR still provides an additional benefit.

7.4.1.3 Applicability to Other Scene-Text Domains

Unlike prior STU literature [284, 265, 181, 22, 266, 74], we further explore other scene-text domains (Table 7.3). We show that PreSTU is also effective on book (OCR-VQA), document (DocVQA), illustration (ChartQA), diagram (AI2D), and screenshot domains (WidgetCap & Screen2Words). This demonstrates the applicability of PRESTU to many different real-world STU problems.

Model	OCR VQA %Acc	Doc VQA %ANLS	Chart QA %RelaxedAcc	AI2D %Acc	Widget Cap CIDEr	Screen2 Words CIDEr
NoPreSTU	71.5	47.5	40.5	64.5	63.9	98.5
PreSTU-SplitOCR	72.2	50.1	50.7	69.3	125.6	113.8

Table 7.3: **PreSTU on other scene-text domains (Val split).** See §7.4.1.3 for a detailed discussion.

7.4.1.4 Comparison to Prior Works

So far our results provide strong evidence for the benefit of our proposed objectives. In this section, we provide a comparison to prior works as further context. While apples-to-apples comparison has become increasingly difficult, we make our best attempt to analyze our results in the context of these works. For example, TAP's objective has coupled the use of object detection signals, which we do not resort to. More importantly, many prior works [22, 13, 264] do not release code, rely on private data, and/or require too large-scale pre-training that is prohibitively costly to reproduce.

We first compare PRESTU to recent works focusing on STU tasks (Rows Non-TAP to LaTr in Table 7.4). Overall, PRESTU establishes strong results on all tasks. Concretely, PreSTU achieves better results than all prior smaller-scale works (*i.e.*, TAP, TAG, LOGOS). More interestingly, with much less data, we even outperform two larger models ConCap/UniTNT (139.1 vs. 105.6/109.4 in CIDEr) on TextCaps and (56.3% vs. 55.4%) on TextVQA.

PreSTU, however, performs worse than another larger model LaTr on TextVQA/ST-VQA. We attribute this to the superiority of LaTr's V&L backbones. As shown in Table 7.5, LaTr$_{\text{base}}$ with no pre-training significantly outperforms our baseline (NOPRESTU) on TextVQA (52.3% vs. 45.2%). LaTr and PRESTU use different scene-text pre-training data: LaTr uses five times larger data than PRESTU (64M vs. 13M in Table 7.4), which covers more *diverse* scene text. This is particularly beneficial to TextVQA/ST-VQA, which contain scene text from multiple domains (*e.g.*, brand, vehicle, etc.) and may explain why LaTr outperforms PreSTU.

In contrast, OCR-VQA [191] only covers book-related scene text. Thus, pre-training data becomes less important than pre-training approaches, and PreSTU outperforms LaTr (72.2% vs. 67.5% in Table 7.5). Moreover, while LaTr only shows its effectiveness on VQA tasks, PreSTU shows on both VQA and image captioning tasks.

We further compare PreSTU to extremely large-scale V&L models pre-trained on more than 2B $\langle image, text \rangle$ pairs. Interestingly, our best model even outperforms two much larger models Flamingo [13] and GIT2 [264] on some tasks; using much less data, we achieve better results than Flamingo (56.3% vs. 54.1%, Table 7.4) on TextVQA and than GIT2 (72.2% vs. 69.9%, Table 7.5) on OCR-VQA.

Recently, PaLI [45], a large-scale V&L model (ViT-e/mT5-XXL) pre-trained on 10B $\langle image, text \rangle$ pairs, reports SOTA results on all major V&L tasks, except for VizWiz-Captions (Table 7.4). It is worth noting that PreSTU (specifically, our OCR) was an ingredient in the pre-training objective of PaLI to tackle OCR and STU tasks, demonstrating OCR's utility in large-scale SOTA models.

The closest to PreSTU in terms of model/data sizes is $\mathrm{GIT_L}$, a smaller-scale version of GIT2 (347M parameters and 20M $\langle image, text \rangle$ pairs). As shown in Table 7.5, PreSTU outperforms (or is on par with) $\mathrm{GIT_L}$ on all tasks, demonstrating efficiency with respect to model/data sizes. See more comparisons in the supplementary material.

Model	Model Size	Data Size	Pre-training Objective	Test Benchmark					
				TextCaps	VW-Cap	ST-VQA	TextVQA	VW-VQA	VQAv2
				CIDEr	CIDEr	ANLS	Acc	Acc	Acc
NoPreSTU	473M	0	-	96.9	87.2	56.7	44.8	57.2	75.2
PreSTU	473M	13M	SPLITOCR	126.1	90.2	65.5	55.2	61.3	76.2
			SPLITOCR→VQA/CAP	139.1	105.6	N/A	56.3	62.0	76.1
Non-TAP [284]	146M	0	-	93.4	-	51.7	44.8	-	-
TAP [284]		1.5M*	MLM+ITM+RPP	109.7	-	59.7	54.0	-	-
TAG [265]		88K*	MLM+ITM+RPP	-	-	60.2	53.7	-	-
LOGOS [181]		88K*	ROILOCAL	-	-	57.9	51.1	-	-
ConCap [266]	559M	129M	VLM+ITM+ITC	105.6	-	-	-	-	-
UniTNT [74]				109.4	-	66.0	55.4	-	80.1
LaTr [22]	831M	64M	MLM	-	-	69.6	61.6	-	-
Flamingo [13]	80B	2.3B	VLM	-	-	-	54.1	65.4	82.1
GIT2 [264]	5B	12.9B	VLM	145.0	**120.8**	75.8	67.3	70.1	81.9
PaLI [45]†	16B	10B	our OCR w/ others	**160.4**	-	**79.9**	**73.1**	**73.3**	**84.3**

Table 7.4: **Comparison to prior works.** See §7.4.1.4 for a detailed discussion. VW-Cap: VizWiz-Captions, VW-VQA: VizWiz-VQA, MLM: Masked Language (visual region) Modeling, ITM: Image-Text Matching, RPP: Relative Position Prediction, VLM: Visual Language Modeling, ITC: Image-Text Contrastive Loss, ROILOCAL: ROI localization. *: dedicated scene-text understanding data, including ST-VQA, TextVQA, TextCaps, and OCR-CC. †: our objective OCR is an ingredient in their pre-training objectives.

7.4.2 Analysis

We aim to understand PreSTU in detail. We show (a) the importance of different components of our design choice, (b) its zero-shot transferability, (c) the effect of pre-training image resolution, (d) the effect of pre-training data size, and (e) the effect of downstream OCR quality.

Detailed ablation. As shown in Figure 7.2, our PreSTU consists of two (optional) pre-training stages, followed by fine-tuning on downstream tasks. Here, we aim to understand the gain brought by each component. We consider different combinations of the design choices at each stage and organize the results stage-by-stage into Table 7.6. We have the following three major observations.

Model	Model Size	Data Size	Pre-training Objective	Val or test-dev Benchmark					
				TextCaps	ST-VQA	TextVQA	VW-VQA	VQAv2	OCR-VQA
				CIDEr	ANLS	Acc	Acc	Acc	Acc
NoPreSTU	473M	0	-	100.0	55.6	45.2	57.7	74.8	71.5
PreSTU	473M	13M	SPLITOCR	134.6	**62.7**	55.6	61.9	76.0	**72.2**
			SPLITOCR→VQA/CAP	**141.7**	N/A	**56.7**	**62.5**	**76.1**	-
$\text{LaTr}_{\text{base}}$ [22]	281M	0	-	-	-	52.3	-	-	-
$\text{LaTr}_{\text{base}}$ [22]	281M	64M	MLM	-	67.5	58.0	-	-	67.5
GIT_{L} [264]	347M	20M	VLM	106.3	44.6	37.5	**62.5**	75.5	62.4
GIT2 [264]	5B	12.9B	VLM	148.6	75.1	68.4	71.0	81.7	69.9

Table 7.5: **Comparison to GIT_{L} (similar model/data sizes to PreSTU).** PreSTU outperforms (or is on par with) GIT_{L} on all tasks. GIT2/$\text{LaTr}_{\text{base}}$-64M are for reference to show that PreSTU even outperforms these large-scale works on OCR-VQA.

First, SPLITOCR is significantly and consistently better than OCR (Rows with SPLITOCR vs. Rows with OCR in their Stage-1). OCR is a **"pure"** OCR prediction task, a variant of our main SPLITOCR (OCR-conditioned OCR prediction) in which the splitting point is always at the beginning. At first glance, such a result may seem counterintuitive: predicting the entire scene text is strictly harder than predicting part of the OCR text given the other part. When thought of carefully, this result indicates that OCR may put too much emphasis on *recognizing* scene text, at the expense of *connecting* scene text to its visual context. In other words, this highlights how SPLITOCR is able to balance the two capabilities that we identify as important for STU (§7.1).

Second, SPLITOCR (or OCR) makes the visual component (ViT) *inherently* better at recognizing text (gap between "Yes" and "No" Rows with Stage-1 pre-training vs. gap between "Yes" and "No" Rows without Stage-1 pre-training). Without Stage-1 (*e.g.*, VQA/CAP), removing OCR signals during fine-tuning leads to more than a 33% drop on TextVQA and a 49 CIDEr point drop on TextCaps. With Stage-1, these drops become less than 17%

and 26 CIDEr points, respectively. For TextCaps, SPLITOCR with “No” OCR input tokens during fine-tuning even outperforms the baseline *with* OCR input (116.6 vs. 100.0 in CIDEr). In summary, *recognizing* scene text via Stage-1 pre-training is important (*i.e.*, cannot be achieved via VQA or CAP alone).

Third, having two sources of OCR signals is beneficial. OCR signals by pre-trained ViT (Row SPLITOCR→VQA/CAP with “No”) and OCR signals by the off-the-shelf system (Row NOPRESTU "Yes") are complementary; we achieve the best result when leveraging both OCR signal sources (Row SPLITOCR→VQA/CAP with “Yes”). See more ablation studies in the supplementary material.

Zero-shot transferability on scene-text VQA. Table 7.7 shows zero-shot transferability of SPLITOCR on TextVQA. We observe that performing SPLITOCR and then fine-tuning on ST-VQA (SPLITOCR→VQA) already leads to a strong model; SPLITOCR→VQA *without* fine-tuning (44.3%) is competitive to NOPRESTU *with* fine-tuning on TextVQA training set (45.2%), while ST-VQA alone (VQA) only achieves 35.7%. This suggests that SPLITOCR enables generalization for STU and may remove the need to collect TextVQA data entirely!

Effect of image resolutions during pre-training. We hypothesize that pre-training with high-resolution images is important for scene-text recognition; Table 7.8 supports this argument. Further, pre-training with the 224x224 image resolution (standard resolution for many vision tasks) almost does not help; it achieves the accuracy of 47.1%, close to 45.2% of NOPRESTU baseline (Table 7.6 Row 2), suggesting non-standard resolution must be considered to reap the benefit of STU pre-training.

Effect of pre-training data scale. How much data do we need to learn to recognize text? Table 7.9 shows the performance of TextVQA given checkpoints pre-trained on 1%, 3%, 10%, and 30% subsets of CC15M. We find that the TextVQA performance goes up as more

Pre-training Stage-1	 Stage-2	Fine-tuning OCR input	TextVQA Val Acc	TextCaps Val CIDEr
-	-	No	19.5	40.1
		Yes	45.2	100.0
-	VQA/CAP	No	13.7	81.1
		Yes	47.2	130.2
OCR	-	No	35.8	110.4
		Yes	49.9	126.7
OCR	VQA/CAP	No	38.6	108.9
		Yes	51.9	134.4
SPLITOCR	-	No	39.4	116.6
		Yes	55.6	134.6
SPLITOCR	VQA/CAP	No	44.3	118.4
		Yes	**56.7**	**141.7**

Table 7.6: **Main ablation studies** for validating the importance of our main components: SPLITOCR, VQA/CAP, and having OCR input during fine-tuning. See §7.4.2 for a detailed discussion. OCR refers to predicting the entire OCR text.

pre-training data is included. This highlights the importance of data scale in acquiring *transferable* scene-text recognition skills.

Effect of downstream OCR systems. We study the effect of different OCR systems during fine-tuning (Table 7.10). We observe that the SPLITOCR-pre-trained model is more robust to the change in downstream OCR systems than NOPRESTU. Indeed, SPLITOCR + Rosetta can even perform better than NOPRESTU + gOCR. This result is consistent with Table 7.6, where we experiment with removing OCR texts entirely during fine-tuning. We also find that gOCR is the most effective. Interestingly, it is even better than human-annotated TextOCR; we hypothesize this is because TextOCR only provides word-level annotation whereas gOCR provides some grouping.

Model	Pre-training Objective	Fine-tuning	TextVQA Val Acc
NoPreSTU	-	-	0.04
		TextVQA	45.2
PreSTU	VQA	-	35.7
	SPLITOCR→VQA	-	44.3

Table 7.7: **Zero-shot transferability on TextVQA.** Our zero-shot SPLITOCR→VQA (*without* fine-tuning on TextVQA) is competitive to supervised NoPreSTU (*with* fine-tuning on TextVQA).

Model	Pre-training		Fine-tuning	TextVQA
	Objective	Resolution	Resolution	Val Acc
PreSTU	SPLITOCR	224	640	47.1
		384		50.2
		480		53.1
		640		**55.6**

Table 7.8: **Effects of image resolutions.** TextVQA accuracy goes up as the pre-training image resolution increases, emphasizing the necessity of high-resolution images during pre-training.

Model	Pre-training			TextVQA
	Objective	Proportion	# of Data	Val Acc
PreSTU	SPLITOCR	1%	130K	42.3
		3%	390K	45.4
		10%	1.3M	50.6
		30%	3.9M	53.0
		100%	13M	**55.6**

Table 7.9: **Importance of pre-training data scale.** TextVQA performance improves as more pre-training data, showing the importance of data scale in learning *transferable* scene-text recognition.

Model	Pre-training Objective	Fine-tuning OCR System	TextVQA Val Acc
NoPreSTU	-	TextOCR [235]	44.0
		Rosetta [26]	36.7
		gOCR	45.2
PreSTU	SPLITOCR	TextOCR [235]	54.8
		Rosetta [26]	50.7
		gOCR	**55.6**

Table 7.10: **Effect of downstream OCR systems on TextVQA.** SPLITOCR makes the model more robust to the change in OCR systems during fine-tuning.

7.5 Summary

We introduce a simple recipe for scene-text understanding, consisting of OCR-aware pre-training objectives operating from image pixels. Our task-agnostic objective SPLITOCR teaches the model to recognize scene text and to connect scene text to its visual context. Our task-specific objectives VQA and CAP further strengthen that connection. We conduct comprehensive experiments to demonstrate the utility of this recipe.

Part VI: Conclusion

Chapter 8: Conclusion

In this dissertation, we advance V&L systems through the lens of data. We explore three aspects of V&L data to achieve an integrated understanding of visual and linguistic content.

V&L Data Curation. We explore a novel data augmentation method to synthesize V&L data and enrich resources for model training. This leads to substantial gains in model performance without the significant costs commonly incurred through manual annotation. Additionally, we benchmark the current leading V&L models, evaluating their ability to compare objects, scenes, and situations, and identify notable deficiencies. Our benchmark not only highlights these limitations but also establishes a solid foundation for future enhancements in the comparative capabilities of V&L models.

V&L Data Representations. We propose new V&L representations to enhance alignment between images and text. We encode detailed visual information (extracted from the document) into semantic representations, facilitating alignment with visual features for zero-shot image classification. Similarly, we incorporate visual neighboring information from screenshots into HTML representations to improve web navigation.

V&L Data Learning. We introduce a new pre-training learning recipe that encourages V&L models to recognize text from an image and connect it to the rest of the image content. This results in significant performance gains in various visual question answering and image captioning tasks.

8.0.1 Future for V&L

We are living in the era of multimodal large language models (MLLMs), which demonstrate remarkable performance on multiple V&L tasks. Some express concerns about the future of research, given the high level of performance already achieved. However, I believe that they have not yet reached artificial general intelligence (AGI), and there is still much to explore. Here, I suggest three different routes to further advance V&L systems.

V&L benchmarks. Several recent works [288, 60, 170, 178] focus on curating new V&L benchmarks that could challenge the current leading MLLMs. For instance, MMMU [288] introduces a V&L benchmark designed to assess MLLMs on extensive multi-discipline tasks that require college-level subject knowledge. Similarly, Mind2Web [60] evaluates MLLMs as web agents that are tasked to complete web instructions. All these works find that there is still a noticeable performance gap between humans and MLLMs. This research direction (*i.e.*, buliding V&L benchmarks) thus helps identify the capabilities that current MLLMs are missing or need to improve, thereby advancing V&L systems further.

Capabilities of V&L models. Multiple capabilities have already been reported as weaknesses in current V&L models. As mentioned in chapter 4, these models face pronounced challenges in comparison tasks such as capturing the fine-grained difference between two similar images or understanding spatial proximity and quantity relativity. Furthermore, prior studies [289, 174, 123, 25, 167, 105, 251, 195] highlighted their limited abilities in finding a correct caption among similar captions for an image; understanding negation in the prompt; being robust against (visual) hallucinations; solving OCR-related V&L tasks; understanding spatial relationships among visual objects; counting objects in the image. Thus, our future research could focus on enhancing these capabilities to build more advanced V&L systems.

V&L + X. V&L has broadened its scope from the traditional combination of image and text to include additional modalities, referred to as "Image + Text + X". For instance, embodied AI [234] and web navigation [60] requires models to comprehend an agent's prior movements (*i.e.*, trajectories) beyond the current scene and instruction. Text-to-video generation [29] and video question answering [300] emphasize the need to process sequences of images (frames), demanding a deeper understanding of object motion and temporal localization. The integration of audio into V&L [10] involves the joint learning of raw signals with images and text. All these works, which introduce new dimensions beyond V&L, bring fresh challenges and opportunities that are worth exploring in the future.

Appendix A: Enriching V&L training materials with data augmentation

In this appendix, we provide details and results omitted in chapter 3.

A.1 Additional Implementation Details

A.1.1 Baseline VQA models.

We validate SIMPLEAUG with three VQA models in our experiments: Bottom-Up Top-Down (UpDn)[11] [16], Learned-Mixin+H (LMH)[12] [53], and LXMERT[13] [247]. All baseline models are implemented using officially released codebase. More details of code and data are publicly available at `https://github.com/heendung/simpleAUG`.

A.1.2 Optimization

UpDn and LMH. We maintain the default settings in UpDn and LMH except for using the mini-batch size of 512 on VQA v2 and $1,024$ on VQA-CP v2. Following the official implementations, our visual features are the output of Faster R-CNN [218] object detector trained on Visual Genome [141], provided by [16]. We optimize UpDn and LMH using stochastic gradient descent (SGD) with Adamax [135] and learning rate 2×10^{-4}. Training

[11] UpDn model implementation: `https://github.com/yanxinzju/CSS-VQA`.

[12] LMH model implementation: `https://github.com/chrisc36/bottom-up-attention-vqa`.

[13] LXMERT model implementation: `https://github.com/airsplay/lxmert`.

a baseline UpDn or LMH model on a single NVIDIA RTX A6000 takes around 2 hours for convergence.

LXMERT. Similar to UpDn and LMH models, LXMERT leverages the object features from the Faster R-CNN detection provided by [16]. We train a LXMERT model using the mini-batch size of 256. Following [247], we use Adam [135] as the optimizer with a linear decayed learning rate schedule. Training a baseline LXMERT model on a single NVIDIA RTX A6000 takes around 8 hours for convergence.

Multi-stage training. As discussed in § 3.4.2 and § 3.4.4 of the main paper, we train the VQA models with a three-stage paradigm ($\mathscr{O} \rightarrow \mathscr{A} \rightarrow \mathscr{O}$): first with original triplets $\mathscr{O}$, then with the SimpleAug triplets $\mathscr{A}$, and then with $\mathscr{O}$ again. *In each of these three stages, we follow the same optimization procedures as we train the baseline VQA models in the first stage.* We report the best results on VQA v2 validation set [82] and VQA-CP v2 test set [6].

A.1.3 Additional details of SimpleAug

As mentioned in the main paper, for each annotated IQA triplet (i, q, a) in the dataset, SimpleAug propagates q to other relevant images. To begin with, we find unique questions by filtering out any duplicate sentences. We then extract meaningful words from the unique questions in line with [42, 275]. Concretely, we remove the question type from q and then apply a spaCy part-of-speech (POS) tagger [100] to extract "nouns". To handle the synonyms, we further consider the singular/plural forms and super-categories of nouns[14]. Moreover, we remove non-informative words (*e.g.*, "picture" or "photo") in the sentence. For example, in a question, "What is the man doing in the picture?", the word "picture" refers to an image itself but not any specific object. There are around 8% of triplets like

[14] Paraphrase database [73] or WordNet [189] could be used to handle other synonyms.

this in VQA v2 [82]. While it is possible that both sentence and image may contain such non-informative contents (*e.g.*, “How many pictures on the wall?”), there are $<1\%$ such questions.

A.2 Results on GQA Dataset

We further conduct a preliminary study of SIMPLEAUG on the popular GQA dataset [107], which focuses on compositional VQA tasks and consists of 22M questions about various day-to-day images. Each image in GQA is associated with a scene graph [121] which consists of the objects, attributes, and relationships.

We focus on binary questions (35% of all questions) and propagate a question q to an image i according to the image’s scene graph. Particularly, we leverage the semantic type of the question (*e.g.*, “attribute”, “relation”) and the scene graph to generate the answer. For example, suppose q asks if an object contains a certain “attribute”, we check the scene graph’s node of that object to determine the answer. SIMPLEAUG can improve the accuracy of UpDn from 56.06% to 56.52%, justifying its generalizability and applicability.

A.3 Human Evaluation on SIMPLEAUG Triplets

SIMPLEAUG requires no sentence/image generation steps, and thus all examples are natural annotations from humans, largely alleviating the artificial noise that the previous methods may have. To further evaluate the quality of the augmented triplets, we randomly select 500 images and pick 5 augmented QA pairs per image from each type (4 by propagation Y/N, Num, Other, Color and 1 by paraphrasing). For those $2,500$ triplets, we ask 5 different crowd workers to evaluate “relatedness $(1/0)$” of the augmented questions and “correctness $(1/0)$” of the answer given the question and image. That is, if the question

Table A.1: **Human evaluation.** The Relatedness and Correctness are shown on different types / question types.

SIMPLEAUG	Type	Relatedness (%)	Correctness (%)
Propagation	Y/N	82.60	52.20
	Color	87.20	77.20
	Num	89.20	60.80
	Other	88.00	80.20
	Overall	86.75	67.60
Paraphrasing	Overall	80.80	64.40

makes sense for the corresponding image, rate 1, otherwise 0; if the answer is correct, rate 1, otherwise 0 (see Figure A.1). Table A.1 shows the human study results. The average relatedness / correctness are 86.75% / 67.60% for propagation and 80.80% / 64.40% for paraphrasing.

We note that these generated data are based on human-annotated questions in the dataset. Therefore, there are no artifacts in the questions. Moreover, these generated data are to augment the original data. Thus, even if they contain noise, they can consistently improve the model's performance.

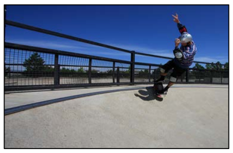
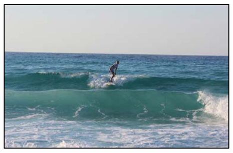
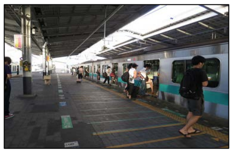
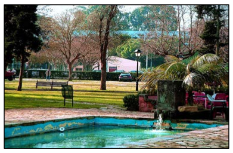
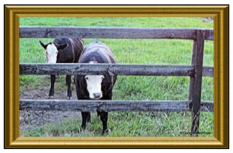
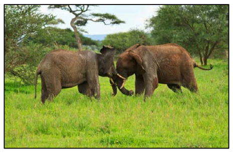
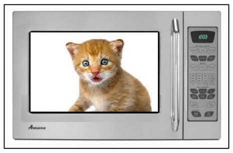
Image	Type	Question	Answer	Relatedness (1 / 0)	Correctness (1 / 0)
	*Y/N* *Color* *Num* *Other* *Paraphrasing*	• *Is the ramp completely covered in snow?* • *What color is his shirt?* • *How many people are standing on there surfboard?* • *What sport is depicted?* • *Is this man wearing any safety gear?*	*No* *Blue* *1* *Skateboarding* *yes*		
	*Y/N* *Color* *Num* *Other* *Paraphrasing*	• *Are there waves shown in this picture?* • *What color is the sky?* • *How many people gathered around?* • *What sport is represented in this scene?* • *What is the man walking on?*	*Yes* *Blue* *1* *Surfing* *Surfboard*		
	*Y/N* *Color* *Num* *Other* *Paraphrasing*	• *Is this photo pulling into a station?* • *What color is his shirt?* • *How many trains can you see?* • *What vehicle is shown?* • *Is the boy walking?*	*Yes* *Black* *2* *Train* *Yes*		
	*Y/N* *Color* *Num* *Other* *Paraphrasing*	• *Is the water clean and safe?* • *What is the color of the grass?* • *How many people might live here?* • *What are the yellow vehicle?* • *Is that the ocean?*	*Yes* *Green* *3* *Car* *No*		
	*Y/N* *Color* *Num* *Other* *Paraphrasing*	• *Are these sheep marked?* • *What is the color of the grass?* • *How many cows are stacked?* • *Which animal is it?* • *What color is the photo frame?*	*Yes* *Green* *2* *Cow* *Yellow*		
	*Y/N* *Color* *Num* *Other* *Paraphrasing*	• *Is the elephant crying?* • *What is the color of the grass?* • *How many animal is there in the picture?* • *What type of animal is behind them?* • *Which elephant is bigger?*	*Yes* *Green* *2* *Elephant* *Right*		
	*Y/N* *Color* *Num* *Other* *Paraphrasing*	• *Is it on display?* • *What color is the button?* • *How many cats are in the picture?* • *What kind of animal is pictured?* • *What color are the cat's eyes?*	*Yes* *Black* *1* *Cat* *Blue*		

Figure A.1: **Examples in human study.** For each image, we pick 5 triplets created by SIMPLEAUG (4 by propagation Y/N, Num, Other, Color and 1 by paraphrasing) and ask crowd workers to evaluate the IQA triplets by the question's *relatedness (1 / 0)* to the image and the answer's *correctness (1 / 0)* to the image and question.

Appendix B: Curating data for a V&L benchmark

In this appendix, we provide details and results omitted in chapter 4.

All codes, data, and instructions for our COMPBENCH can be found in `https://github.com/RaptorMai/CompBenchReview`. COMPBENCH is released under a Creative Commons Attribution 4.0 License (CC BY 4.0).

B.1 Discussions

B.1.1 Limitations

While we conducted a human evaluation study to establish the upper bound performance on COMPBENCH, the study is currently limited to 140 samples assessed by five evaluators. We plan to expand the study to a larger scale in future work.

B.1.2 Social impacts

COMPBENCH evaluates the comparative reasoning abilities of MLLMs in images. A potential negative impact of our work is that malicious users might exploit our concept (*i.e.*, comparison) to compare ethical or offensive content. Therefore, it is essential to incorporate effective safeguards in MLLMs to filter out any inappropriate materials.

Public Dataset	License
MIT-States [111]	N/A
Fashionpedia [115]	CC BY 4.0
VAW [205]	Adobe Research License
CUB-200-2011 [261]	CC BY
Wildfish++ [308]	N/A
MagicBrush [296]	CC BY 4.0
Spot-the-diff [114]	N/A
CelebA [175]	Research-only, non-commercial
FER-2013 [80]	N/A
SoccerNet [77]	MIT License
CompCars [283]	Research-only, non-commercial
NYU-Depth V2 [236]	N/A
VQAv2 [81]	CC BY 4.0
Q-Bench2 [297]	N/A

Table B.1: **License of Assets**.

B.1.3 Ethical considerations

All fourteen datasets that we used to curate CompBench adhere to strict guidelines to exclude any harmful, unethical, or offensive content. Additionally, we instruct human annotators to avoid generating any personally identifiable information or offensive content during our annotation process. Finally, we do not conduct any study to compare harmful, ethical, or offensive content between the two images.

B.1.4 License of assets

All fourteen datasets are publicly available, and Table B.1 details the licensing information for the assets in each dataset. We release our CompBench under a Creative Commons Attribution 4.0 License (CC BY 4.0) to enhance global accessibility and foster innovation and collaboration in research.

B.2 COMPBENCH Curation Details

B.2.1 Annotation Details

We create UI interfaces for annotation using Python in Jupyter Notebook and store the annotations in JSON files. In the following sections, we provide detailed descriptions of the annotation process for each dataset, which are omitted in the main text.

MagicBrush [296] is a large-scale, manually annotated dataset for instruction-guided real image editing. For each image, MagicBrush utilizes DALL-E 2 [211] to generate an edited version of the image based on language instructions, such as "let the flowers in the vase be blue." Our goal is to identify pairs of similar images. We thus use CLIP [207] to evaluate the visual similarity between the original and edited images. Only pairs exceeding a predetermined similarity threshold are selected as candidate samples for our COMPBENCH. For each selected pair, we then construct a multiple-choice question to ask the difference between two images in the pairs. Concretely, we first use GPT-4V [2] to extract all relevant objects and their attributes from the edited image with the following prompt:

> "Please extract as many components as possible from the provided images. The following examples illustrate some potential components, but the list is not exhaustive. Only provide the component names, separated by commas. If a human or an animal is shown in the images and features such as hair, eyes, hands, mouth, ears, and legs are visible, ensure to include them. Similarly, try to identify all components in as much detail as possible.
>
> Examples of components: leg, eye, ear, food, pillow, flower, plate, window, door, chair, dining table, sofa, banana, bowl, sugar, blender, berry, lizard, watermelon, motorcycle, apple, curtain, cookies, cake, hair, hat, dresses, bacon, butter, jam, bread, surfboard, t-shirt, pants, hands, fridge, plants, cabinet, sink, car, girl, boy."

We treat objects and their attributes (if found) as options for the questions. However, GPT-4V [2] may not capture all relevant objects (options) in the images. We thus request human annotators to add as many relevant options as possible. Finally, annotators are required to select the obvious difference between two images as the correct answer among options and verify the quality of the generated samples (Figure B.1).

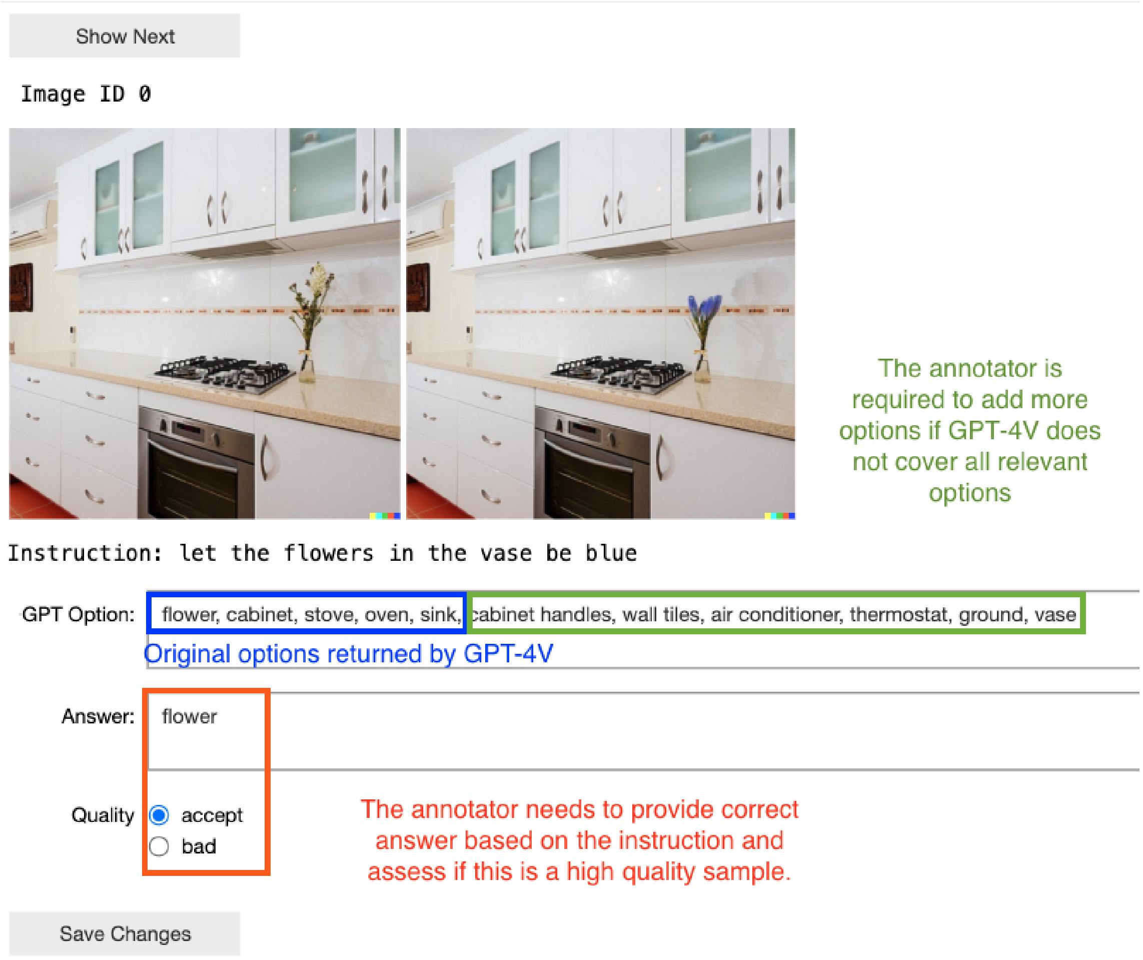


Image ID 0

The annotator is required to add more options if GPT-4V does not cover all relevant options

Instruction: let the flowers in the vase be blue

GPT Option: flower, cabinet, stove, oven, sink, cabinet handles, wall tiles, air conditioner, thermostat, ground, vase

Original options returned by GPT-4V

Answer: flower

Quality ● accept ○ bad

The annotator needs to provide correct answer based on the instruction and assess if this is a high quality sample.

Figure B.1: **Annotation Interface for MagicBrush.**

Spot-the-diff [114] offers video-surveillance image pairs from outdoor scenes, along with descriptions and pixel-level masks of their differences. Similar to MagicBrush, we aim to construct a multiple-choice question to find the obvious difference between the two images. We first prompt the text-only GPT-4 to extract the potentially correct objects from the descriptions of the differences using the following prompt:

> "These sentences describe the differences between the two images. Extract the objects from these sentences. for example, ["there are more people", "the car moved"], you should return "people, car". Please only provide the answer without any explanation and separate the answer names by commas."

Given the extracted objects and the images, GPT-4V is tasked with finding relevant options in the images based on the following prompt:

> "Please list all the objects and attributes associated with the image, for example, black cars, people, trees, white trucks, and yellow poles. Only provide one attribute (adjective) per object. Please only provide the answer without any explanation and separate the answer names with commas. Ensure to include these objects: [OBJECTS FROM LAST STEP]"

We then instruct human annotators to include additional options (if necessary) and identify the most evident difference between two images from the available options as the correct answer (Figure B.2).

MIT-States [111] includes 245 objects with 115 visual attributes or states from online sources such as food or device websites. Each folder in this dataset is named by (adjective, noun), *e.g.*, tall tree, where the adjective describes the state or the attributes and the noun is the object. All the images in this folder share the same adjective and noun. We apply rule-based approaches to generate questions about relative degrees of attributes or states between objects (e.g., "Which tree is taller?"). We then present the questions with the corresponding images in this folder to annotators. The annotators are tasked to select pairs from all the images, label the correct answers (binary: left/right), and filter out any irrelevant or nonsensical questions about the images. In addition, the annotators are required to determine the attribute or state types by selecting from the following options: Size, Color, Texture, Shape, Pattern, State, or None. We filter out examples where the type or answer is None. The annotation UI interface is shown in Figure B.3.

VAW [205] provides a large-scale collection of 620 unique attributes, including color, shape, and texture. We process VAW in the same manner as MIT-States, as detailed in Figure B.3.

CUB-200-2011 [261] catalogs 15 bird parts and their attributes (e.g., “notched tail”). We group images by species with the same attributes (e.g., “curved bill”) and extract visually similar image pairs from each group. We then prompt GPT-4 to transform visual attributes into questions that compare them using the following in-context prompt:

> “I want to turn some text describing the attributes of birds into a question comparing these attributes between birds in two different images. Here are some examples: Attribute: has_bill_shape::hooked, Questions: Which bird has a more hooked bill? Attribute: has_crown_color::brown, Questions: Which bird has more brown on its crown?
>
> Please turn this list of attributes into these questions in this format or style. I want a dictionary format output. [ATTRIBUTE LIST]”

The annotators receive all images in each group along with corresponding comparative questions generated by GPT-4. They are asked to select the pairs from the images and label the correct answers (binary: left/right). The annotation interface is shown in Figure B.4.

Wildfish++ [308] details 22 characteristics (e.g., “brown pelvic fins”) of various fish species and provides detailed descriptions of the differences between two visually similar species. Using the characteristics and the descriptions of difference, we first ask annotators to generate comparative questions (*e.g.*, “Which fish has lighter brown pelvic fins?”). Subsequently, we pass all images from the two similar species along with the corresponding question to the annotators. They select one image from each group to form a pair and label the correct answers as either left or right (Figure B.5).

Fashionpedia [115] is tailored to clothing and accessories and contains 27 types of apparel along with 294 detailed attributes. We group images by (attribute, type), *e.g.*, square neckline. We apply rule-based approaches to generate questions about relative degrees of attributes (*e.g.*, “Which neckline is more square?”) for each group. We then present images of the same type with different attributes, such as “square neckline” and “oval neckline” to the annotators. The annotators are required to select one image from each group to form a pair, choose one between questions from two attributes, and label the correct answer (binary: left/right). The annotation UI interface is shown in Figure B.6.

NYU-Depth V2 [236] features indoor scenes with object segments and depths. Using the segmentation maps, we identify objects within each image and group images containing the same objects. We apply rule-based approaches to generate questions about spatial relative comparisons (*e.g.*, “Which [OBJECT] is closer to the camera?”). The annotator needs to select pairs from all the images in the same group and label the correct answers either left or right (Figure B.7).

CelebA [175] is a large-scale facial attributes dataset featuring over 200K celebrity images, each annotated with 40 attributes. We focus on images labeled with the “smiling” attribute, as it is the only attribute related to the emotion in the dataset. We generate a comparative question such as “Which person smiles more?”. The annotators are tasked with selecting pairs from all images with the smiling attribute and labeling the correct answers either left or right (Figure B.8).

FER-2013 [80] contains grayscale images along with categories describing the emotion of the person, including Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. We leverage rule-based approaches to generate questions about relative emotional comparisons (*e.g.*, “Which person looks more [EMOTIONAL ADJECTIVE]?”). The annotators are required to select pairs from images that share the same emotional attribute and determine the correct answers as either left or right (Figure B.9).

SoccerNet [77], CompCars [283], VQAv2 [81], Q-bench2 [297] are automatically processed to generate samples for COMPBENCH using their metadata and CLIP visual similarity.

B.2.2 Language Prompts for MLLMs

Table B.2 summarizes our language prompts for evaluating MLLMs. We observe that in the case of SoccerNet [77], Gemini1.0-pro [251] always predicts the answer “Left” for binary questions (*e.g.*, “These are two frames related to [SOCCER_ACTION] in a soccer match. Which frame happens first? Please only return one option from (Left, Right) without any other words.”). We thus prompted the Gemini to answer open-ended questions (as shown in Table B.2) instead. We then task human evaluators with verifying whether its responses (*i.e.*, textual descriptions) match the ground-truth answers to calculate its performance. For a fair comparison, we apply the same open-ended questions to other models (*i.e.*, GPT-4V [2], LLaVA-1.6 [169], VILA-1.5 [163]) and report their accuracies.

B.2.3 Model Evaluation

We use official APIs to evaluate proprietary MLLMs, GPT-4V [2] and Gemini [251]. For GPT-4V, we use the version of gpt-4-turbo[15]. For Gemini, we use the Gemini1.0

[15] https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4

Pro Vision[16]. For open source models such as LLaVa-1.6-34b [169][17] and VILA-1.5-40b [163][18], we utilize their official source codes and conduct inference on NVIDIA RTX 6000 Ada GPUs.

B.2.4 Human Annotators & Evaluators

We recruited five in-house human annotators from our research team to work on COMPBENCH. The annotators are instructed to avoid generating any personally identifiable information or offensive content during the annotation process. Furthermore, we recruited another five human evaluators, who were not involved in the annotation, to measure the upper bound performance on COMPBENCH. The workloads for annotation and evaluation were distributed equally among annotators and evaluators.

[16] https://ai.google.dev/gemini-api/docs/models/gemini#gemini-1.0-pro-vision

[17] https://github.com/haotian-liu/LLaVA

[18] https://github.com/Efficient-Large-Model/VILA

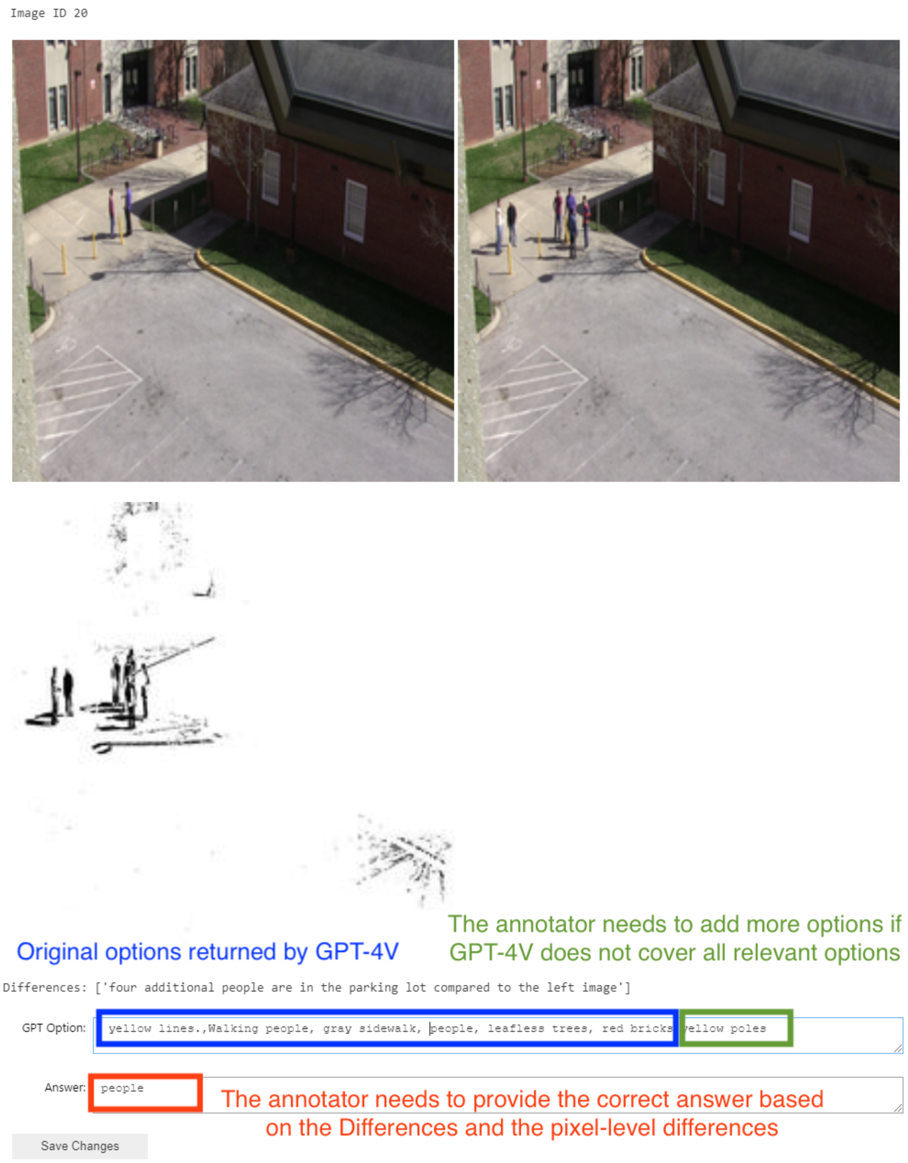


Image ID 20

Original options returned by GPT-4V

The annotator needs to add more options if GPT-4V does not cover all relevant options

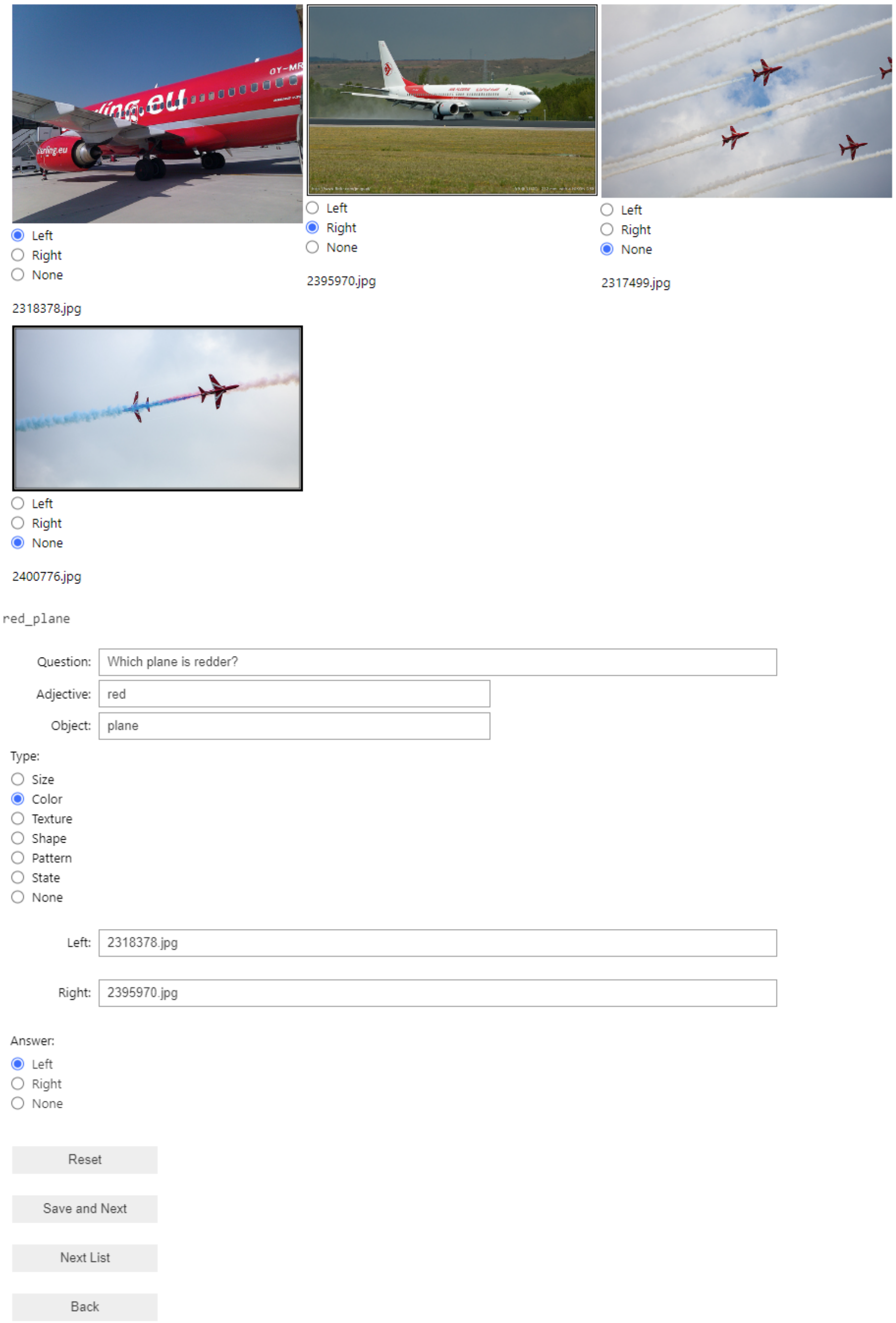
Differences: ['four additional people are in the parking lot compared to the left image']

GPT Option: yellow lines.,Walking people, gray sidewalk, people, leafless trees, red bricks yellow poles

Answer: people

The annotator needs to provide the correct answer based on the Differences and the pixel-level differences

Figure B.2: **Annotation Interface for Spot-the-diff.**



● Left
○ Right
○ None

2318378.jpg

○ Left
● Right
○ None

2395970.jpg

○ Left
○ Right
● None

2317499.jpg

○ Left
○ Right
● None

2400776.jpg

red_plane

Question: Which plane is redder?

Adjective: red

Object: plane

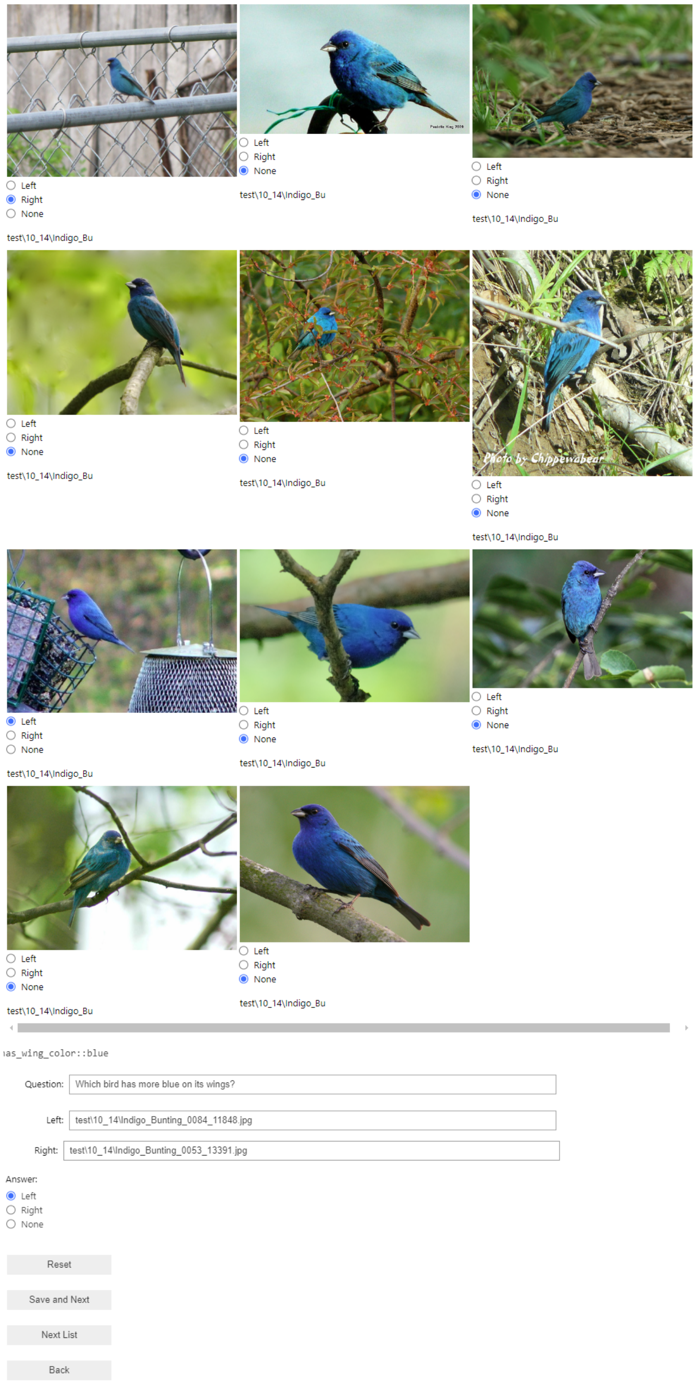
Type:
○ Size
● Color
○ Texture
○ Shape
○ Pattern
○ State
○ None

Left: 2318378.jpg

Right: 2395970.jpg

Answer:
● Left
○ Right
○ None

Figure B.3: **Annotation Interface for MIT-States and VAW.**



○ Left
● Right
○ None

test\10_14\Indigo_Bu

○ Left
○ Right
● None

test\10_14\Indigo_Bu

○ Left
○ Right
● None

test\10_14\Indigo_Bu

○ Left
○ Right
● None

test\10_14\Indigo_Bu

○ Left
○ Right
● None

test\10_14\Indigo_Bu

○ Left
○ Right
● None

test\10_14\Indigo_Bu

● Left
○ Right
○ None

test\10_14\Indigo_Bu

○ Left
○ Right
● None

test\10_14\Indigo_Bu

○ Left
○ Right
● None

test\10_14\Indigo_Bu

○ Left
○ Right
● None

test\10_14\Indigo_Bu

○ Left
○ Right
● None

test\10_14\Indigo_Bu

```
ıas_wing_color::blue
```

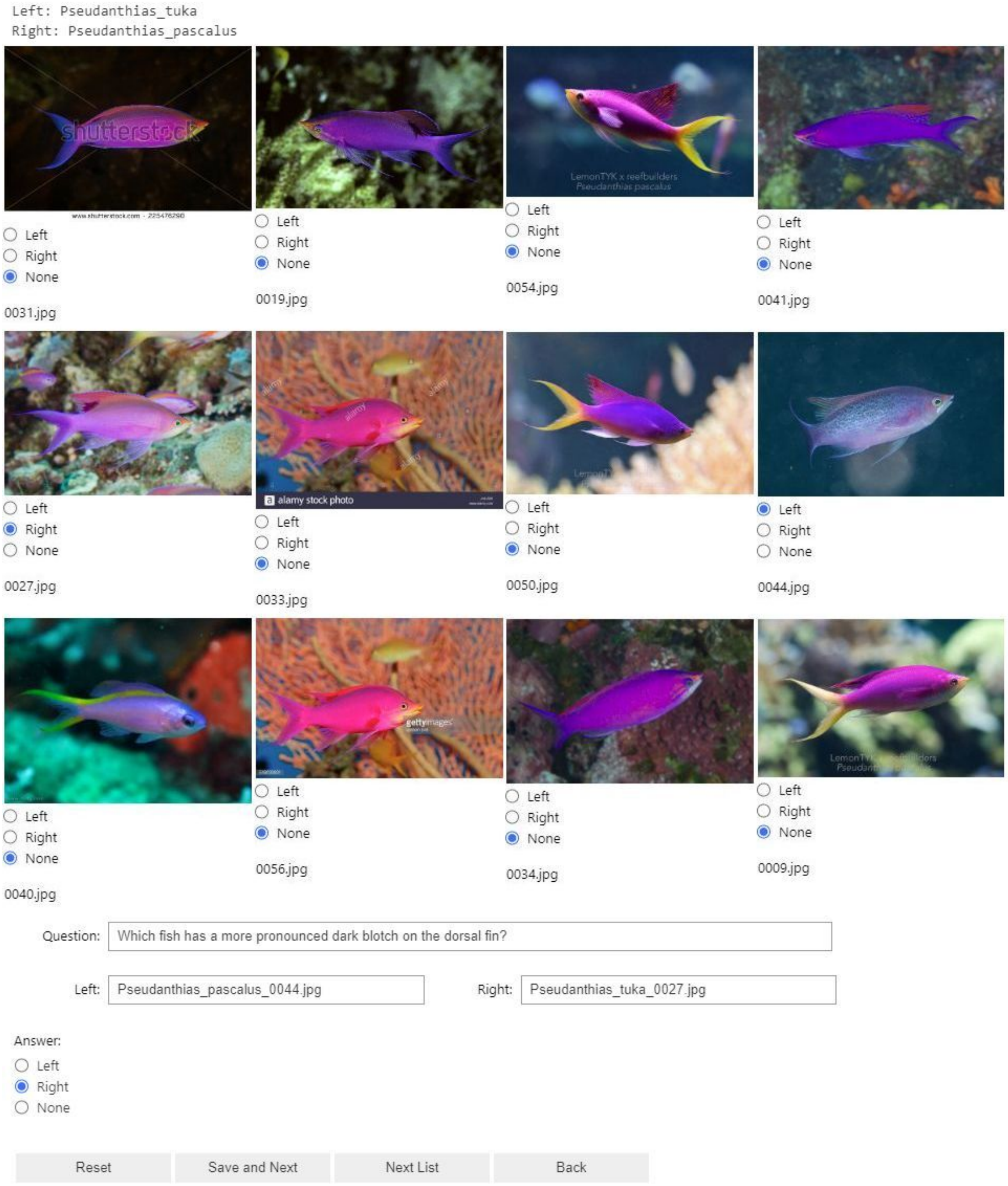
Question: Which bird has more blue on its wings?

Left: test\10_14\Indigo_Bunting_0084_11848.jpg

Right: test\10_14\Indigo_Bunting_0053_13391.jpg

Answer:
● Left
○ Right
○ None

Figure B.4: **Annotation Interface for CUB-200-2011.**



```
Left: Pseudanthias_tuka
Right: Pseudanthias_pascalus
```

○ Left
○ Right
● None

0031.jpg

○ Left
○ Right
● None

0019.jpg

○ Left
○ Right
● None

0054.jpg

○ Left
○ Right
● None

0041.jpg

○ Left
● Right
○ None

0027.jpg

○ Left
○ Right
● None

0033.jpg

○ Left
○ Right
● None

0050.jpg

● Left
○ Right
○ None

0044.jpg

○ Left
○ Right
● None

0040.jpg

○ Left
○ Right
● None

0056.jpg

○ Left
○ Right
● None

0034.jpg

○ Left
○ Right
● None

0009.jpg

Question: Which fish has a more pronounced dark blotch on the dorsal fin?

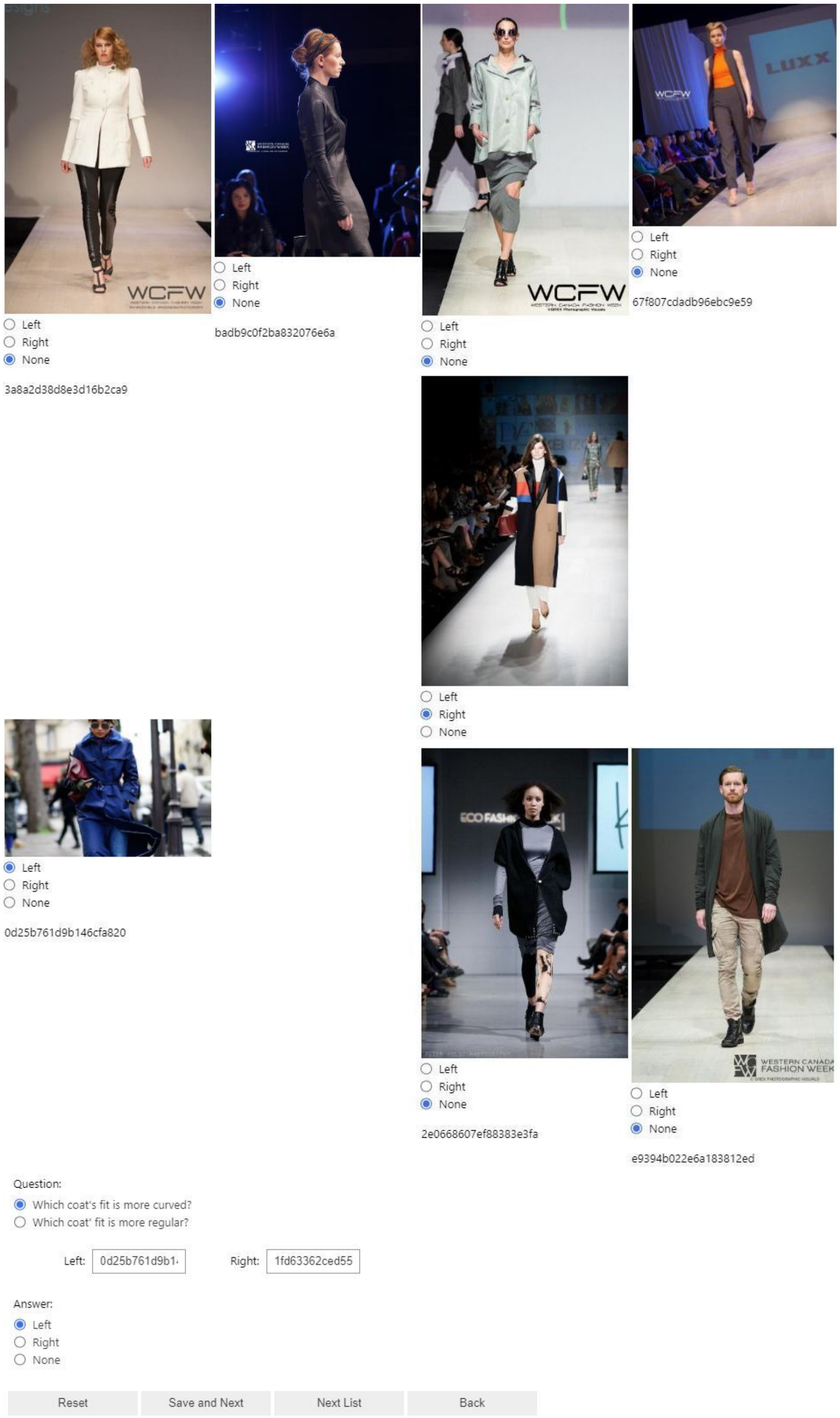
Left: Pseudanthias_pascalus_0044.jpg

Right: Pseudanthias_tuka_0027.jpg

Answer:

○ Left
● Right
○ None

Figure B.5: **Annotation Interface for Wildfish++.**



○ Left
○ Right
● None

3a8a2d38d8e3d16b2ca9

○ Left
○ Right
● None

badb9c0f2ba832076e6a

○ Left
○ Right
● None

○ Left
○ Right
● None

67f807cdadb96ebc9e59

○ Left
● Right
○ None

● Left
○ Right
○ None

0d25b761d9b146cfa820

○ Left
○ Right
● None

2e0668607ef88383e3fa

○ Left
○ Right
● None

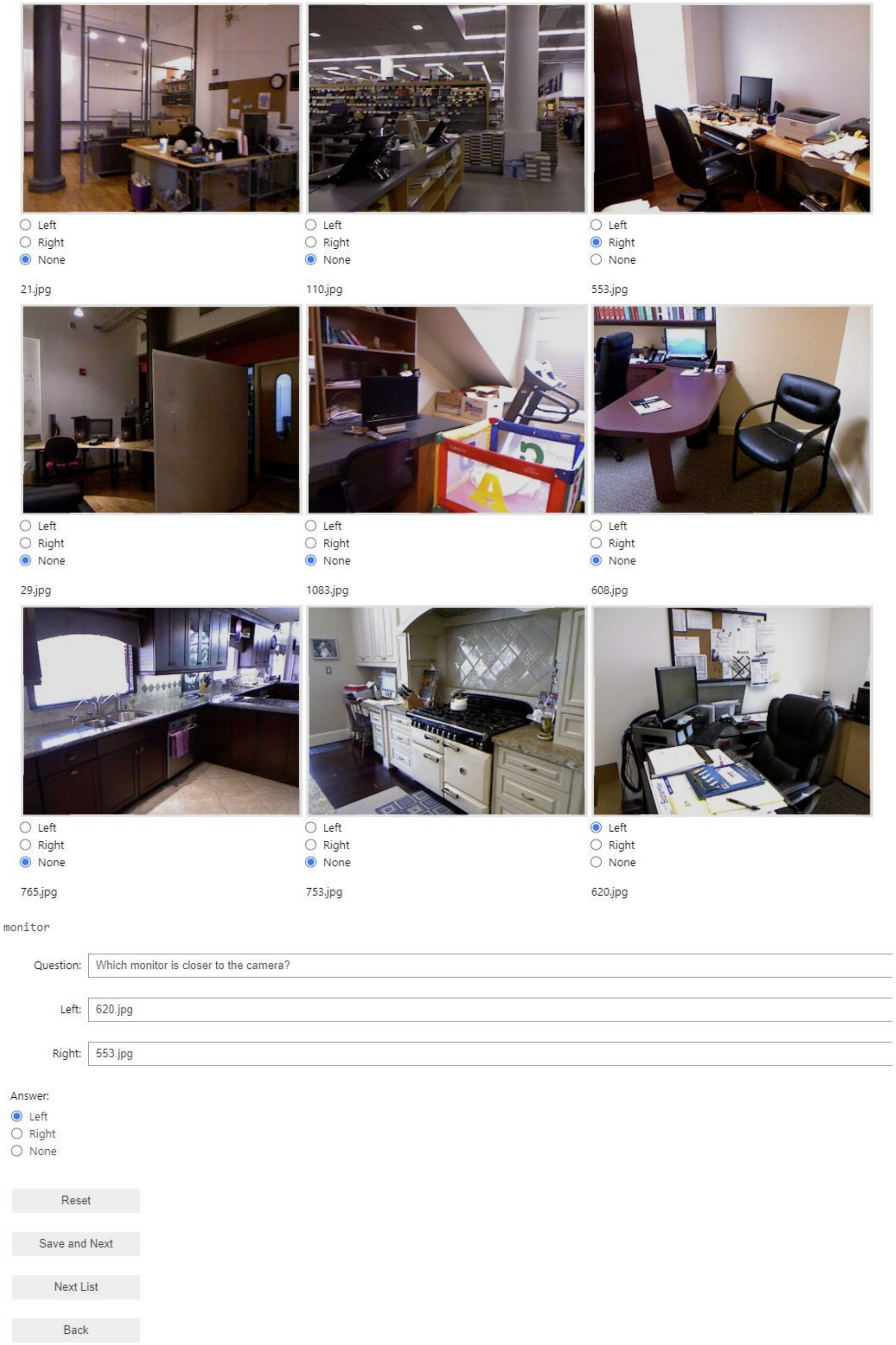
e9394b022e6a183812ed

Question:
● Which coat's fit is more curved?
○ Which coat' fit is more regular?

Left: 0d25b761d9b1 Right: 1fd63362ced55

Answer:
● Left
○ Right
○ None

Figure B.6: **Annotation Interface for Fashionpedia.**



○ Left
○ Right
● None

21.jpg

○ Left
○ Right
● None

110.jpg

○ Left
● Right
○ None

553.jpg

○ Left
○ Right
● None

29.jpg

○ Left
○ Right
● None

1083.jpg

○ Left
○ Right
● None

608.jpg

○ Left
○ Right
● None

765.jpg

○ Left
○ Right
● None

753.jpg

● Left
○ Right
○ None

620.jpg

monitor

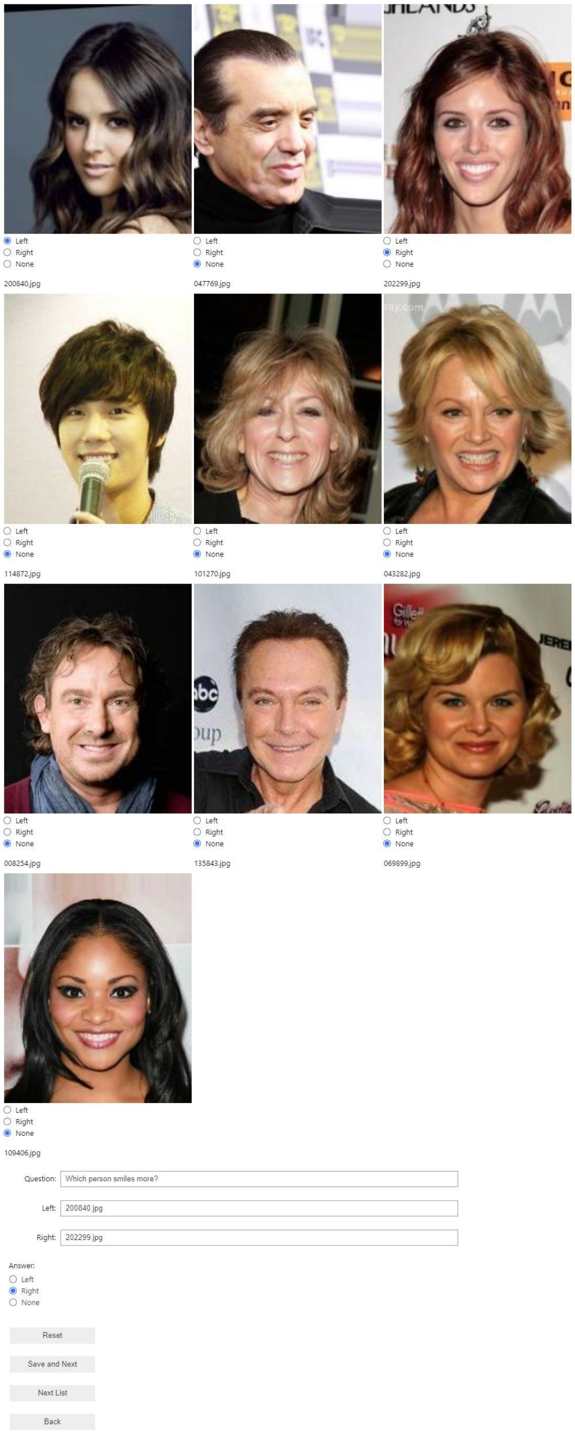
Question: Which monitor is closer to the camera?

Left: 620.jpg

Right: 553.jpg

Answer:
● Left
○ Right
○ None

Figure B.7: **Annotation Interface for NYU-Depth V2.**



● Left
○ Right
○ None

200840.jpg

○ Left
○ Right
● None

047769.jpg

○ Left
● Right
○ None

202299.jpg

○ Left
○ Right
● None

114872.jpg

○ Left
○ Right
● None

101270.jpg

○ Left
○ Right
● None

043282.jpg

○ Left
○ Right
● None

008254.jpg

○ Left
○ Right
● None

135843.jpg

○ Left
○ Right
● None

069899.jpg

○ Left
○ Right
● None

109406.jpg

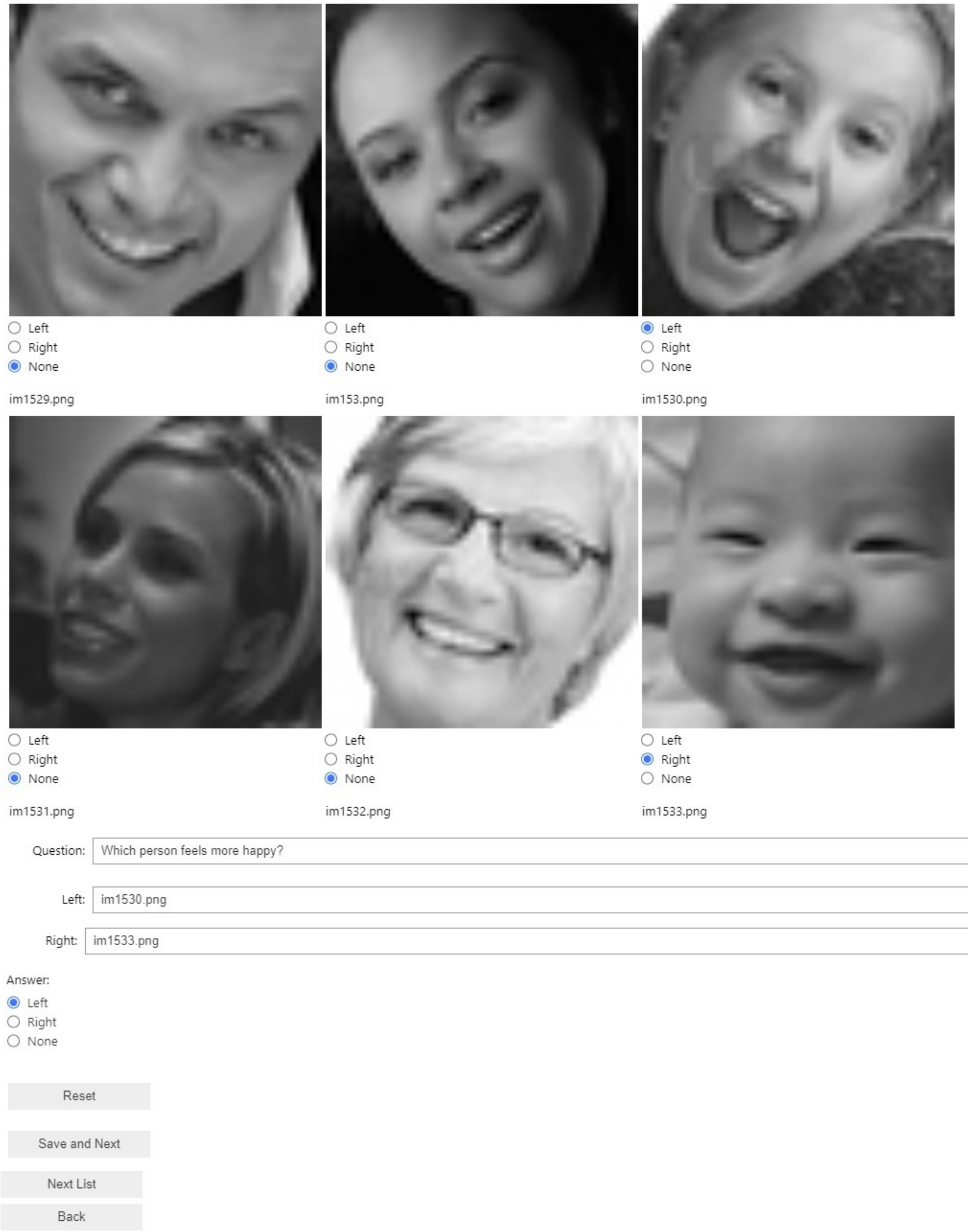
Question: Which person smiles more?

Left: 200840.jpg

Right: 202299.jpg

Answer:
○ Left
● Right
○ None

Figure B.8: **Annotation Interface for CelebA.**



○ Left
○ Right
● None

im1529.png

○ Left
○ Right
● None

im153.png

● Left
○ Right
○ None

im1530.png

○ Left
○ Right
● None

im1531.png

○ Left
○ Right
● None

im1532.png

○ Left
● Right
○ None

im1533.png

Question: Which person feels more happy?

Left: im1530.png

Right: im1533.png

Answer:
● Left
○ Right
○ None

Figure B.9: **Annotation Interface for FER-2013.**

Dataset	Model	Lagnauge Prompt
ST, FA, VA, CU, WF, CE, FE, ND	GPT-4V LLaVA-1.6 VILA-1.5	“[QUESTION] If you choose the first image, return Left, and if you choose the second image, return Right.”
	Gemini1.0-pro	“[QUESTION] If you choose the first image, return First, and if you choose the second image, return Second. Please only return either First or Second without any other words, spaces, or punctuation.”
MB, SD	GPT-4V LLaVA-1.6 VILA-1.5 Gemini1.0-pro	“What is the most obvious difference between the two images? Choose from the following options. If there is no obvious difference, choose None. Options: None, [OPTIONS].” ”
SN	GPT-4V LLaVA-1.6 VILA-1.5 Gemini1.0-pro	“These are two frames related to [SOCCER_ACTION] in a soccer match. Which frame happens first?”
CC	GPT-4V LLaVA-1.6 VILA-1.5	“Based on these images, which car is newer in terms of its model year or release year? Note that this question refers solely to the year each car was first introduced or manufactured, not its current condition or usage. If you choose the first image, return Left, and if you choose the second image, return Right. Please only return either Left or Right without any other words, spaces, or punctuation.”
	Gemini1.0-pro	Based on these images, which car is newer in terms of its model year or release year? Note that this question refers solely to the year each car was first introduced or manufactured, not its current condition or usage. If you choose the first image, return First, and if you choose the second image, return Second. Please only return either First or Second without any other words, spaces, or punctuation.”
VQ	GPT-4V LLaVA-1.6 VILA-1.5 Gemini1.0-pro	“[QUESTION] If the second image has more, return Right. If the first image has more, return Left. If both images have the same number, return Same.”
QB	GPT-4V LLaVA-1.6 VILA-1.5 Gemini1.0-pro	“[QUESTION] Options: [OPTIONS]”

Table B.2: **Language prompts for evaluating MLLMs**. ST: MIT-States [111], FA: Fashionpedia [115], VA: VAW [205], CU: CUB-200-2011 [261], WF: Wildfish++ [308], MB: MagicBrush [296], SD: Spot-the-diff [114], CE: CelebA [175], FE: FER-2013 [80], SN: SoccerNet [77], CC: CompCars [283], ND: NYU-Depth V2 [236], VQ: VQAv2 [81], QB: Q-Bench2 [297].

B.3 Training details on LLaVA-1.6

We conduct a study to evaluate whether fine-tuning enhances the comparative capabilities of MLLMs. Concretely, we focus on two relativities: Temporality and Quantity. For temporality, we construct a total of 20.6K training examples from SoccerNet [77], following the similar data collection and annotation protocol described in the main text. For quantity, we curate a total training set of 20.9K samples from VQAv2 [81], based on the similar data collection and annotation pipeline in the main text. We fine-tune LLaVA-1.6-34b [169] on each of these training datasets separately, using LoRA techniques. We follow similar hyperparameter settings as those provided in the official LLaVA source codes. For instance, batch size/the number of epochs/learning rate are 16/3/2e-5, respectively. See the training script in our GitHub repository for the complete configuration. All models are fine-tuned on four NVIDIA RTX 6000 Ada GPUs.

Appendix C: Aligning semantic representations with visual features

In this appendix, we provide details omitted in chapter 5.

C.1 Contribution

Our contribution is not merely in the method we developed, but also in the direction we explored. Most of the efforts in ZSL have focused on algorithm design to associate visual features and pre-defined semantic representations. Yet, it is also important to improve semantic representations. Indeed, one reason that ZSL performs poorly on large-scale datasets is the poor semantic representations [37]. We therefore chose to investigate this direction by revisiting document representations, with the goal to make our contributions widely applicable. To this end, we deliberately kept our method simple and intuitive, but also provided insights for future work to build upon. Our manual inspection identified important properties of visual sentences like the clustering structure, enabling us to efficiently extract them. We chose to not design new ZSL algorithms but make our semantic representations compatible with existing ones to clearly demonstrate the effectiveness of improving semantic representations.

C.2 More Related Work

Zero-shot learning (ZSL) algorithms construct visual classifiers based on semantic representations. Some recent work applies generative models to generate images or visual features of unseen classes [279, 278, 306], so that conventional supervised learning algorithms can be applied.

Knowledge bases usually contain triplets of entities and relationships. The entities are usually objects, locations, etc. For ZSL, we need entities to be fine-grained (e.g., "beaks") and capture more visual appearances. YAGO [241] and DBpedia [290] leverage Wikipedia infoboxes to construct triplets, which is elegant but not suitable for ZSL since Wikipedia infoboxes contain insufficient visual information. Thus, these datasets and construction methods may not be directly applicable to ZSL. Nevertheless, the underlying methodologies are inspiring and could serve as the basis for future work. The datasets also offer inter-class relationships that are complementary to visual descriptions, and may be useful to establish class relationships in ZSL algorithms like SynC [36].

C.3 Statistics of Wikipedia Pages

We use a Wikipedia API to extract pages from Wikipedia for ImageNet 21,842 classes. Among 21,842 classes, we find that some classes have multiple Wikipedia pages because of their ambiguous class names. For example, a class "*black widow*" in ImageNet refers to a spider with dark brown or a shiny black in colour, but it also refers to the name of a "*Marvel Comics*" character in Wikipedia. We therefore exclude such classes and also classes that do not have word vectors, resulting in 15,833 classes. The Wikipedia pages of the 15K classes contain 1,260,889 sentences where each class has 80 sentences on average. We also investigate the number of sentences by our filters (*i.e.* $\mathbf{Vis}_{\mathbf{sec}}$, $\mathbf{Vis}_{\mathbf{cls}}$, $\mathbf{Vis}_{\mathbf{sec\text{-}clu}}$). As a result,

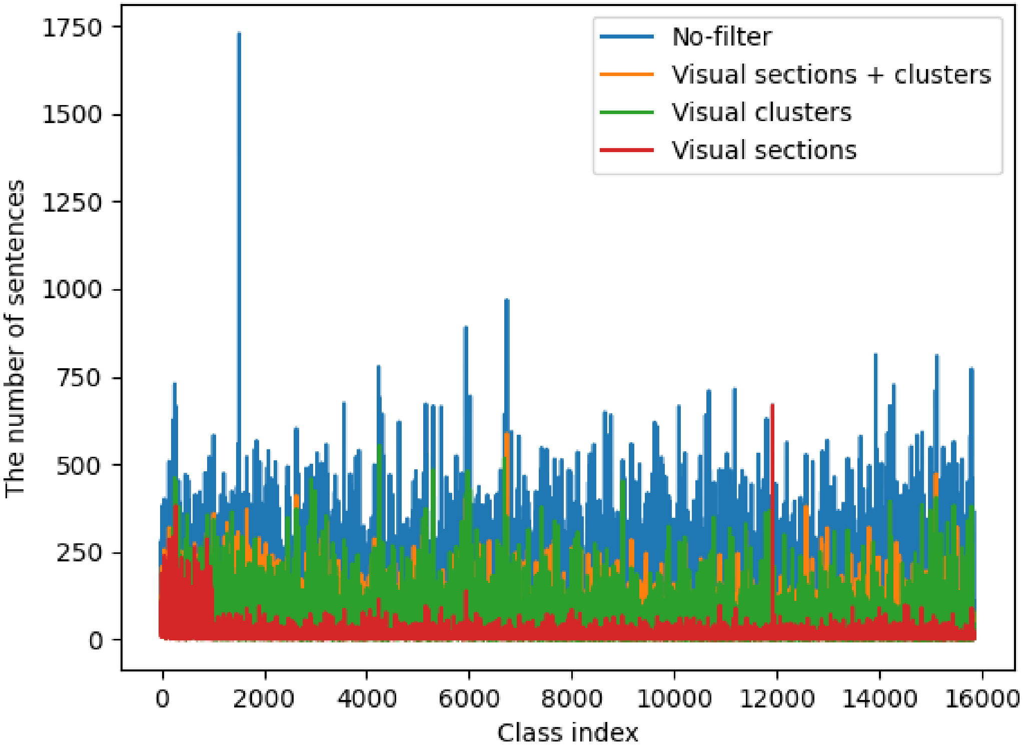


Figure C.1: Statistics of **Wikipedia** pages.

we correspondingly find 213,585, 534,852, 542,645 sentences, which are 16%, 42%, 43% of all sentences in 15K classes, respectively (See Figure C.1).

C.4 Weighted Average Representations

C.4.1 Observation

Two similar classes may have similar averaged visual sentence embeddings since they share many common descriptions. For example, Figure C.2 shows that the averaged embedding (*i.e.*, BERT_p and BERT_f) between “Kerry Blue Terrier” and “Soft-coated Terrier” are overly similar since they share a number of sentences containing the common dog features such as “a breed of dog” or “having a coat or a tail”. Thus, if we represent their semantic representations $\boldsymbol{a}_c$ as the averaged embeddings, ZSL models may not differentiate them.

C.4.2 Algorithm

In Section 5.3.4 of the main text, we introduce $\lambda(\cdot)$ to give each sentence $\boldsymbol{h}$ of a document a weight. We note that, while learning $\lambda(\cdot)$ can enlarge the distance of $\boldsymbol{a}_c$ between similar classes, we should not overly maximize the distance to prevent semantically similar classes (*e.g.*, different breed of dogs) end up being less similar than dissimilar classes (*e.g.*, dogs and cats). To this end, we introduce a margin loss with τ in Equation 5.5, which only penalize overly similar semantic representations.

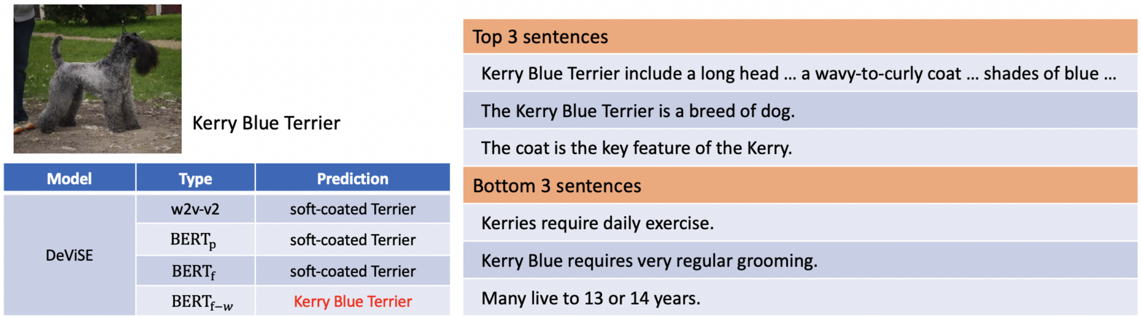
We also note that, the purpose of $\lambda(\cdot)$ is to improve $\boldsymbol{a}_c$ from the simple **average** embedding $\bar{\boldsymbol{a}}_c$. We therefore initialize $\lambda(\cdot)$ such that the initial $\boldsymbol{a}_c$ is similar to $\bar{\boldsymbol{a}}_c$. We do so by first learning b_ψ with the following objective:

$$\sum_{c \in S \cup U} \max\{0, \varepsilon - \cos(\boldsymbol{a}_c, \bar{\boldsymbol{a}}_c)\}. \tag{C.1}$$

We set $\varepsilon = 0.9$, forcing $\boldsymbol{a}_c$ and $\bar{\boldsymbol{a}}_c$ to have a similarity larger than 0.9.

C.4.3 Results

Figure C.2 demonstrates the effectiveness of the weighted average embedding $\text{BERT}_{\text{f-}\mathbf{w}}$. While other semantic representations predict “Kerry Blue Terrier” as other similar dog, “soft-coated Terrier”, $\text{BERT}_{\text{f-}\mathbf{w}}$ is able to classify the image correctly. In addition, based on the attention weights, we report the Top 3 sentences and the Bottom 3 sentences. The Top 1st sentence contains the inherent features for “Kerry Blue Terrier” such as *long head* or



Model	Type	Prediction
DeViSE	w2v-v2	soft-coated Terrier
	BERT_p	soft-coated Terrier
	BERT_f	soft-coated Terrier
	BERT_{f-w}	Kerry Blue Terrier

Top 3 sentences
Kerry Blue Terrier include a long head … a wavy-to-curly coat … shades of blue …
The Kerry Blue Terrier is a breed of dog.
The coat is the key feature of the Kerry.
Bottom 3 sentences
Kerries require daily exercise.
Kerry Blue requires very regular grooming.
Many live to 13 or 14 years.

Figure C.2: Qualitative analysis of a class *Kerry Blue Terrier*. w2v-v2, BERT_p, and BERT_f can not distinguish between *Kerry Blue Terrier* and *Soft-coated Terrier* since two classes share the common features of dogs such as “a breed of dog” or “having a coat or a tail”. On the other hand, our weighted average $\text{BERT}_\text{f-w}$ is able to differentiate them by weighting on the sentences. We report the Top 3 sentences and the Bottom 3 sentences based on the attention weights.

soft-to-curly coat while the Top 2nd and 3rd sentences describe general features of dogs. On the other hand, the Bottom 3 sentences do not have visual appearance of the object. This suggest that our weighted representation $\text{BERT}_\text{f-w}$ is more representative to “Kerry Blue Terrier” than other semantic representations.

C.5 Dataset, Features, Metrics, and ZSL Algorithm

For visual features, we use the $2,048$-dimensional ResNet visual features [94] provided by [277]. Word vectors can be found in [37]. Followed by [277], we use the average *per-class* Top-1 accuracy as our metric. Instead of simply averaging over all test images (*i.e.* the average *per-sample* Top-1 accuracy), this accuracy is obtained by first taking average over all images in each test class independently and then taking average over all test classes. Compared to the average *per-sample* accuracy, the *per-class* accuracy is a more suitable for ImageNet since the dataset is highly imbalanced [37]. The state-of-the-art algorithms in ZSL are EXEM and HVE proposed by [37] and [171], respectively. To make fair comparison with our models, we evaluate their algorithms on the same number of our test classes using their official codes.

C.5.1 ImageNet

We follow [277, 36] to consider three tasks, 2-Hop, 3-Hop, and ALL, corresponding to $1,509, 7,678$ and $20,345$ unseen classes that have word vectors and are within two, three, and arbitrary tree hop distances to the $1,000$ seen classes.

We search Wikipedia and successfully retrieve pages for **15,833** classes, of which **1,290**, **5,984**, and **14,840** are for 2-Hop, 3-Hop, and ALL.

C.5.2 AwA2

Animals with Attributes2 (AwA2) provides 37,322 images of 50 animal classes. On average, each class includes 746 images. It also provides 85 visual attributes that are manually annotated by humans. In AwA2, classes are split into 40 seen classes and 10 unseen classes. For GZSL, a total of 50 classes is used for testing.

C.5.3 aPY

Attribute Pascal and Yahoo (aPY) contains 15,339 images of 32 classes with 64 attributes. The classes are split into 20 seen classes and 12 unseen classes. A total of 32 classes is used for testing on GZSL.

C.5.4 DeViSE [69] vs. EXEM [37] vs. HVE [171]

All algorithms learn feature transformations to associate visual features $\boldsymbol{x}$ and semantic representations $\boldsymbol{a}_c$. The key differences are what and how to learn. DeViSE* learns two MLPs $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\phi}}$ to embed $\boldsymbol{x}$ and $\boldsymbol{a}_c$ into a common space, while HVE embeds them into a hyperbolic space. EXEM learns kernel regressors to embed $\boldsymbol{a}_c$ into the visual space. On how to learn, DeViSE* and HVE force each image $\boldsymbol{x}$ to be similar to the true class $\boldsymbol{a}_c$ by a margin loss and a ranking loss respectively, while EXEM learns to regress the averaged visual features of a class from $\boldsymbol{a}_c$.

C.6 Implementation Details

C.6.1 Sentence representations from BERT

Sentence representations can be defined in multiple ways such as a [CLS] token embedding or an average word embedding from different layers in BERT [215]. In our experiments, the average word embedding from the second last layer of BERT achieve the best results in all cases.

C.6.2 Hyperparameters

DeViSE [69] has a tunable margin $\Delta \geq 0$ (cf. Section 5.3.1 in the main text) which its default value is 0.1. We try multiple values 0.1, 0.2, 0.5, and 0.7 to find the best setting. DeViSE uses Adam optimizer which its learning rate is $1e^{-3}$ by default. We try different possible values, $1e^{-3}$, $5e^{-4}$, $2e^{-4}$, and $1e^{-4}$. Among all 16 possible combination of the margin and learning rate, we find that margin of $\mathbf{0.2}$ and learning rate of $\boldsymbol{2e^{-4}}$ achieve the best results on all our cases.

C.6.3 Fine-tuned models

For fine-tuning, DeViSE* is first attached to a BERT model. Then, we train the model with jointly fine-tuning BERT parameters based on the DeViSE* objective. Regards to BERT training, [101] demonstrates that fine-tuning only last few n layers (*e.g.* 2 or 4)

Model	Type	Filter	Threshold τ	2-Hop
DeViSE*	$\text{BERT}_{\text{p-w}}$	$\text{Vis}_{\text{sec-clu}}$	0.98	15.97
			0.97	16.09
			0.96	16.32
			0.95	16.13
	$\text{BERT}_{\text{f-w}}$	$\text{Vis}_{\text{sec-clu}}$	0.88	20.34
			0.86	20.44
			0.82	20.33
			0.80	20.47

Table C.1: Results of per-class Top-1 accuracy(%) on 2-Hop with different thresholds τ and semantic representation types. The best is in red and the second best in blue.

can outperform fine-tuning all layers in some NLP tasks. [138] also shows that the fine-tuning procedure is more effective to the last few layers than earlier layers. Considering the computational resources and time, we therefore set n equal to 2. After fine-tuning, we freeze BERT parameters and further train DeViSE*.

C.7 Ablation Study

Table C.1 shows the results on 2-Hop with different thresholds τ introduced in Equation 5.5. We obtain the weighted average $\text{BERT}_{\text{p-w}}$ by taking an input $\boldsymbol{h}$ from BERT_{p} and learning MLP b_{ψ} with different τ (similar for $\text{BERT}_{\text{f-w}}$). Then, we measure 2-Hop accuracy based on $\text{BERT}_{\text{p-w}}$ (or $\text{BERT}_{\text{f-w}}$). Note that BERT_{p} and BERT_{f} have different ranges of τ, since BERT_{f} already has lower similarity between classes. This is because BERT_{f} is trained with images (from seen classes) during fine-tuning, which makes BERT_{f} more aligned with visual features and thus is more representative. We choose τ based on the ImageNet validation set of the seen classes.

Table C.2 shows that the weighted average embedding $\text{BERT}_{\text{p-w}}$ makes similar classes less similar. Originally, a class “Sea boat” has overly similar semantic representations with

Class	Top3 Similar Classes	Similarity	
		BERT_p	$\text{BERT}_\text{p-w}$
Sea boat	Scow	0.94	0.91
	Row boat	0.93	0.91
	Canoe	0.93	0.91

Table C.2: Similarity of Top 3 similar classes with *Sea boat* drops after applying the weighting approach.

other type of boats (i.e. BERT_p). After applying our weighting approach, the classes become less similar (e.g. 0.94 to 0.91 between "Sea boat" and "Scow").

C.8 Qualitative Results

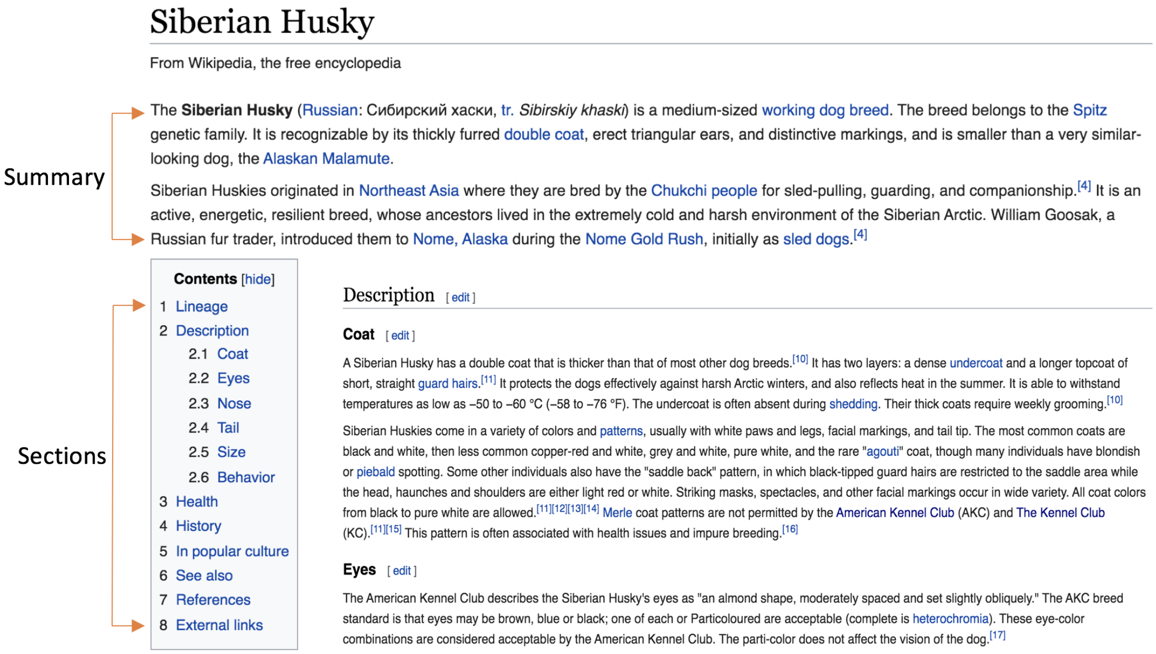
C.8.1 Visual sections and clusters

We provide additional illustrations of visual sections and clusters of Section 5.3 in the main text.

Figure C.3 shows visual and non-visual sections in a Wikipedia page **Siberian Husky**. We note that the summary paragraph and sections such as *Description* contain visual sentences while sections such as *Health* or *History* do not. Similarly, Table C.3 shows two clusters: the top cluster is visual, consisting of information about *hunting* and *preys* of animals while the bottom cluster includes *mythology* sentences not visually related.

Clusters
⋯hunt shortly after sunset, eating small animals ⋯
⋯ if food is scarce, it has been known to eat tomatoes ⋯
Tigers are capable of taking down larger prey like adult gaur ⋯
Tigers will also prey on such domestic livestock as cattle, horses, ⋯
Panda is a Roman goddess of peace and travellers ⋯
The Ibex is also a national emblem of the great ancient Axum empire.
In Aztec mythology, the jaguar was considered to be the totem animal of ⋯
It is the national animal of Guyana, and is featured in its coat of arms ⋯

Table C.3: K-means sentence clusters. The top cluster has *visual* information about *hunting* and *preys* while the bottom one contains *non-visual* description such as *mythology*.



Siberian Husky

From Wikipedia, the free encyclopedia

The **Siberian Husky** (Russian: Сибирский хаски, tr. *Sibirskiy khaski*) is a medium-sized working dog breed. The breed belongs to the Spitz genetic family. It is recognizable by its thickly furred double coat, erect triangular ears, and distinctive markings, and is smaller than a very similar-looking dog, the Alaskan Malamute.

Siberian Huskies originated in Northeast Asia where they are bred by the Chukchi people for sled-pulling, guarding, and companionship.[4] It is an active, energetic, resilient breed, whose ancestors lived in the extremely cold and harsh environment of the Siberian Arctic. William Goosak, a Russian fur trader, introduced them to Nome, Alaska during the Nome Gold Rush, initially as sled dogs.[4]

Contents [hide]

1 Lineage
2 Description
2.1 Coat
2.2 Eyes
2.3 Nose
2.4 Tail
2.5 Size
2.6 Behavior
3 Health
4 History
5 In popular culture
6 See also
7 References
8 External links

Description [edit]

Coat [edit]

A Siberian Husky has a double coat that is thicker than that of most other dog breeds.[10] It has two layers: a dense undercoat and a longer topcoat of short, straight guard hairs.[11] It protects the dogs effectively against harsh Arctic winters, and also reflects heat in the summer. It is able to withstand temperatures as low as –50 to –60 °C (–58 to –76 °F). The undercoat is often absent during shedding. Their thick coats require weekly grooming.[10]

Siberian Huskies come in a variety of colors and patterns, usually with white paws and legs, facial markings, and tail tip. The most common coats are black and white, then less common copper-red and white, grey and white, pure white, and the rare "agouti" coat, though many individuals have blondish or piebald spotting. Some other individuals also have the "saddle back" pattern, in which black-tipped guard hairs are restricted to the saddle area while the head, haunches and shoulders are either light red or white. Striking masks, spectacles, and other facial markings occur in wide variety. All coat colors from black to pure white are allowed.[11][12][13][14] Merle coat patterns are not permitted by the American Kennel Club (AKC) and The Kennel Club (KC).[11][15] This pattern is often associated with health issues and impure breeding.[16]

Eyes [edit]

The American Kennel Club describes the Siberian Husky's eyes as "an almond shape, moderately spaced and set slightly obliquely." The AKC breed standard is that eyes may be brown, blue or black; one of each or Particoloured are acceptable (complete is heterochromia). These eye-color combinations are considered acceptable by the American Kennel Club. The parti-color does not affect the vision of the dog.[17]

Figure C.3: Visual sections on *Siberian Husky*.

Appendix D: HTML representations with visual contextualization

In this appendix, we provide details omitted in chapter 6.

D.1 Model implementation & training details

As mentioned in the main text, we implement DUAL-VCR on top of MindAct algorithm [60]. We exactly follow its implementation[19] but provide the details for reference.

D.1.1 DUAL-VCR-enhanced element ranker

MindAct utilizes a small ranking LM to measure the importance of each element e_t for action prediction. Concretely, at each time step t, the ranking LM takes the element's HTML text tokens h_{e_t}, the task description q, and the previous actions $\{a_1, a_2, \cdots, a_{t-1}\}$ as input and outputs its importance,

$$s_{e_t} = f(q, h_{e_t}, \{a_1, a_2, \cdots, a_{t-1}\}) \tag{D.1}$$

DUAL-VCR aims to expand this ranking LM to integrate (i) each element's visual features and textual features and (ii) both the candidate element and its neighbor elements. (See Figure 6.4 of the main text for an illustration.)

[19] `https://github.com/OSU-NLP-Group/Mind2Web`

Integrating visual and textual features. We first extract each element's visual features from the Pix2Struct Vision Transformer (ViT) [145], pre-trained on webpage screenshots. Concretely, Pix2Struct learns rich representations of webpages by asking to predict an HTML-based parse from a masked screenshot. We input the whole screenshot I_t to Pix2Struct$_{\text{base}}$ and apply RoIAlign [93] on its output embeddings to obtain the element's visual features v_{e_t} based on its bounding box. On the HTML document side, we extract the element's HTML text h_{e_t}, using the triplet of its ID, HTML text, and bounding box provided in the HTML document.

Intergrating visual neighbor elements. Based on our key insight on webpages—web developers tend to arrange semantically relevant and task-related elements in proximity to each other on the screenshot to enhance user experiences—we contextualize each element e_t with its "visual" neighboring elements M_{e_t}. We measure the center points of all elements in the screenshot using their bounding boxes and calculate their pairwise Euclidean distances[20]. For each *candidate* element to be ranked by MindAct, we search for the closest M elements to form its context jointly.

Aligning visual and textual embedding spaces. After obtaining each element's visual features v_{e_t} and textual features h_{e_t}, we align them in the same embedding space. Following the recent practice of vision-and-language models (*e.g.*, BLIP-2 [150], LLaVA-1.5 [168]), we apply two linear projection layers W to map visual features into the textual embedding space. We then introduce a learnable positional embedding to (i) pair each projected visual feature u_{e_t} with its associated text tokens h_{e_t} and (ii) encode the relative distance between the candidate element e_t and its neighboring elements M_{e_t}. Concretely, we add the same positional embedding p_{e_t} to the candidate element's (projected) visual feature u_{e_t} and textual

[20] `https://scikit-learn.org`

Dataset	# Domains	# Websites	Website Type	# Tasks	Avg # Actions	Avg # HTML Elements	Avg # HTML Tokens
MiniWoB++ [108]	-	100	Simplified	100	3.6	28	500
Mind2Web [60]	31	137	Real-world	2,350	7.3	1,135	44,402

Table D.1: **Detailed Statistics of Mind2Web [60].** Min2Web is the first real-world web navigation benchmark, collecting over 100 real-world websites across various domains. Unlike previous benchmarks [108, 285], Mind2Web provides an extensive amount of real-world webpage content, including over 1K/44K HTML elements/tokens on average.

feature h_{e_t}. Besides, we sort the neighbors M_{e_t} based on their spatial distances from the candidate element e_t. We then encode the relative positional embedding $p_{m^k_{e_t}}$ (based on the spatial distance from the candidate) to each neighbor element's visual features $u_{m^k_{e_t}}$ and corresponding text tokens $h_{m^k_{e_t}}$. We denote the set of the neighbors' visual features by $U_{M_{e_t}}$. Similarly, $H_{M_{e_t}}$ and $P_{M_{e_t}}$ represent the set of their textual features and that of their positional embeddings, respectively. These positionally encoded visual and textual token embeddings (of the candidate and the neighbor elements) are passed into the ranking LM f; the visual features are prepended to the textual embeddings, serving as soft visual prompts,

$$\begin{aligned} s_{e_t} &= f(q, R_{e_t}, \{a_1, a_2, \cdots, a_{t-1}\}), \\ R_{e_t} &= [u_{e_t} + p_{e_t}; U_{M_{e_t}} + P_{M_{e_t}}; h_{e_t} + p_{e_t}; H_{M_{e_t}} + P_{M_{e_t}}] \end{aligned} \tag{D.2}$$

Training Details. In training, we only learn the projection layer W, the positional embeddings P, and the ranking LM f while keeping the ViT frozen. For the ranking LM, we use DeBERTa$_{\text{base}}$ [95], a small encoder-only LM. We exactly follow the configuration of MindAct. Specifically, we train the LM (together with a linear classifier) with a batch size of 32 and a learning rate of 3e-5 for 5 epochs. The LM outputs the element's importance score through a sigmoid activation function. The score is optimized with a binary cross-entropy loss, where the ground-truth element serves as a positive example, and elements randomly sampled from the webpage are considered negative examples. The LM is trained on a single Nvidia A6000 48GB GPU. During inference, we score all candidate elements in the webpage and select top-K elements for the action predictor.

D.1.2 DUAL-VCR-enhanced action predictor

Due to the high computational cost of directly passing an entire HTML document into LLMs, MindAct [60] restricts its input to only the top-K candidate elements selected from the ranking LM. Concretely, MindAct combines the selected elements into an HTML snippet H_t and feeds it into an LLM g, along with the task description q ("Find one-way flights from New York to Toronto.") and the previous actions $\{a_1, a_2, \cdots, a_{t-1}\}$ ("Type New York in the From box"). At each time step t, the objective is to predict an action a_t, composing of the target element e_t (*e.g.*, "[textbox] To") and its associated operation o_t (*e.g.*, "Type Toronto"),

$$\begin{aligned} & a_t = g(q, H_t, \{a_1, a_2, \cdots, a_{t-1}\}), \\ & a_t : \{e_t, o_t\} \end{aligned} \tag{D.3}$$

We note that MindAct converts the target element prediction problem into multiple-choice question-answering. Instead of directly generating the target element, they split top-K candidates into multiple clusters of five element options (including the "None" option) and ask the LLM to pick one element from each cluster. If more than one element is selected, they form a new group with the chosen ones and iterate this process until a single element is selected.

The action predictor of DUAL-VCR takes the same input as MindAct, except for appending each candidate element with its neighboring elements. We generate an HTML snippet S_t based on the top-K candidate elements and their adjacent elements, and input the snippet (with the task description and the previous actions) to the LLM g and predict the action a_t,

$$a_t = g(q, S_t, \{a_1, a_2, \cdots, a_{t-1}\}) \tag{D.4}$$

Training Details. We again adopt the configuration from MindAct. We train Flan-T5$_{\text{base}}$ [52], an instruction fine-tuned encoder-decoder LLM, with a batch size of 32 and a learning rate of 5e-5 for 5 epochs. We optimize its parameters with the language modeling loss on a single Nvidia A6000 48GB GPU.

D.2 Dataset Details

Mind2Web [60] recently proposed the first real-world web navigation benchmark, consisting of over 2,000 open-ended tasks from more than 100 real-world websites. They collect the websites across 31 diverse domains, including travel, shopping, entertainment, public service, etc. Unlike other existing benchmarks [108, 285] limited to simulated environments, Mind2Web instead focuses on real-world environments (Table D.1). For instance, Mind2Web provides real-world websites with rich content, including thousands of HTML elements, tens of thousands of HTML tokens, and 7.3 web-related actions per task on average.

Data Collection. Given a real-world website (*e.g.*, an airline website), Mind2Web first asks annotators to write open-ended realistic tasks (*e.g.*, “Find one-way flights from New York to Toronto.”) relevant to the website. The workers are then required to complete the defined task with a sequence of actions. Specifically, each action is composed of element selection and operation selection. The annotators should first find an element (*e.g.*, “[textbox] From”) relevant to the task on the webpage and perform an operation (*e.g.*, “Type New York”) on the element.

Dataset Split. The Mind2Web dataset provides a training split with 1,009 real-world tasks collected from 73 websites. Each task consists of a sequence of action samples. In total, there exist 7,775 samples in the training split. Mind2Web evaluates a web agent on three different test splits. **Test$_{\textbf{Cross-Domain}}$** measures the agent's generalizability to a new domain where it has not seen any websites or tasks associated with that domain during training. The split contains 912 tasks with 5,911 samples from 73 real-world websites. In **Test$_{\textbf{Cross-Website}}$**, while the agent is not exposed to test websites, it is trained on websites from the same domain and potentially with similar tasks. This configuration enables us to evaluate the agent's capacity to adapt to entirely new websites within familiar domains and tasks. This split consists of 177 tasks, along with 1,373 samples obtained from 10 websites. **Cross-Task** is a conventional test split, which is the random 20% of the dataset. The split has 252 tasks with 2,094 samples from 69 websites.

Task Details. The Mind2Web task consists of a sequence of actions, each comprising a pair of an actionable HTML element (*e.g.*, "[textbox] To") and an operation (*e.g.*, "Type Toronto"). Mind2Web provides three common operations: Click, Type, and Select. For Type and Select operations, an additional argument (*e.g.*, "Toronto") is required.

D.3 Additional Experiments

More powerful action predictor. We scale up the predictor from Flan-T5$_{\text{base}}$ to Flan-T5$_{\text{large}}$ to check whether our visual neighbors are still beneficial with the larger model. As shown in Table D.2, DUAL-VCR still achieves notable gains, suggesting the complementary capabilities of LLMs and our visual neighbors.

Neighbors from an HTML tree. An HTML document can be represented as a DOM tree, a hierarchical tree of HTML objects (*e.g.*, Element: <head>). Thus, we can also extract

Ranker	Action Predictor	Cross-Task		
		Ele. Acc	Op. F1	Step SR
MINDACT$_{\text{RANK}}$	MINDACT$_{\text{PRED-LARGE}}$	51.4	75.6	48.7
	DUAL-VCR$_{\text{PRED-LARGE}}$	54.2	79.5	50.9

Table D.2: **DUAL-VCR with a larger predictor.** We increase the size of the predictor from Flan-T5$_{\text{base}}$ to Flan-T5$_{\text{large}}$. Even with the larger predictor, DUAL-VCR notably outperforms the baseline, showing the complementarity of DUAL-VCR and LLMs.

each element's neighbors from the HTML tree. We compare the tree-based neighbors with our neighbors obtained from the screenshot (Table D.3). Our visual neighbors (DUAL-VCR$_{\text{PRED}}$) significantly outperform those defined by the HTML tree (HTMLTREENEI$_{\text{PRED}}$), suggesting that visual-spatial context is more beneficial.

Ranker with whole visual tokens. In the main text, we show that DUAL-VCR (*i.e.*, the use of visual neighbors) is more effective than the use of the entire image for web navigation (*e.g.*, DUAL-VCR$_{\text{PRED}}$ vs. WHOLEIMAGE$_{\text{PRED}}$, DUAL-VCR$_{\text{VNEI-TXT+VIS}}$ vs. WHOLEIMAGE$_{\text{RANK}}$). To further substantiate the efficacy of DUAL-VCR over using the whole image, we conduct additional experiments (Table D.3). Specifically, we train a ranker (WHOLEVISTOK$_{\text{RANK}}$) using *all visual tokens* extracted from the whole image based on the Pix2Struct ViT [145]. Like the previous results in the main text, WHOLEVISTOK$_{\text{RANK}}$ outperforms the baseline (*e.g.*, 44.1% vs. 42.0%), suggesting the benefit of utilizing the entire image. However, WHOLEVISTOK$_{\text{RANK}}$ falls short of DUAL-VCR$_{\text{VNEI-TXT+VIS}}$ (46.0%), which uses significantly fewer inputs (*i.e.*, only neighboring elements). This again supports the advantages of DUAL-VCR over the whole image regarding computational efficiency and performance.

Type of pre-trained visual features. Table D.4 summarizes the importance of the type of pre-trained visual features on web navigation. As discussed in the main text, to train

Ranker	Action Predictor	Cross-Task Ele. Acc
MindAct$_{\text{RANK}}$	MindAct$_{\text{PRED}}$	42.0
	WholeImage$_{\text{PRED}}$	43.6
	HTMLTreeNei$_{\text{PRED}}$	43.8
	Dual-VCR$_{\text{PRED}}$	44.4
WholeImage$_{\text{RANK}}$	MindAct$_{\text{PRED}}$	43.9
WholeVisTok$_{\text{RANK}}$		44.1
Dual-VCR$_{\text{VNEI-TXT}}$		44.6
Dual-VCR$_{\text{VNEI-TXT+VIS}}$		**46.0**
-	WholeHTML$_{\text{PRED}}$	38.6

Table D.3: **Additional results for Table 6.6 in the main text.** Our neighbors defined by a screenshot (Dual-VCR$_{\text{PRED}}$) notably outperform the neighbors defined by an HTML tree (HTMLTreeNei$_{\text{PRED}}$). Moreover, Dual-VCR$_{\text{VNEI-TXT+VIS}}$ is significantly better than WholeVisTok$_{\text{RANK}}$, which uses all visual tokens of the entire image. This again highlights the benefit of Dual-VCR in both computational efficiency and performance.

the ranker, we extract the element's visual features using Pix2Struct [145]'s VIT, pre-trained on webpage screenshots. We investigate if these pre-trained "screenshot" visual features (Dual-VCR$_{\text{VNEI-TXT+VIS-WEB}}$) indeed contain meaningful HTML context for downstream web navigation tasks. Concretely, we compare them with features extracted from ViT pre-trained on COCO [164], an object recognition benchmark containing common objects in "natural images". We denote a ranker using the COCO visual features by Dual-VCR$_{\text{VNEI-TXT+VIS-COCO}}$. We first observe that Dual-VCR$_{\text{VNEI-TXT+VIS-COCO}}$ outperforms Dual-VCR$_{\text{VNEI-TXT}}$ that only leverages elements' HTML text features to train the ranker (*e.g.*, 45.2% vs. 44.6% on Ele. Acc). This implies that even if visual features are from a different domain (*i.e.*, natural images), incorporating them is still helpful in web navigation tasks. However, compared to Dual-VCR$_{\text{VNEI-TXT+VIS-WEB}}$, which uses both HTML visual and textual features, Dual-VCR$_{\text{VNEI-TXT+VIS-COCO}}$ performs less (*e.g.*, 46.0% vs. 45.2% on Ele. Acc). This highlights that the pre-trained "screenshot" visual features indeed contain

Ranker	Cross-Task		
	Ele. Acc	Op. F1	Step SR
DUAL-VCR$_{\text{VNEI-TXT}}$	44.6	75.7	43.2
DUAL-VCR$_{\text{VNEI-TXT+VIS-COCO}}$	45.2	76.3	43.4
DUAL-VCR$_{\text{VNEI-TXT+VIS-WEB}}$	**46.0**	**78.6**	**44.8**

Table D.4: **Effects of different types of pre-trained visual features.** The pre-trained screenshot visual features [145] are more beneficial on the downstream web navigation than those extracted from ViT pre-trained on natural images of COCO [164].

HTML-related context, which benefits more in completing the downstream web navigation tasks.

Existing/Concurrent Works. A number of previous studies [108, 285, 157, 244, 31, 166, 116, 240, 230] have explored web navigation but mainly worked on *simplified* websites [108, 285], which deviate from the focus of our study. Our attention is instead directed towards *real-world* scenarios involving various real-world websites with extensive raw HTML documents (*e.g.*, Mind2Web). We have identified a few *concurrent* works [71, 86, 299, 91, 99, 48] exploring Mind2Web, but they mostly focus on (i) large-scale pre-training, requiring substantial amounts of pre-training HTML data, or (ii) evaluating the potential of recent vision-and-language models (*e.g.*, GPT4-V [195]) as a web agent. As their codes or pre-training datasets have not been released yet, replicating their work would be prohibitively costly. We thus do not consider them in our studies.

Appendix E: Learning from data with appropriate learning objectives

In this appendix, we provide details omitted in chapter 7.

E.1 V&L model implementation details

Our model is an encoder-decoder V&L architecture consisting of ViT-B/16 [64] as a visual module and mT5-Base [282] as a language module. For the vision module, we adopt a transformer-based vision model ViT [64] pre-trained on JFT-3B dataset [293], the extension of JFT-300M [243], with 3 billion images collected from the web. Our language module is initialized from mT5-Base [282], a multilingual variant of T5 [209], pre-trained on a new Common Crawl-based dataset with 101 different languages.

During training, all parameters in vision and language blocks are updated simultaneously. We choose Adafactor [231] as an optimizer with $\beta_1 = 0$ and second-moment exponential decay = 0.8. For a learning rate, we schedule a linear warmup for 1K steps with inverse square-root decay. Our V&L architecture is implemented in Jax/Flax [27] based on the open-source T5X [220] framework.

We have done extensive hyperparameter tuning for our experiments. For instance, we find that the best hyper-parameter configuration for SPLITOCR pre-training is — initial (peak) learning rate: 1e-3, batch size: 256, image resolution: 640x640, the length of input/target text tokens: 40/26, and dropout: 0.1. For TextVQA, we achieve the best result with initial

Hyper-parameter	Pre-training	Downstream					
	SPLITOCR	ST-VQA	TextVQA	VW-VQA	VQAv2	TextCaps	VW-Cap
Initial (peak) learning rate	1e-3	9e-4	2e-4	9e-4	1e-3	2e-4	2e-4
Batch size	256	256	256	256	512	256	256
Image resolution	640x640	640x640	640x640	640x640	640x640	640x640	640x640
Length of input text tokens	40	72	72	72	72	56	56
Length of target text tokens	26	8	8	8	8	64	64
Dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Table E.1: **Best hyper-parameters for our experiments.** Among hyper-parameters of our V&L model, we find that initial (peak) learning rate, batch size, image resolution, length of input/target text tokens, and dropout are major components affecting the performance of our tasks. VW-VQA: VizWiz-VQA, VW-Cap: VizWiz-Captions

learning rate: 2e-4 and the length of input/target text tokens: 72/8 (See Table E.1 for more details).

E.2 Pre-training & Scene-text V&L datasets

We provide more details about pre-training and scene-text V&L datasets used in our experiments.

Scene-Text on CC15M. We estimate the portion of scene text on CC15M with a study on 300 randomly sampled images. We manually check each image and found: 59% (177/300) have scene text; only 13% (38/300) are watermark-only images. This aligns with TAP's report [284] on CC3M (scene-text: 42%, watermark-only: 5%). Note that TAP mentioned *"only the CC dataset contains a reasonable portion of images with meaningful scene text regions"*, suggesting CC15M is suitable for STU pre-training.

ST-VQA [23] is for scene-text VQA dataset. Its images are collected from various resources: COCO-Text [260], Visual Genome [141], VizWiz [88], ICDAR [129, 128], ImageNet [59], and IIIT-STR [190]. Since there is no official validation set, we follow the split provided by M4C [104], resulting in 23K/26K training/validation VQA examples.

TextVQA [237] for scene-text VQA. It is a subset of Open Images [139] with scene-text related QA pairs from human annotators with ten ground-truth answers. It has 34K/5K training/validation VQA examples from 21K/3K images.

VizWiz-VQA [88]. The dataset contains 20K/3K training/validation VQA examples collected from blind users. Due to the nature of the questions asked by blind people, we identify this benchmark as a candidate to benefit from scene-text understanding, even though it was not directly designed for scene-text VQA.

VQAv2 [81]. We further evaluate PRESTU on standard VQA benchmark to check if the scene-text recognition can also help on general VQA tasks. Following [118], we use the VQAv2 train/dev splits of *train2014/minival2014, which are 592K/65K VQA examples in total.

TextCaps [235] for scene-text image captioning task. It uses the same subset of OpenImages images with TextVQA. Each image has five ground-truth captions, totaling 100K/15K training/validation captions.

VizWiz-Captions [89]. Like Vizwiz-VQA, this benchmark was generated by blind users to solve their daily visual challenges. It contains 23.4K/7.7K training/validation images, where each image is paired with five captions. In total, there are 117K/38K training/validation image captions.

OCR-VQA [191] is an OCR-based VQA dataset about images of book covers. Concretely, it requires models to answer visual questions by reading/interpreting the text on the book covers (*e.g.*, author, title). In summary, OCR-VQA provides 207K images of book covers and more than 1 million VQA examples.

DocVQA [186] asks for the textual (handwritten, typewritten, printed) content on the document images. In contrast with general VQA [81], models should understand additional visual cues, including layout (*e.g.*, tables), style (*e.g.*, font, color), and non-textual elements (*e.g.*, tick boxes). In total, DocVQA contains 50K VQA examples with more than 12K document images.

ChartQA [184] is a VQA benchmark based on charts. Specifically, it covers more than 23K VQA examples from 17K charts. In ChartQA, models are required to perform complex reasoning (*e.g.*, logical and arithmetic operations) to understand charts and the corresponding questions.

AI2D [131] is a VQA dataset of illustrative diagrams. The task of AI2D is to answer diagram-related questions by analyzing the diagram structure and identifying its visual entities and their semantic relationships. AI2D provides 5K diagrams with 15K VQA examples in total.

WidgetCap [158] aims to generate language descriptions for UI elements (widgets) in the mobile interface. Mobile apps often lack widget captions in their interfaces, which recently becomes a primary issue for mobile accessibility. WidgetCap attempts to solve this challenge by providing an evaluation benchmark containing more than 162K language phrases (*i.e.*, captions) with 61K UI elements.

Screen2Words [263] is an image captioning task to generate a short summary of the mobile screen. To complete the task, models should have the capability of understanding the screen and conveying its content and functionalities in a concise language phrase. Screen2Words consists of 112K captions for 22K mobile screens in total.

E.3 More comparisons to prior works

Comparison to TAP. While PRESTU adopts a *general* pre-training dataset (*i.e.*, CC15M), TAP's pre-training data aggregates scene-text *dedicated* downstream data, including ST-VQA, TextVQA, TextCaps, and OCR-CC. Thus, even if the size of TAP's pre-training data (1.5M) is smaller, it may align better with the downstream tasks. However, since TAP's approach focuses on the specific downstream tasks, it is less applicable to other V&L tasks, whereas PRESTU provides a more flexible interface.

Moreover, TAP adopts closed-set prediction by training an answer classifier based on the dataset-specific vocabulary. This may benefit the accuracy of the corresponding downstream task. In contrast, PRESTU chooses open-ended prediction as it is more generalizable in practice and is adopted by many recent works (*e.g.*, PaLI, GIT).

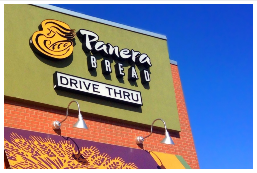
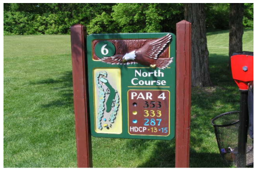
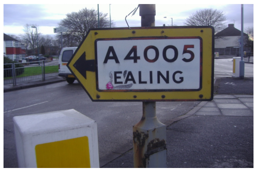
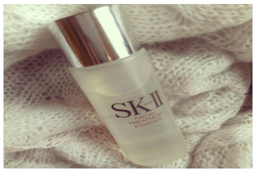
Image	gOCR token	PreSTU OCR token prediction
	panera bread drive thru	panera bread drive thru
 6 North Course PAR 4 353 333 287 HDCP - 13-15	north course par 4 353 333 287 hdcp - 13-15	north course par 4 333 333 287
	a 4005 ealing	a 4005 ealing
	sk - ii facial treatment essence	sk - ii facial treatment essence

Figure E.1: **PreSTU’s OCR token prediction.** The quality of OCR tokens generated by SplitOCR is comparable to that of gOCR system. This shows the possibility of leveraging SplitOCR as an alternative OCR system when other systems are not available.

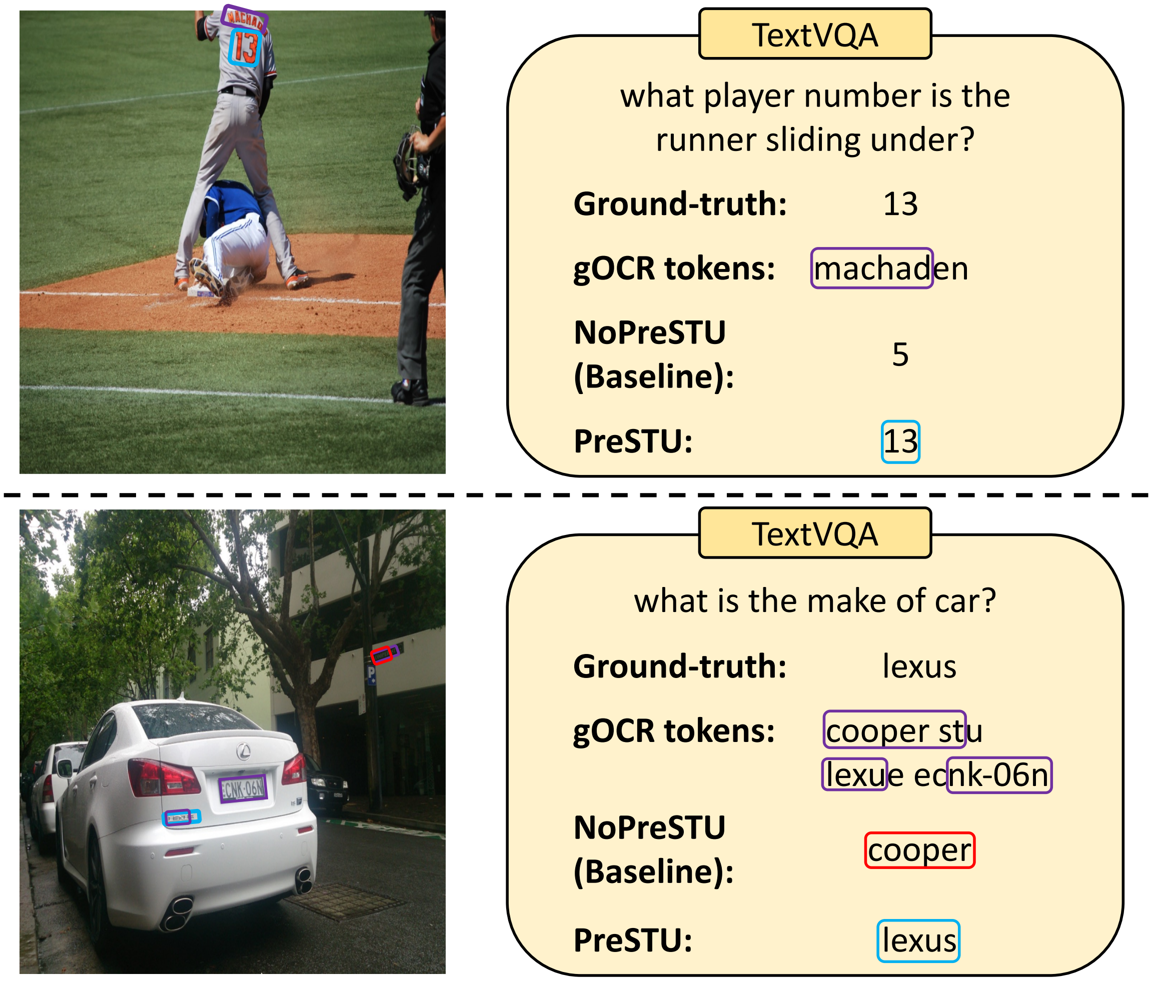


Figure E.2: **gOCR tokens vs. PreSTU prediction on TextVQA.** gOCR system does not detect some OCR tokens in the image (*e.g.*, "13") or detects them incorrectly (*e.g.*, "lexue"). This leads NoPreSTU to predict wrong answers (*e.g.*, "5" or "cooper"). On the other hand, SPLITOCR with gOCR tokens as input predicts the answers correctly with correct OCR tokens (*e.g.*, "13" or "lexus").

E.4 More ablation studies

SPLITOCR vs. CAP. Table 7.1 of the main text shows the effectiveness of SPLITOCR against VQA on VQA tasks. We further check its benefit over CAP on VQA tasks. As shown in Table E.2, SPLITOCR consistently improves over CAP (*e.g.*, 53.2% vs. 49.3%) on TextVQA, further supporting that SPLITOCR is important for higher accuracy.

We also investigate the effect of the order of pre-training stages. Concretely, we switch the order between SPLITOCR and CAP and demonstrate that applying SPLITOCR first (*i.e.*, default setting) is better (Table E.3).

Order of OCR. PRESTU uses the fixed OCR order to standardize the target output sequence during pre-training. Compared to the random order, we see its advantage with consistent improvements (*e.g.*, 132.4 vs. 134.6 on TextCaps CIDEr / 55.3% vs. 55.6% on TextVQA).

OCR System. We note that different prior works often use different *commercial* OCR engines to obtain their best results. Thus, it is hard to perform a fair comparison without extra costs. That said, we did evaluate PRESTU with different OCR engines (including Rosetta-en) at the downstream stage (Table 7.10 of the main text). A similar setup is used in LaTr [22]: Rosetta-en/Amazon-OCR for downstream TextVQA/pre-training, respectively. In this setup, PRESTU outperforms LaTr on TextVQA Val (50.7% vs. 48.4%).

Model	Pre-training Objective	TextVQA Val Acc
PRESTU	CAP	49.3
	SPLITOCR→CAP	53.2
	CAP→VQA	50.0
	SPLITOCR→CAP→VQA	55.0

Table E.2: **SPLITOCR vs. CAP on VQA tasks.** SPLITOCR is crucial for higher accuracy.

Model	Pre-training Objective	TextCaps Val CIDEr
PRESTU	SPLITOCR→CAP	141.7
	CAP→SPLITOCR	135.4

Table E.3: **Effect of switching pre-training stages.** Applying SPLITOCR first (*i.e.*, default setting) is more effective.

E.5 Qualitative results

Figure E.1 shows some examples of OCR tokens generated by SPLITOCR. Our SPLITOCR detects all (or almost all) OCR tokens in the images correctly, competitive to the gOCR system.

In §7.4.2 of the main text, we demonstrate that having two sources of OCR signals is beneficial (OCR signals by pre-trained ViT with SPLITOCR and OCR signals by gOCR system). Figure E.2 further supports this finding qualitatively. For instance, gOCR alone does not detect some OCR tokens in the image (*e.g.*, “13”) or detects them incorrectly (*e.g.*, “lexue”). This leads NOPRESTU to predict wrong answers (*e.g.*, “5” or “cooper”). On the other hand, SPLITOCR with gOCR tokens as input predicts the answers correctly with

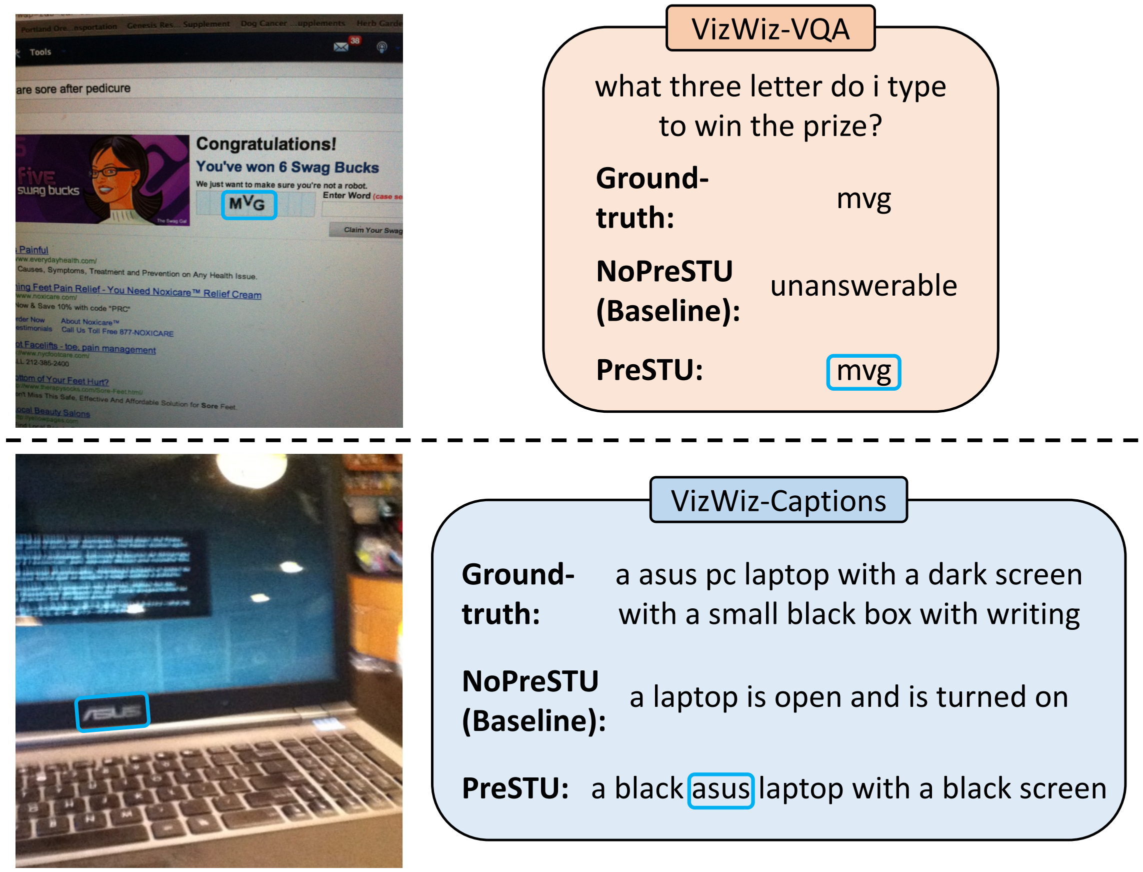

VizWiz-VQA
what three letter do i type to win the prize?
Ground-truth: mvg
NoPreSTU (Baseline): unanswerable
PreSTU: mvg
Congratulations!
You've won 6 Swag Bucks
MVG
VizWiz-Captions
Ground-truth: a asus pc laptop with a dark screen with a small black box with writing
NoPreSTU (Baseline): a laptop is open and is turned on
PreSTU: a black asus laptop with a black screen
ASUS

Figure E.3: **Qualitative results on VizWiz-VQA [88] and VizWiz-Captions [89].**

correct OCR tokens (*e.g.*, "13" or "lexus"), demonstrating that two sources of OCR signals (*i.e.*, ViT and gOCR) are complementary.

Figure E.3 provides qualitative results for VizWiz-VQA and VizWiz-Captions, demonstrating the applicability of PreSTU to different VQA and image captioning tasks.

E.6 Contributions

While our SPLITOCR is inspired by SimVLM [271], the motivation is fundamentally different and it is not trivial to apply the prefix idea in the first place for OCR-aware pre-training. Concretely, SimVLM aims to serve downstream tasks that generate text like captions or answers (with optional text input). Thus, it is understandable why SimVLM could help. In contrast, for downstream STU tasks, *OCR strings often serve only as the text input.* Therefore, while it makes sense to apply our second stage pre-training (CAP & VQA) with OCR strings as the input, it is not intuitive to develop a separate OCR-only pre-training stage (SPLITOCR) that leverages the idea of SimVLM. We came up with SPLITOCR purely from the two essential STU capabilities: (i) recognizing text in an image, (ii) connecting the text to its visual context. Our contribution thus lies in how to fulfill the two requirements via a unified manner, which turns out to be a SimVLM-like objective.

Besides SPLITOCR, another key contribution of our work is the comprehensive investigation of pre-training STU capabilities using a combination of easily reproducible objectives and a standard network architecture, on domains much more diverse than in previous works. Thus, we believe that our extensive analysis is valuable to the community.

Finally, we demonstrate the effectiveness of our OCR-aware method in large-scale settings. We choose CC15M as the pre-training dataset, which is often considered large-scale, and PaLI [45], an extremely large-scale model (with 10B data), utilizes our objective to achieve SOTA results on nearly all STU tasks (cf. §7.4.1.4 of the main text). This shows the utility of our pre-training objectives even in SOTA large-scale models.

Bibliography

[1] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *Proceedings of CVPR*, 2020.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[4] Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *Proceedings of CVPR*, 2020.

[5] Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *Proceedings of EMNLP*, 2016.

[6] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of CVPR*, 2018.

[7] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *ICCV*, 2019.

[8] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *TPAMI*, 38(7):1425–1438, 2015.

[9] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015.

[10] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, 2021.

[11] Ziad Al-Halah and Rainer Stiefelhagen. Automatic discovery, association estimation and learning of semantic attributes for a thousand categories. In *CVPR*, 2017.

[12] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[13] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[14] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Word spotting and recognition with embedded attributes. *In TPAMI*, 2014.

[15] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016.

[16] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of CVPR*, 2018.

[17] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[18] Anthropic. Model card and evaluations for claude models. 2023.

[19] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of ICCV*, 2015.

[20] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *In ICLR*, 2024.

[21] Pratyay Banerjee, Tejas Gokhale, Yezhou Yang, and Chitta Baral. Self-supervised vqa: Answering visual questions using images and captions. *arXiv preprint arXiv:2012.02356*, 2020.

[22] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *CVPR*, 2022.

[23] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, C.V. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, 2019.

[24] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *In TACL*, 2017.

[25] Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, et al. An introduction to vision-language modeling. *arXiv preprint arXiv:2405.17247*, 2024.

[26] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018.

[27] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. Jax: composable transformations of python+ numpy programs. *Version 0.2*, 2018.

[28] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[29] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[30] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *In NeurIPS*, 2020.

[31] Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A Plummer. A dataset for interactive vision-language navigation with unknown command feasibility. In *ECCV*, 2022.

[32] Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. RUBi: Reducing unimodal biases in visual question answering. In *Proceedings of NeurIPS*, 2019.

[33] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, 2007.

[34] Kai-Po Chang, Chi-Pin Huang, Wei-Yuan Cheng, Fu-En Yang, Chien-Yi Wang, Yung-Hsuan Lai, and Yu-Chiang Frank Wang. Rapper: Reinforced rationale-prompted paradigm for natural language explanation in visual question answering. In *The Twelfth International Conference on Learning Representations*, 2024.

[35] Kuang-Yu Chang, Chu-Song Chen, and Yi-Ping Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In *CVPR*, 2011.

[36] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016.

[37] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Classifier and exemplar synthesis for zero-shot learning. *IJCV*, 128(1):166–201, 2020.

[38] Soravit Changpinyo, Wei-Lun Chao, and Fei Sha. Predicting visual exemplars of unseen classes for zero-shot learning. In *ICCV*, 2017.

[39] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

[40] Wei-Lun Chao, Hexiang Hu, and Fei Sha. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *Proceedings of NAACL*, 2018.

[41] Wei-Lun Chao, Hexiang Hu, and Fei Sha. Cross-dataset adaptation for visual question answering. In *Proceedings of CVPR*, 2018.

[42] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of CVPR*, 2020.

[43] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[44] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

[45] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Alexander Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari,

Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *ICLR*, 2023.

[46] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[47] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[48] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*, 2024.

[49] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. `https://lmsys.org/blog/2023-03-30-vicuna/`, 2023.

[50] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[51] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *In NeurIPS*, 2017.

[52] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[53] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of EMNLP*, 2019.

[54] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[55] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al.

Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[56] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[57] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *In NeurIPS*, 2024.

[58] Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163:90–100, 2017.

[59] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. Ieee, 2009.

[60] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *In NeurIPS*, 2024.

[61] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, 2014.

[62] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[63] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[64] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[65] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[66] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *ICCV*, 2013.

[67] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009.

[68] Maxwell Forbes, Christine Kaeser-Chen, Piyush Sharma, and Serge Belongie. Neural naturalist: Generating fine-grained image comparisons. *In EMNLP*, 2019.

[69] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[70] Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In *ECML*, 2003.

[71] Hiroki Furuta, Ofir Nachum, Kuang-Huei Lee, Yutaka Matsuo, Shixiang Shane Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models. *In ICLR*, 2024.

[72] Chuang Gan, Yandong Li, Haoxiang Li, Chen Sun, and Boqing Gong. VQS: Linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation. In *Proceedings of ICCV*, 2017.

[73] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb: The paraphrase database. In *Proceedings of NAACL*, 2013.

[74] Roy Ganz, Oren Nuriel, Aviad Aberdam, Yair Kittenplon, Shai Mazor, and Ron Litman. Towards models that can see and read. *arXiv preprint arXiv:2301.07389*, 2023.

[75] Yuan Gao, Kunyu Shi, Pengkai Zhu, Edouard Belval, Oren Nuriel, Srikar Appalaraju, Shabnam Ghadar, Zhuowen Tu, Vijay Mahadevan, and Stefano Soatto. Enhancing vision-language pre-training with rich supervisions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13480–13491, 2024.

[76] Itai Gat, Idan Schwartz, Alexander Schwing, and Tamir Hazan. Removing bias in multi-modal classifiers: Regularization by maximizing functional entropies. In *Proceedings of NeurIPS*, 2020.

[77] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *CVPR Workshops*, 2018.

[78] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. MUTANT: A training paradigm for out-of-distribution generalization in visual question answering. In *Proceedings of EMNLP*, 2020.

[79] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. Vqa-lol: Visual question answering under the lens of logic. In *Proceedings of ECCV*, 2020.

[80] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing: 20th International Conference, ICONIP*, 2013.

[81] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

[82] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of CVPR*, 2017.

[83] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

[84] Gabriel Grand and Yonatan Belinkov. Adversarial regularization for visual question answering: Strengths, shortcomings, and side effects. In *Proceedings of ACL Workshop*, 2019.

[85] Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*, 2023.

[86] Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. *In ICLR*, 2024.

[87] Danna Gurari, Qing Li, Chi Lin, Yinan Zhao, Anhong Guo, Abigale Stangl, and Jeffrey P Bigham. Vizwiz-priv: A dataset for recognizing the presence and purpose of private visual information in images taken by blind people. In *Proceedings of CVPR*, 2019.

[88] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz Grand Challenge: Answering visual questions from blind people. In *CVPR*, 2018.

[89] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *ECCV*, 2020.

[90] Wei Han, Hantao Huang, and Tao Han. Finding the evidence: Localization-aware answer prediction for text visual question answering. *arXiv preprint arXiv:2010.02582*, 2020.

[91] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.

[92] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[93] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[94] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[95] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *In ICLR*, 2021.

[96] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *CVPR*, 2018.

[97] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *ICLR*, 2021.

[98] Jack Hessel, David Mimno, and Lillian Lee. Quantifying the visual concreteness of words and topics in multimodal datasets. *arXiv preprint arXiv:1804.06786*, 2018.

[99] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

[100] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings. *convolutional neural networks and incremental parsing*, 7(1), 2017.

[101] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv:1902.00751*, 2019.

[102] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *In ICLR*, 2022.

[103] Hexiang Hu, Wei-Lun Chao, and Fei Sha. Learning answer embeddings for visual question answering. In *Proceedings of CVPR*, 2018.

[104] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *CVPR*, 2020.

[105] Wen Huang, Hongbin Liu, Minxin Guo, and Neil Zhenqiang Gong. Visual hallucinations of multi-modal large language models. *arXiv preprint arXiv:2402.14683*, 2024.

[106] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

[107] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the CVPR*, 2019.

[108] Peter C. Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, Rachita Chhaparia, Alistair Muldal, Josh Abramson, Petko Georgiev, Alex Goldin, Adam Santoro, and Timothy P. Lillicrap. A data-driven approach for learning to control computers. In *ICML*, 2022.

[109] Taichi Iki and Akiko Aizawa. Do berts learn to use browser user interface? exploring multi-step tasks with unified vision-and-language berts. *arXiv preprint arXiv:2203.07828*, 2022.

[110] Inflection AI. Inflection-2. 2023.

[111] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015.

[112] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

[113] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *In IJCV*, 2016.

[114] Harsh Jhamtani and Taylor Berg-Kirkpatrick. Learning to describe differences between pairs of similar images. *In EMNLP*, 2018.

[115] Menglin Jia, Mengyun Shi, Mikhail Sirotenko, Yin Cui, Claire Cardie, Bharath Hariharan, Hartwig Adam, and Serge Belongie. Fashionpedia: Ontology, segmentation, and an attribute localization dataset. In *ECCV*, 2020.

[116] Sheng Jia, Jamie Kiros, and Jimmy Ba. Dom-q-net: Grounded rl on structured language. *In ICLR*, 2019.

[117] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. Mantis: Interleaved multi-image instruction tuning. *arXiv preprint arXiv:2405.01483*, 2024.

[118] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0.1: the winning entry to the VQA challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

[119] Chenchen Jing, Yuwei Wu, Xiaoxun Zhang, Yunde Jia, and Qi Wu. Overcoming language priors in vqa via decomposed linguistic representations. In *Proceedings of AAAI*, 2020.

[120] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *Proceedings of CVPR*, 2017.

[121] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of CVPR*, 2015.

[122] Kushal Kafle, Mohammed Yousefhussien, and Christopher Kanan. Data augmentation for visual question answering. In *Proceedings of International Conference on Natural Language Generation*, 2017.

[123] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's" up" with vision-language models? investigating their struggle with spatial reasoning. *arXiv preprint arXiv:2310.19785*, 2023.

[124] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P Xing. Rethinking knowledge graph propagation for zero-shot learning. In *CVPR*, 2019.

[125] Yash Kant, Dhruv Batra, Peter Anderson, Alexander Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. Spatially aware multimodal transformers for TextVQA. In *ECCV*, 2020.

[126] Yash Kant, Abhinav Moudgil, Dhruv Batra, Devi Parikh, and Harsh Agrawal. Contrast and classify: Alternate training for robust vqa. In *Proceedings of ICCV*, 2021.

[127] Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher D. Manning. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *ACL*, 2021.

[128] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *ICDAR*, 2015.

[129] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *ICDAR*, 2013.

[130] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of CVPR*, 2015.

[131] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.

[132] Jihyung Kil, Farideh Tavazoee, Dongyeop Kang, and Joo-Kyung Kim. Ii-mmr: Identifying and improving multi-modal multi-hop reasoning in visual question answering. *In ACL Findings*, 2024.

[133] Geunwoo Kim, Pierre Baldi, and Stephen McAleer. Language models can solve computer tasks. *In NeurIPS*, 2023.

[134] Hyounghun Kim and Mohit Bansal. Improving visual question answering by referring to generated paragraph captions. *Proceedings of ACL*, 2019.

[135] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, 2015.

[136] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[137] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *CVPR*, 2017.

[138] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*, 2019.

[139] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github. com/openimages*, 2017.

[140] Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Information maximizing visual question generation. In *Proceedings of CVPR*, 2019.

[141] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[142] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[143] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36, 2024.

[144] Yannick Le Cacheux, Adrian Popescu, and Herve Le Borgne. Webly supervised semantic embeddings for large scale zero-shot learning. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[145] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *ICML*, 2023.

[146] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023.

[147] Bingjia Li, Jie Wang, Minyi Zhao, and Shuigeng Zhou. Two-stage multimodality fusion for high-performance text-based visual question answering. In *ACCV*, 2022.

[148] Gang Li and Yang Li. Spotlight: Mobile ui understanding using vision-language models with a focus. *In ICLR*, 2023.

[149] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *ICCV*, 2017.

[150] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[151] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[152] Ling Li and Hsuan-Tien Lin. Ordinal regression by extended binary classification. *In NeurIPS*, 2006.

[153] Linjie Li, Zhe Gan, and Jingjing Liu. A closer look at the robustness of vision-and-language pre-trained models. *arXiv preprint arXiv:2012.08673*, 2020.

[154] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.

[155] Tianhong Li, Sangnie Bhardwaj, Yonglong Tian, Han Zhang, Jarred Barber, Dina Katabi, Guillaume Lajoie, Huiwen Chang, and Dilip Krishnan. Leveraging unpaired data for vision-language generative models via cycle consistency. *arXiv preprint arXiv:2310.03734*, 2023.

[156] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[157] Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. Mapping natural language instructions to mobile ui action sequences. *In ACL*, 2020.

[158] Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget captioning: generating natural language description for mobile user interface elements. *arXiv preprint arXiv:2010.04295*, 2020.

[159] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022.

[160] Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou. Visual question generation as dual task of visual question answering. In *Proceedings of CVPR*, 2018.

[161] Zujie Liang, Weitao Jiang, Haifeng Hu, and Jiaying Zhu. Learning to contrast the counterfactual samples for robust visual question answering. In *Proceedings of EMNLP*, 2020.

[162] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004.

[163] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. *In CVPR*, 2024.

[164] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[165] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of ECCV*, 2014.

[166] Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration. *In ICLR*, 2018.

[167] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2023.

[168] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[169] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *In NeurIPS*, 2024.

[170] Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding? *arXiv preprint arXiv:2404.05955*, 2024.

[171] Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9273–9281, 2020.

[172] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[173] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *CVPR*, 2018.

[174] Yuliang Liu, Zhang Li, Hongliang Li, Wenwen Yu, Mingxin Huang, Dezhi Peng, Mingyu Liu, Mingrui Chen, Chunyuan Li, Lianwen Jin, et al. On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv:2305.07895*, 2023.

[175] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *In ICCV*, 2015.

[176] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[177] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[178] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[179] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *In ICLR*, 2024.

[180] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *In NeurIPS*, 2022.

[181] Xiaopeng Lu, Zhen Fan, Yansen Wang, Jean Oh, and Carolyn P Rosé. Localize, group, and select: Boosting text-vqa by scene text modeling. In *ICCV*, 2021.

[182] Yao Lu. Unsupervised learning on neural network outputs: with application in zero-shot learning. *arXiv preprint arXiv:1506.00990*, 2015.

[183] Siwen Luo, Feiqi Cao, Felipe Nunez, Zean Wen, Josiah Poon, and Caren Han. Scenegate: Scene-graph based co-attention networks for text visual question answering. *arXiv preprint arXiv:2212.08283*, 2022.

[184] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *In ACL Findings*, 2022.

[185] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *WACV*, 2022.

[186] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021.

[187] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[188] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[189] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[190] Anand Mishra, Karteek Alahari, and C.V. Jawahar. Image retrieval using textual cues. In *ICCV*, 2013.

[191] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019.

[192] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of CVPR*, 2021.

[193] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *CVPR*, 2016.

[194] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[195] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[196] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[197] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *ICCV*, 2023.

[198] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[199] Devi Parikh and Kristen Grauman. Relative attributes. In *ICCV*, 2011.

[200] Badri Patro and Vinay P Namboodiri. Differential attention for visual question answering. In *Proceedings of CVPR*, 2018.

[201] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.

[202] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[203] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[204] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, et al. Combined scaling for zero-shot transfer learning. *Neurocomputing*, 555:126658, 2023.

[205] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *CVPR*, 2021.

[206] Ruizhi Qiao, Lingqiao Liu, Chunhua Shen, and Anton Van Den Hengel. Less is more: zero-shot learning from online textual documents with noise suppression. In *CVPR*, 2016.

[207] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[208] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *In NeurIPS*, 2024.

[209] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *In JMLR*, 2020.

[210] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. In *Proceedings of NeurIPS*, 2018.

[211] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[212] Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. Sunny and dark outside?! improving answer consistency in vqa through entailed question generation. In *Proceedings of EMNLP*, 2019.

[213] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016.

[214] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[215] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[216] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015.

[217] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[218] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016.

[219] Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of ACL*, 2019.

[220] Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, et al. Scaling up models and data with `t5x` and `seqio`. *arXiv preprint arXiv:2203.17189*, 2022.

[221] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.

[222] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[223] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *CVPR*, 2019.

[224] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[225] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *ECCV*, 2022.

[226] Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. Taking a HINT: Leveraging explanations to make vision and language models more grounded. In *Proceedings of ICCV*, 2019.

[227] Ramprasaath R Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Tulio Ribeiro, Besmira Nushi, and Ece Kamar. SQuINTing at VQA Models: Introspecting vqa models with sub-questions. In *Proceedings of CVPR*, 2020.

[228] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *Proceedings of CVPR*, 2019.

[229] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[230] Peter Shaw, Mandar Joshi, James Cohan, Jonathan Berant, Panupong Pasupat, Hexiang Hu, Urvashi Khandelwal, Kenton Lee, and Kristina Toutanova. From pixels to ui actions: Learning to follow instructions via graphical user interfaces. *In NeurIPS*, 2023.

[231] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *ICLR*, 2018.

[232] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of CVPR*, 2019.

[233] Robik Shrestha, Kushal Kafle, and Christopher Kanan. A negative case analysis of visual grounding methods for vqa. In *Proceedings of ACL*, 2020.

[234] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *CVPR*, 2020.

[235] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *ECCV*, 2020.

[236] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[237] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019.

[238] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[239] Chan Hee Song, Jihyung Kil, Tai-Yu Pan, Brian M Sadler, Wei-Lun Chao, and Yu Su. One step at a time: Long-horizon vision-and-language navigation with milestones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15482–15491, 2022.

[240] Abishek Sridhar, Robert Lo, Frank F Xu, Hao Zhu, and Shuyan Zhou. Hierarchical prompting assists large language model on web navigation. *In EMNLP*, 2023.

[241] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Journal of Web Semantics*, 6(3):203–217, 2008.

[242] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *In ACL*, 2019.

[243] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

[244] Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. Meta-gui: Towards multi-modal conversational agents on mobile gui. *In EMNLP*, 2022.

[245] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*, 2023.

[246] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *EMNLP*, 2020.

[247] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[248] Ruixue Tang, Chao Ma, Wei Emma Zhang, Qi Wu, and Xiaokang Yang. Semantic equivalent adversarial data augmentation for visual question answering. In *Proceedings of ECCV*, 2020.

[249] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[250] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[251] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[252] Damien Teney and Anton van den Hengel. Actively seeking and learning from live data. In *Proceedings of CVPR*, 2019.

[253] Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel. On the value of out-of-distribution testing: An example of goodhart's law. In *Proceedings of NeurIPS*, 2020.

[254] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024.

[255] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[256] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[257] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[258] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017.

[259] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[260] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.

[261] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*, 2011.

[262] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[263] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, 2021.

[264] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. GIT: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.

[265] Jun Wang, Mingfei Gao, Yuqian Hu, Ramprasaath R Selvaraju, Chetan Ramaiah, Ran Xu, Joseph F JaJa, and Larry S Davis. Tag: Boosting text-vqa via text-aware visual question-answer generation. *arXiv preprint arXiv:2208.01813*, 2022.

[266] Ning Wang, Jiahao Xie, Jihao Wu, Mingbo Jia, and Linlin Li. Controllable image captioning via prompting. *arXiv preprint arXiv:2212.01803*, 2022.

[267] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

[268] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018.

[269] Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *CVPR*, 2020.

[270] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *In EMNLP*, 2022.

[271] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *ICLR*, 2022.

[272] Zixu Wang, Yishu Miao, and Lucia Specia. Latent variable models for visual question answering. *arXiv preprint arXiv:2101.06399*, 2021.

[273] Spencer Whitehead, Hui Wu, Yi Ren Fung, Heng Ji, Rogerio Feris, and Kate Saenko. Learning from lexical perturbations for consistent visual question answering. *arXiv preprint arXiv:2011.13406*, 2020.

[274] Jialin Wu, Zeyuan Hu, and Raymond J Mooney. Generating question relevant captions to aid visual question answering. In *Proceedings of ACL*, 2019.

[275] Jialin Wu and Raymond J Mooney. Self-critical reasoning for robust visual question answering. In *Proceedings of NeurIPS*, 2019.

[276] xAI. Grok-1 model card. 2024.

[277] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 41(9):2251–2265, 2018.

[278] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018.

[279] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019.

[280] Guo-Sen Xie, Li Liu, Xiaobo Jin, Fan Zhu, Zheng Zhang, Jie Qin, Yazhou Yao, and Ling Shao. Attentive region embedding network for zero-shot learning. In *CVPR*, 2019.

[281] Dongsheng Xu, Qingbao Huang, and Yi Cai. Device: Depth and visual concepts aware transformer for textcaps. *arXiv preprint arXiv:2302.01540*, 2023.

[282] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *NAACL*, 2021.

[283] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, 2015.

[284] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. TAP: Text-aware pre-training for text-vqa and text-caption. In *CVPR*, 2021.

[285] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *In NeurIPS*, 2022.

[286] Pengcheng Yin and Graham Neubig. A syntactic neural model for general-purpose code generation. In *The 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada, July 2017.

[287] Zhou Yu, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao. Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE Transactions on Neural Networks and Learning Systems*, (99):1–13, 2018.

[288] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *In CVPR*, 2024.

[289] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022.

[290] Amrapali Zaveri, Dimitris Kontokostas, Mohamed A Sherif, Lorenz Bühmann, Mohamed Morsey, Sören Auer, and Jens Lehmann. User-driven quality evaluation of dbpedia. In *Proceedings of the 9th International Conference on Semantic Systems*, 2013.

[291] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[292] Xiaoyu Zeng, Yanan Wang, Tai-Yin Chiu, Nilavra Bhattacharya, and Danna Gurari. Vision skills needed to answer visual questions. *Proceedings of the ACM on Human-Computer Interaction*, 2020.

[293] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022.

[294] Cheng Zhang, Wei-Lun Chao, and Dong Xuan. An empirical study on leveraging scene graphs for visual question answering. In *Proceedings of BMVC*, 2019.

[295] Cheng Zhang, Tai-Yu Pan, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. MosaicOS: A simple and effective use of object-centric images for long-tailed object detection. In *Proceedings of ICCV*, 2021.

[296] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *In NeurIPS*, 36, 2024.

[297] Zicheng Zhang, Haoning Wu, Erli Zhang, Guangtao Zhai, and Weisi Lin. A benchmark for multi-modal foundation models on low-level vision: from single images to pairs. *arXiv preprint arXiv:2402.07116*, 2024.

[298] Bo Zhao, Yanwei Fu, Rui Liang, Jiahong Wu, Yonggang Wang, and Yizhou Wang. A large-scale attribute dataset for zero-shot learning. In *CVPRW*, 2019.

[299] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024.

[300] Yaoyao Zhong, Junbin Xiao, Wei Ji, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. *arXiv preprint arXiv:2203.01225*, 2022.

[301] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *In ICLR*, 2024.

[302] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Generalized zero-shot recognition based on visually semantic embedding. In *CVPR*, 2019.

[303] Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. Multimodal c4: An open, billion-scale corpus of images interleaved with text. *Advances in Neural Information Processing Systems*, 36, 2024.

[304] Xi Zhu, Zhendong Mao, Chunxiao Liu, Peng Zhang, Bin Wang, and Yongdong Zhang. Overcoming language priors with self-supervised learning for visual question answering. In *Proceedings of IJCAI*, 2020.

[305] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *CVPR*, 2014.

[306] Yizhe Zhu, Mohamed Elhoseiny, Bingchen Liu, Xi Peng, and Ahmed Elgammal. A generative adversarial approach for zero-shot learning from noisy texts. In *CVPR*, 2018.

[307] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7W: Grounded question answering in images. In *Proceedings of CVPR*, 2016.

[308] Peiqin Zhuang, Yali Wang, and Yu Qiao. Wildfish++: A comprehensive fish benchmark for multimedia research. *IEEE Transactions on Multimedia*, 23:3603–3617, 2020.