

UC Santa Barbara

UC Santa Barbara Electronic Theses and Dissertations

Title

Towards Collaborative Generative AI for Vision-and-Language Studies

Permalink

<https://escholarship.org/uc/item/0xd1437c>

Author

Zhu, Wanrong

Publication Date

2024

Peer reviewed|Thesis/dissertation

University of California
Santa Barbara

Towards Collaborative Generative AI for Vision-and-Language Studies

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

Doctor of Philosophy
in
Computer Science

by

Wanrong Zhu

Committee in charge:

Professor William Yang Wang, Chair
Professor Miguel Eckstein
Professor Xin Eric Wang

June 2024

The Dissertation of Wanrong Zhu is approved.

Professor Miguel Eckstein

Professor Xin Eric Wang

Professor William Yang Wang, Committee Chair

June 2024

Towards Collaborative Generative AI for Vision-and-Language Studies

Copyright © 2024

by

Wanrong Zhu

To my family and friends, and all my loved ones.

Thank you for your unwavering love, faith, and support,
always ensuring I never felt alone on this long journey.

Acknowledgements

I would first like to express my deepest gratitude to my incredibly supportive advisor, William Yang Wang, without whom this dissertation would not have been possible. With his foresight, William guided me to explore vision-and-language problems, which became a topic of great interest throughout my graduate years. As my advisor, he not only shared his insights but also continuously encouraged me to aim higher, greatly benefiting my entire graduate experience.

I would also like to thank my co-advisors, Xin Eric Wang and Miguel Eckstein. Xin generously shared his hands-on research experience during my junior years and was always responsive and open to discussions whenever I needed it. Miguel contributed interdisciplinary ideas, combining computer science and psychology, often inspiring new research topics from a fresh perspective.

Beyond my on-campus advisors, I extend my sincere gratitude to my internship mentors: Pradyumna Narayana at Google AdsAI, Bo Pang and Ashish Thapliyal at Google Research, Jack Hessel and Youngjae Yu at the Allen Institute for AI, and Jennifer Healey and Ruiyi Zhang at Adobe Research. My summers were enriched by their warm hospitality and insightful guidance. I also thank Zhiting Hu, my first research advisor during my internship at CMU LTI. He demonstrated the attitude of a dedicated researcher and served as a role model, inspiring me to pursue my PhD.

I am grateful to my amazing collaborators, with whom I shared inspiration and explored various research topics with great enthusiasm: Arjun Reddy Akula, Anas Awadalla, Hritik Bansal, Sugato Basu, Yejin Choi, Jesse Dodge, Alex Fang, Weixi Feng, Tsu-Jui Fu, Samir Yitzhak Gadre, Irena Gao, Xuehai He, Varun Jampani, Zekun Li, Yujie Lu, Yuankai Qi, Ludwig Schmidt, Raphael Schumann, Kazoo Sone, Radu Soricut, Xinyi Wang, Qi Wu, Wenda Xu, An Yan, Xianjun Yang, and Zhengyuan Yang.

Furthermore, I cherish the enjoyable and unforgettable memories with my friends, and I thank them for their generous help and support throughout this journey: Zhiyu Chen, Ning Duan, Zechao Huang, Xin Jiang, Sharon Levy, Heyun Li, Jiachen Li, Shiyang Li, Chong Liu, Yifan Lu, Yihai Long, Alex Mei, Deepak Nathani, Jing Qian, Yujie Quan, Michael Saxon, Mu Jiun Shie, Hao Tan, Yi-Lin Tuan, Hong Wang, Zeyu Wang, Yue Wei, Zhisheng Ye, Ming Yin, Hanwen Zha, and Congying Zhang.

Last but not least, I sincerely want to thank my family for their unconditional love and support. My mom, dad, and younger sister have always been there for me, celebrating my every achievement, no matter how small, and steadfastly believing in me during difficult times. I am also incredibly fortunate to have met my partner, Xiyu Zhou, who has been by my side throughout this journey. He brings joy and laughter into my life and supports me with both his words and actions. His companionship has been crucial to my emotional well-being.

Curriculum Vitæ

Wanrong Zhu

Education

- Sep. 2019 - June 2024 **University of California, Santa Barbara**, USA
Ph.D. in Computer Science
- Sep. 2015 - July 2019 **Peking University**, China
B.S. in Computer Science

Publications

1. Raphael Schumann, **Wanrong Zhu**, Weixi Feng, Tsu-Jui Fu, Stefan Riezler, William Yang Wang. “VELMA: Verbalization Embodiment of LLM Agents for Vision and Language Navigation in Street View”. *The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2024)*.
2. **Wanrong Zhu***, Jack Hessel*, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, Yejin Choi. “Multimodal C4: An Open, Billion-scale Corpus of Images Interleaved With Text”. *The Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS Datasets & Benchmarks Track 2023)*. (*equal contributions)
3. Yonatan Bitton*, Hritik Bansal*, Jack Hessel*, Rulin Shao, **Wanrong Zhu**, Anas Awadalla, Josh Gardner, Rohan Taori, Ludwig Schimdt. “VisIT-Bench: A Benchmark for Vision-Language Instruction Following Inspired by Real-World Use”. *The Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS Datasets & Benchmarks Track 2023)*.
4. Weixi Feng*, **Wanrong Zhu***, Tsu-Jui Fu, Varun Jampani, Arjun Reddy Akula, Xuehai He, Sugato Basu, Xin Eric Wang, William Yang Wang. “LayoutGPT: Compositional Visual Planning and Generation with Large Language Models”. *The Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023)*. (*equal contributions)
5. Xinyi Wang, **Wanrong Zhu**, Michael Saxon, Mark Steyvers, William Yang Wang. “Large Language Models Are Implicitly Topic Models: Explaining and Finding Good Demonstrations for In-Context Learning”. *The Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023)*.
6. **Wanrong Zhu**, Xinyi Wang, Yujie Lu, Tsu-Jui Fu, Xin Eric Wang, Miguel Eckstein, William Yang Wang. “Collaborative Generative AI: Integrating GPT-k for Efficient Editing in Text-to-Image Generation”. *The 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023, Short)*.

7. **Wanrong Zhu**, An Yan, Yujie Lu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, William Yang Wang. “Visualize Before You Write: Imagination-Guided Open-Ended Text Generation”. *The 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023, Findings)*.
8. **Wanrong Zhu**, Xin Eric Wang, An Yan, Miguel Eckstein, William Yang Wang. “ImaginE: An Imagination-Based Automatic Evaluation Metric for Natural Language Generation”. *The 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023, Findings)*.
9. Yujie Lu, Weixi Feng, **Wanrong Zhu**, Wenda Xu, Xin Eric Wang, Miguel Eckstein, William Yang Wang. “Neuro-Symbolic Causal Language Planning with Commonsense Prompting”. *The 11th International Conference on Learning Representations (ICLR 2023, Spotlight)*.
10. **Wanrong Zhu**, Bo Pang, Ashish Thapliyal, William Yang Wang, Radu Soricut. “End-to-end Dense Video Captioning as Sequence Generation”. *The 29th International Conference on Computational Linguistics (COLING 2022)*.
11. Yujie Lu, **Wanrong Zhu**, Xin Eric Wang, Miguel Eckstein, William Yang Wang. “Imagination-Augmented Natural Language Understanding”. *The 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2022, Oral)*.
12. **Wanrong Zhu**, Yuankai Qi, Pradyumna Narayana, Kazoo Sone, Sugato Basu, Xin Eric Wang, Qi Wu, Miguel Eckstein, William Yang Wang. “Diagnosing Vision-and-Language Navigation: What Really Matters”. *The 2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2022, Oral)*.
13. **Wanrong Zhu**, Xin Eric Wang, Tsu-Jui Fu, An Yan, Pradyumna Narayana, Kazoo Sone, Sugato Basu, William Yang Wang. “Multimodal Text Style Transfer for Outdoor Vision-and-Language Navigation”. *The 16th conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*.
14. **Wanrong Zhu**, Xin Eric Wang, Pradyumna Narayana, Kazoo Sone, Sugato Basu, William Wang. “Towards Understanding Sample Variance in Visually Grounded Language Generation: Evaluations and Observations”. *The 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020, Short)*.
15. Zhiting Hu, Haoran Shi, Bowen Tan, Wentao Wang, Zichao Yang, Tiancheng Zhao, Junxian He, Lianhui Qin, Di Wang, Xuezhe Ma, Zhengzhong Liu, Xiaodan Liang, **Wanrong Zhu**, Devendra Singh Sachan, Eric P. Xing. “Texar: A Modularized, Versatile, and Extensible Toolkit for Text Generation”. *The 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, (ACL 2019)*.

Preprints & Technical Reports

1. Xuehai He, Weixi Feng, Kaizhi Zheng, Yujie Lu, **Wanrong Zhu**, Jiachen Li, Yue Fan, Jianfeng Wang, Linjie Li, Zhengyuan Yang, Kevin Lin, William Yang Wang, Lijuan Wang, Xin Eric Wang. “WorldEval: A Multi-discipline Multi-facet Video Understanding Benchmark Towards World Model Evaluation”. *arXiv preprint*.
2. An Yan, Zhengyuan Yang, Junda Wu, **Wanrong Zhu**, Jianwei Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Julian McAuley, Jianfeng Gao, Lijuan Wang. “List Items One by One: A New Data Source and Learning Paradigm for Multimodal LLMs”. *arXiv preprint*.
3. **Wanrong Zhu**, Jennifer Healey, Ruiyi Zhang, William Yang Wang, Tong Sun. “Automatic Layout Planning for Visually-Rich Documents with Instruction-Following Models”. *arXiv preprint*.
4. An Yan, Zhengyuan Yang, **Wanrong Zhu**, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, Zicheng Liu, Lijuan Wang. “GPT-4V in Wonderland: Large Multimodal Models for Zero-Shot Smartphone GUI Navigation”. *arXiv preprint*.
5. Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, **Wanrong Zhu**, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, Ludwig Schmidt. “OpenFlamingo: An Open-Source Framework for Training Large Autoregressive Vision-Language Models”. *Technical Report*.
6. Yujie Lu, Pan Lu, Zhiyu Chen, **Wanrong Zhu**, Xin Eric Wang, William Yang Wang. “Multimodal Procedural Planning via Dual Text-Image Prompting”. *arXiv preprint*.
7. An Yan, Jiacheng Li, **Wanrong Zhu**, Yujie Lu, William Yang Wang, Julian McAuley. “CLIP also Understands Text: Prompting CLIP for Phrase Understanding”. *arXiv preprint*.
8. **Wanrong Zhu**, Zhiting Hu, Eric P. Xing. “Text Infilling”. *arXiv preprint*.

Awards and Honors

- Rising Star in Machine Learning by University of Maryland Nov. 2023
- Regent’s Fellowship, UC Santa Barbara Sep. 2019

Abstract

Towards Collaborative Generative AI for Vision-and-Language Studies

by

Wanrong Zhu

In recent years, the field of vision-and-language studies has witnessed significant advancements, aiming to bridge the gap between visual perception and linguistic understanding. These studies have explored various approaches to enhance the capabilities of AI systems in generating natural language or visual content, understanding multimodal scenarios, and conducting commonsense reasoning. Despite these advancements, there remains a crucial need for further progress to enable more collaborative and comprehensive interactions between vision and language modalities. This dissertation addresses this need through three primary contributions:

First, I introduce the concept of machine imagination for natural language processing studies. Specifically, I present the use of visual information generated by machines for the automatic evaluation of natural language generation, natural language understanding, and natural language generation.

Second, I explore the utilization of large language models (LLMs) to enhance the performance of vision and multimodal tasks. In particular, I examine the effectiveness of applying LLMs for prompt editing in text-to-image generation, compositional layout planning and generation, and vision-and-language navigation.

Third, I outline my contributions to publicly available open-source vision-and-language research. Specifically, we introduce Multimodal C4, a large-scale multimodal dataset containing interleaved images and text, which we used to train the large-scale multimodal model OpenFlamingo. Additionally, we introduce VisIT-Bench, a public benchmark for

evaluating instruction-following vision-language models in real-world applications.

This dissertation aims to push the boundaries of vision-and-language integration, providing new insights and tools for developing more sophisticated AI systems capable of seamless multimodal interactions.

Contents

Curriculum Vitae	vii
Abstract	x
1 Introduction	1
1.1 Overview	1
1.2 Machine Imagination for NLP Studies	3
1.3 LLM for Vision and Multimodal Studies	5
1.4 Towards Publicly-Available Vision-and-Language Studies	7
Part I Machine Imagination for Natural Language Processing Studies	9
2 Imagination-Augmented Natural Language Understanding	10
2.1 Introduction	10
2.2 Related Work	13
2.3 Our Approach	14
2.4 Experiments	19
3 Imagination-Guided Open-Ended Text Generation	27
3.1 Introduction	27
3.2 Related Work	29
3.3 Method	31
3.4 Experimental Setup	35
3.5 Result and Analysis	38
4 An Imagination-Based Automatic Evaluation Metric for Natural Language Generation	46
4.1 Introduction	46
4.2 Related Work	49
4.3 IMAGINE	50

4.4	Experimental Setup	54
4.5	Results and Analysis	56
Part II LLM for Vision and Multimodal Studies		65
5	Integrating GPT-<i>k</i> for Efficient Editing in Text-to-Image Generation	66
5.1	Introduction	66
5.2	Research Questions and Settings	68
5.3	Prompting T2I w/ GPT- <i>k</i>	70
5.4	Human’s Common Edits vs. GPT- <i>k</i> ’s	72
5.5	Ablations & Analyses	74
6	Compositional Visual Planning and Generation with Large Language Models	77
6.1	Introduction	77
6.2	Related Work	80
6.3	Method	82
6.4	LayoutGPT for Text-Conditioned Image Synthesis	85
6.5	LayoutGPT for Indoor Scene Synthesis	94
7	Verbalization Embodiment of LLM Agents for Vision and Language Navigation	99
7.1	Introduction	99
7.2	Related Work	101
7.3	Urban VLN Environment	102
7.4	Navigation Task	105
7.5	LLM Agent	107
7.6	Experiments	111
Part III Towards Publicly-Available Vision-and-Language Studies		118
8	Multimodal C4: An Open, Billion-scale Corpus of Images Interleaved With Text	119
8.1	Introduction	119
8.2	Related Dataset Work	120
8.3	Data Curation Process	122
8.4	Exploring mmc4	128
8.5	OpenFlamingo: An Early Application of mmc4	131

9	OpenFlamingo: A Framework for Training Autoregressive Vision-Language Models	134
9.1	Introduction	134
9.2	Related work	137
9.3	Approach	138
9.4	Results	144
9.5	Discussion	147
10	VisIT-Bench: A Benchmark for Vision-Language Instruction Following	151
10.1	Introduction	151
10.2	VisIT-Bench: A Real-World Inspired VL Instruction Following Benchmark	155
10.3	VisIT-Bench Analysis	160
10.4	Experiments	162
10.5	Related Work	169
	Bibliography	172

Chapter 1

Introduction

1.1 Overview

Human cognition is inherently multimodal, simultaneously engaging senses such as vision and sound to perceive the world. Studies in cognitive neuroscience, such as those highlighting neural activation in vision-related brain areas when reading text [1], or those illustrating the close relationship between areas processing linguistic and visual semantic information [2], support this viewpoint. Furthermore, visual imagery has been shown to enhance comprehension during language processing [3].

The intersection of visual perception and linguistic understanding within artificial intelligence has undergone remarkable transformations in recent years. Vision-and-language studies aim to bridge the gap between these two modalities, enhancing the ability of AI systems to generate natural language, create visual content, and perform commonsense reasoning. Despite significant progress, a more integrative and collaborative approach is necessary to foster deeper, more effective interactions between vision and language.

This dissertation is motivated by recent technological advances that allow for the visualization of machine imagination, which is the ability of AI to generate and manip-

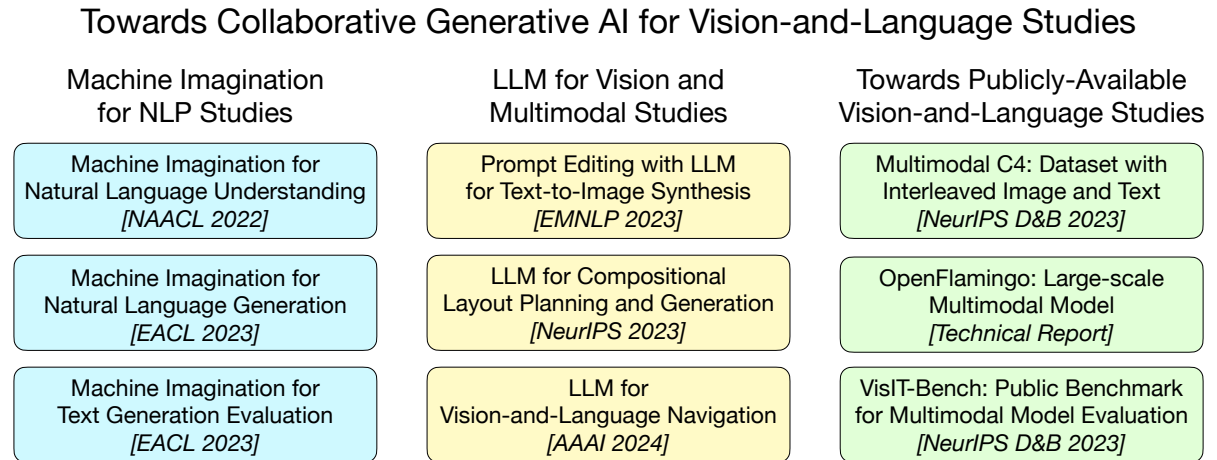


Figure 1.1: Dissertation overview.

ulate images in response to textual context [4, 5, 6, 7]. This capability, combined with the emergence of large language models (LLMs) [8, 9, 10] that provide a wealth of commonsense knowledge interactively, sets the stage for innovative approaches in multimodal studies.

Figure 1.1 shows an overview of this dissertation. The contributions of this dissertation are threefold:

- Introduction of Machine Imagination to NLP Tasks: I explore how machine-generated visual information can enhance traditional NLP tasks, bridging the gap between textual data and visual understanding to improve automatic evaluations in natural language generation and understanding.
- Utilization of Large Language Models for Vision and Multimodal Studies: The capabilities of LLMs are harnessed to bolster vision and multimodal tasks. This includes their application in prompt editing for text-to-image generation, compositional layout planning, and navigation within multimodal contexts.
- Development of Multimodal Resources: Recognizing the need for robust datasets

to train sophisticated models, I have developed and made publicly available large-scale vision-and-language datasets, models, and benchmarks. These resources are designed to advance the field by providing the necessary tools for future research and application.

By addressing these areas, the dissertation aims to push forward the boundaries of how AI systems can integrate and leverage multimodal information, providing new pathways for enhancing machine understanding and interaction in a world where visual and linguistic inputs are intertwined.

1.2 Machine Imagination for NLP Studies

In the first part of this dissertation, I focus on introducing machine imagination for natural language processing (NLP) studies. Specifically, I present the use of visual information generated by machines for natural language understanding (NLU) [11], natural language generation (NLG) [12], and the automatic evaluation of NLG [13].

Human brains integrate linguistic and perceptual information simultaneously to understand natural language, and hold the critical ability to render imaginations. Such abilities enable us to construct new abstract concepts or concrete objects, and are essential in involving practical knowledge to solve problems in low-resource scenarios. However, most existing methods for natural language understanding are mainly focused on textual signals. They do not simulate human visual imagination ability, which hinders models from inferring and learning efficiently from limited data samples. In **Chapter 2**, we introduce iACE to solve natural language understanding tasks from a novel learning perspective—imagination-augmented cross-modal understanding. iACE enables visual imagination with external knowledge transferred from the pre-trained generative vision-and-language models. Extensive experiments on GLUE [14] and SWAG [15] show that

iACE achieves consistent improvement over visually-supervised pre-trained models. More importantly, results in extreme and normal few-shot settings validate the effectiveness of iACE in low-resource natural language understanding circumstances.

When generating text, human writers are gifted at creative visualization, which enhances their writings by forming imaginations as blueprints before putting down the stories in words. Inspired by such a cognitive process, we ask the natural question of whether we can endow machines with the same ability to utilize visual information and construct a general picture of the context to guide text generation. In **Chapter 3**, we propose iNLG that uses machine-generated images to guide language models (LM) in open-ended text generation. The experiments and analyses demonstrate the effectiveness of iNLG on open-ended text generation tasks, including text completion, story generation, and concept-to-text generation in both few-shot and full-data scenarios. Both automatic metrics and human evaluations verify that the text snippets generated by our iNLG are coherent and informative while displaying minor degeneration.

Automatic evaluations for natural language generation conventionally rely on token-level or embedding-level comparisons with the text references. This is different from human language processing, for which visual imagination often improves comprehension. In **Chapter 4**, we propose IMAGINE, an imagination-based automatic evaluation metric for natural language generation. With the help of StableDiffusion [4], a state-of-the-art text-to-image generator, we automatically generate an image as the embodied imagination for the text snippet and compute the imagination similarity using contextual embeddings. Experiments spanning several text generation tasks demonstrate that adding machine-generated images with our IMAGINE displays great potential in introducing multi-modal information into NLG evaluation, and improves existing automatic metrics' correlations with human similarity judgments in both reference-based and reference-free evaluation scenarios.

1.3 LLM for Vision and Multimodal Studies

The second part of this dissertation explores the utilization of large language models (LLMs) to enhance the performance of vision and multimodal tasks. In particular, I examine the effectiveness of applying LLMs for prompt editing in text-to-image generation [16], compositional layout planning [17], and vision-and-language navigation [18].

The field of text-to-image (T2I) generation has garnered significant attention both within the research community and among everyday users. Despite the advancements of T2I models, a common issue encountered by users is the need for repetitive editing of input prompts in order to receive a satisfactory image, which is time-consuming and labor-intensive. Given the demonstrated text generation power of large-scale language models, such as GPT- k , we investigate the potential of utilizing such models to improve the prompt editing process for T2I generation in **Chapter 5**. We conduct a series of experiments to compare the common edits made by humans and GPT- k , evaluate the performance of GPT- k in prompting T2I, and examine factors that may influence this process. We found that GPT- k models focus more on inserting modifiers while humans tend to replace words and phrases, which includes changes to the subject matter. Experimental results show that GPT- k are more effective in adjusting modifiers rather than predicting spontaneous changes in the primary subject matters. Adopting the edit suggested by GPT- k models may reduce the percentage of remaining edits by 20-30%.

Attaining a high degree of user controllability in visual generation often requires intricate, fine-grained inputs like layouts. However, such inputs impose a substantial burden on users when compared to simple text inputs. To address the issue, we study how Large Language Models (LLMs) can serve as visual planners by generating layouts from text conditions, and thus collaborate with visual generative models. In **Chapter 6**, we propose LayoutGPT, a method to compose in-context visual demonstrations

in style sheet language to enhance the visual planning skills of LLMs. As a result, LayoutGPT can generate plausible layouts in multiple domains, ranging from 2D images to 3D indoor scenes. LayoutGPT shows superior performance in converting challenging language concepts to layout arrangements for faithful text-to-image generation. When combined with a downstream image generation model, LayoutGPT outperforms text-to-image models/systems by 20-40% and achieves comparable performance as human users in designing visual layouts for numerical and spatial correctness. Lastly, LayoutGPT achieves comparable performance to supervised methods in 3D indoor scene synthesis, demonstrating its effectiveness and potential in multiple visual domains.

Incremental decision making in real-world environments is one of the most challenging tasks in embodied artificial intelligence. One particularly demanding scenario is vision and language navigation which requires visual and natural language understanding as well as spatial and temporal reasoning capabilities. The embodied agent needs to ground its understanding of navigation instructions in observations of a real-world environment like Street View. Despite the impressive results of LLMs in other research areas, it is an ongoing problem of how to best connect them with an interactive visual environment. In **Chapter 7**, we propose VELMA, an embodied LLM agent that uses a verbalization of the trajectory and of visual environment observations as contextual prompt for the next action. Visual information is verbalized by a pipeline that extracts landmarks from the human written navigation instructions and uses CLIP to determine their visibility in the current panorama view. We show that VELMA is able to successfully follow navigation instructions in Street View with only two in-context examples. We further finetune the LLM agent on a few thousand examples and achieve around 25% relative improvement in task completion over the previous state-of-the-art for two datasets.

1.4 Towards Publicly-Available Vision-and-Language Studies

The third part of this dissertation outlines my contributions to publicly available open-source vision-and-language research. Specifically, we introduce Multimodal C4, a large-scale multimodal dataset containing interleaved images and text [19], which we used to train the large-scale multimodal model OpenFlamingo [20]. Additionally, we introduce VisIT-Bench, a public benchmark for evaluating instruction-following vision-language models in real-world applications [21].

In-context vision and language models like Flamingo [22] support arbitrarily interleaved sequences of images and text as input. This format not only enables few-shot learning via interleaving independent supervised (image, text) examples, but also, more complex prompts involving interaction between images, e.g., “What do image A and image B have in common?” To support this interface, pretraining occurs over web corpora that similarly contain interleaved images+text. To date, however, large-scale data of this form have not been publicly available. In **Chapter 8**, we release Multimodal C4 (**mmc4**), an augmentation of the popular text-only **c4** corpus with images interleaved. We use a linear assignment algorithm to place images into longer bodies of text using CLIP features [23], a process that we show outperforms alternatives. **mmc4** spans everyday topics like cooking, travel, technology, etc. A manual inspection of a random sample of documents shows that a vast majority (88%) of images are topically relevant, and that linear assignment frequently selects individual sentences specifically well-aligned with each image (80%). After filtering NSFW images, ads, etc., the resulting **mmc4** corpus consists of 101.2M documents with 571M images interleaved in 43B English tokens.

In **Chapter 9**, we introduce OpenFlamingo, a family of autoregressive vision-language models ranging from 3B to 9B parameters. OpenFlamingo is an ongoing effort to pro-

duce an open-source replication of DeepMind’s Flamingo models [22]. On seven vision-language datasets, OpenFlamingo models average between 80 - 89% of corresponding Flamingo performance. This technical report describes our models, training data, hyper-parameters, and evaluation suite.

In **Chapter 10**, we introduce VisIT-Bench (**V**isual **I**ns**T**ruction **B**enchmark), a benchmark for evaluation of instruction-following vision-language models for real-world use. Our starting point is curating 70 “instruction families” that we envision instruction-tuned vision-language models *should* be able to address. Extending beyond evaluations like VQAv2 and COCO, tasks range from basic recognition to game playing and creative generation. Following curation, our dataset comprises 592 test queries, each with a human-authored instruction-conditioned caption. These descriptions surface instruction-specific factors, e.g., for an instruction asking about the accessibility of a storefront for wheelchair users, the instruction-conditioned caption describes ramps/potential obstacles. These descriptions enable 1) collecting human-verified reference outputs for each instance; and 2) automatic evaluation of candidate multimodal generations using a text-only LLM, aligning with human judgment. We quantify quality gaps between models and references using both human and automatic evaluations; e.g., the top-performing instruction-following model wins against the GPT-4 reference in just 27% of the comparison.

Part I

Machine Imagination for Natural Language Processing Studies

Chapter 2

Imagination-Augmented Natural Language Understanding

2.1 Introduction

Cognitive neuroscience studies reveal neural activation in vision-related brain areas when reading text [1] and show a tight relationship between brain areas processing linguistic and visual semantic information [2]. In addition, visual imagery improves comprehension during human language processing [3]. Such imagination empowers human brains with generalization capability to solve problems with limited supervision or data samples.

However, the field of Natural language Understanding has mainly been focused on building machines based solely on language, ignoring the inherently grounded imagination from the external visual world. These studies either learn text-only representations from language corpora [24, 25, 26] or implicitly involve retrieved visual supervision in pre-trained language models [27]. Thus, their approaches appear limited in transferring the connection between language understanding and visual imagination to downstream tasks,

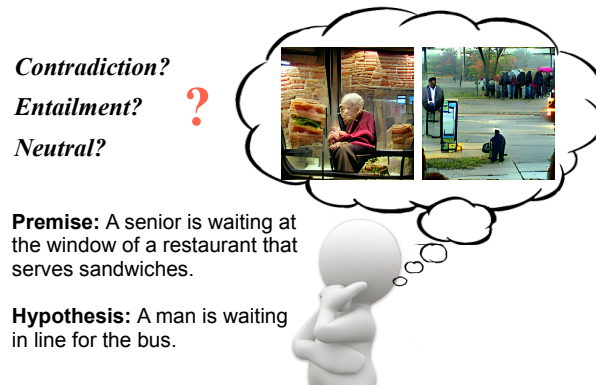


Figure 2.1: Rendering visual imagination is an intuitive way to activate perception for linguistic understanding, e.g. natural language inference.

which are essential to solving low-resource circumstances. In addition, these methods are limited to text-only augmentations, whereas visual imaginations leverage cross-modal augmentations to deal with low-resource situations.

Human brains are multi-modal, integrating linguistic and perceptual information simultaneously. Intuitively, the machines could achieve a higher-level understanding of natural language and better learning transference by imitating the procedure of human imagination behavior.

Inspired by this, we propose to understand language with the integration of linguistic and perceptual information via introducing imagination supervision into text-only NLU tasks. To imitate the imagination-augmented understanding process as shown in Figure 2.1 with text-only data, we devise a procedure with two steps: 1) pre-train a visually-supervised Transformer over paired text and images retrieved from large-scale language corpus and image set, and 2) construct the imagination with a generative model and fine-tune on downstream NLU datasets by learning the paired imagination and natural language in a cross-modal embedding. We show a detailed description of the cross-modal imagination process for a specific Natural Language Inference task in Figure 2.2. In this way, we utilize machine imagination to improve the performance of natural language understanding.

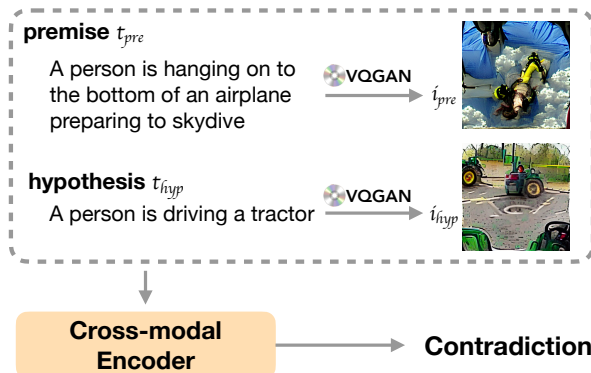


Figure 2.2: A detailed view of our iACE framework fine-tunes on natural language inference task.

We adopt the few-shot learning setting to study the potential of using less human effort of annotation for our proposed iACE to learn the natural language with the help of imagination. Large margin performance gain in both extreme and normal few-shot settings demonstrate the effectiveness of iACE in solving problems with limited data samples. In the full data setting of GLUE [14] and SWAG [15], we observe the consistent performance gain of our proposed iACE over the visually-supervised approach (e.g., VOKEN [27]) upon four language base models (e.g., BERT, RoBERTa).

In summary, the main contributions of our work are as follows:

- We propose to solve the text-only learning problem in natural language understanding tasks from a novel learning perspective: imagination-augmented cross-modal language understanding.
- To address the problem mentioned above, we devise iACE to generate imaginations in a cross-modal representation space to guide the fine-tuning of the visually supervised language models.
- Experimental results in the few-shot setting validate the consistent superiority of iACE over baselines in tackling the low-resource situation. In full settings, iACE maintains the improvement in GLUE and SWAG.

2.2 Related Work

Visually-aided Language Learning Previous research attempt to introduce visual information to improve language learning on various Natural Language Processing (NLP) scenarios, including but not limited to machine translation [28, 29], information retrieval [30, 31], semantic parsing [32, 33], natural language inference [34], bilingual lexicon learning [35, 36], natural language generation evaluation [13], spatial commonsense reasoning [37] and language representation learning [23, 27, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47]. While most of these studies acquire visual information through retrieval from the web or large-scale image sets, a recent line of studies attempt to generate visual supervision from scratch. The visual information can either be provided in the form of representation [40, 48] or concrete images [13, 31]. Though previous studies generate machine imagination, they only tackle specific tasks, such as machine translation [48] or information retrieval [31]. To the best of our knowledge, we are the first to utilize machine abstract imagination from large pretrained vision and language models to improve general NLU tasks. Recently, VOKEN [27] incorporate retrieved token-level visual information into existing transformer models and achieve consistent improvement. iACE is different from this work for two aspects: 1) we explicitly encode visual imagination during fine-tuning. 2) we propose a novel model to borrow knowledge from imagination in both training and inference.

Few-shot Natural Language Understanding Natural Language Understanding (NLU) is a subfield in NLP that involves a broad range of tasks such as question answering, sentiment analysis, and textual entailment. Researchers have collected specific language corpus [14, 15, 49] to train the machines on NLU learning. However, the general language understanding problem remains a challenge. Few-shot learning is a learning

paradigm that aims to predict the correct class of instances with a relatively small amount of labeled training examples [50, 51]. It has been receiving increasing attention for its potential in reducing data collection effort and computational costs and extending to rare cases. To deal with data-scarcity in NLU problems, previous research introduces external knowledge [52], utilizes meta-learning [53, 54, 55] and adopts data augmentation to generate labeled utterances for few-shot classes [56, 57]. Recent studies [8, 9] have shown that large-scale pre-trained language models are able to perform NLU tasks in a few-shot learning manner. The pre-trained multimodal models also display similar few-shot learning ability [58]. Unlike previous studies on pre-trained multimodal Transformers that target solving multimodal tasks, our study introduces imagination from the visual world into language models and aims to improve NLU.

2.3 Our Approach

We illustrate how we solve the existing text-only learning problem in natural language understanding tasks as the Imagination-augmented Cross-modal Language Understanding (ICLU) problems in Section 2.3.1. Then we give a detailed illustration of our proposed iACE’s architecture in Section 2.3.2. Finally, we describe the procedure and training protocol of the perceptual-enhanced linguistic understanding paradigm in Section 2.3.3.

2.3.1 Problem Definition

NLU is concerned with understanding the semantic meaning of the given utterances. Data pieces for NLU can be structured as $(x_{context}, \mathcal{X}, y)$, where $x_{context}$ represents the text context, $\mathcal{X} = \{x_1, x_2, \dots, x_m, m \in \mathbb{N}\}$ denote a set of text snippets, and m denotes the number of text samples for a specific task. The model learns to predict the ground truth label y , which is either regression or a classification label. While NLU is usually

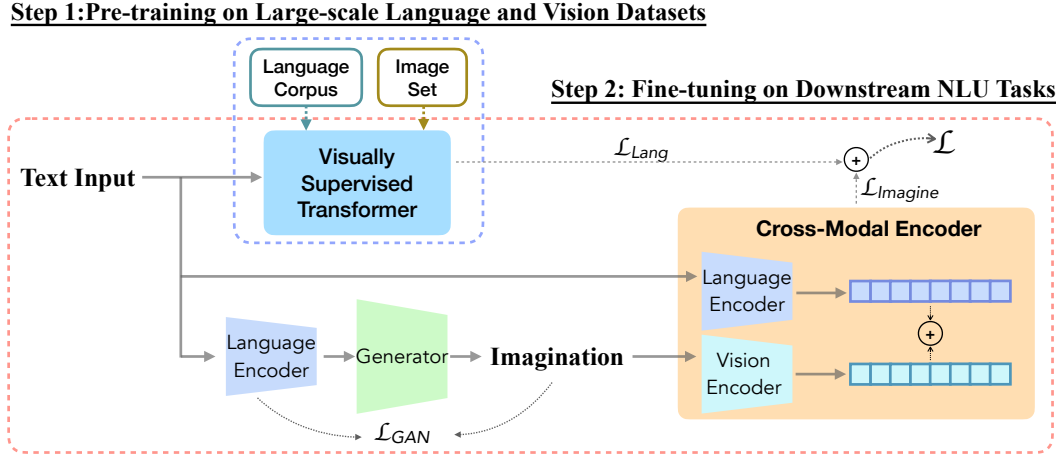


Figure 2.3: **Overview of iACE.** The generator G visualize imaginations close to the encoded texts by minimizing \mathcal{L}_{GAN} . The cross-modal encoder E_c learns imagination-augmented language representation. Two-step learning procedure consists of: 1) pre-train a Transformer with visual supervision from large-scale language corpus and image set, 2) fine-tune the visually supervised pre-trained Transformer and the imagination-augmented cross-modal encoder on downstream tasks.

regarded as a language-only task, we attempt to solve it from a cross-modal perspective by introducing the novel ICLU problem.

In our ICLU problem, data pieces are structured as $(x_{context}, i_{context}, \mathcal{X}, \mathcal{I}, y)$, in which $i_{context}$ represents the visual context related to the text context, and $\mathcal{I} = \{i_1, i_2, \dots, i_n, n \in \mathbb{N}\}$ denotes the imagination set. The “imagination” refers to the images that are visualized from the text. Here, n is the number of visualized sentences for a specific task, which is the same as m by default.

To solve this problem, we devise a novel iACE to construct imagination from textual data and learn the bi-directional alignment between the imagination and text. Specifically, for each piece of text x_j in the sentence set \mathcal{X} , we first follow Radford et al. [23] and Esser et al. [59] and use a generative model to render a descriptive illustration i_j . The visualized imagination will later serve as the visual input in the ICLU problem.

2.3.2 Model Architecture

Overview Figure 2.3 provides an overview of the iACE framework. iACE consists of two modules: 1) the imagination generator G , 2) the imagination-augmented cross-modal encoder E_c . Given the textual sentence $x = \{w_1, w_2, \dots, w_k, k \in \mathbb{N}\}$ (w_j denotes the j -th token in the sentence), G generates corresponding visual imagination i . The cross-modal encoder then encodes x and i as \mathbf{t} and \mathbf{v} , respectively. iACE explicitly provides imagination supervision to the visually-supervised Transformer during fine-tuning on downstream NLU tasks.

Imagination Generator Previous studies introduce visual supervision through retrieval from the web or image sets. However, it is hard to find visuals that perfectly match the topics discussed in each text snippet, especially for the relatively complicated text input for the NLU tasks. Such misalignment between the input text and the retrieved visuals might hinder the model from general language understanding learning. Out of consideration for cross-modal feature alignment, we choose to render specific visualization corresponding to each piece of input text from scratch. Specifically, we construct imagination of the textual input with a large-scale vision and language model guided generative framework - VQGAN+CLIP [60]. For each piece of input text x , we treat it as the prompt and use the VQGAN [59] model to render the imagination i with 128×128 resolution and 200-step optimization. At each optimization step, we use the CLIP [23] model to assess how well the generated image corresponds to the text.

$$\mathcal{L}_{GAN} = 2[\arcsin(\frac{1}{2} \|\mathbf{t} - \mathbf{v}\|)]^2 \quad (2.1)$$

To be specific, CLIP encodes the input text x and the corresponding imagination i as \mathbf{t} and \mathbf{v} , and the training objective is to minimize the distance between \mathbf{t} and \mathbf{v} in the

cross-modal embedding space.

Cross-modal Encoder We adopt CLIP as the cross-modal encoder to encode the input text and the generated imaginations. CLIP [23] is trained on large-scale image-text pairs and is able to align visual and textual input in the embedding space. Specifically, we use the *ViT-B/32* version of Vision Transformer as the image encoder, and Transformer [61] with the architecture modifications described in GPT-2 [9] as the text encoder. For each modality, the self-attention (SA) module is applied to model the regions of imagination or the words of the text as follows:

$$SA(F) = \text{concat}(\text{softmax} \frac{FW_j^Q FW_j^{K^T}}{\sqrt{d_k}} FW_j^V, \dots)W \quad (2.2)$$

where F denotes the set of regions of the imagination or the words of the textual sentence. W_j^Q , W_j^K , and W_j^V represents the weight in the j -th head for query, key and value respectively. d_k is the dimension of the embedding. W is the weight matrix for multiple heads.

To solve the ICLU problem, we learn the bi-directional relationship between the text input and the visualized imagination. We apply late fusion on the text feature \mathbf{t} and visual feature \mathbf{v} to construct the cross-modal feature. Given the set of visual features S_v and textual features S_t , the fused embedding X_S can be given with:

$$X_S = [\text{ReLU}(W_t S_t + b_t), \text{ReLU}(W_j S_v + b_j)] \quad (2.3)$$

where W and b are of two separate fully connected layers to the visual and text embeddings. The fused embeddings X_S will go through two fully connected layers before we receive the final imagination-augmented language representation.

Visually-supervised Transformer We implement the visually-supervised Transformer language model proposed in Tan et al. [27]. The model architecture is a BERT-like pure-language-based masked language model.

2.3.3 Learning Procedure

We introduce a novel paradigm to better understand natural language by incorporating existing language models with visual imagination. As shown in Figure 2.3, the procedure consists of two steps: (1) pre-train the visually-supervised Transformer, and (2) fine-tune the framework with imagination on downstream tasks.

Step 1: Visually-supervised Pre-training We pre-train a visually-supervised Transformer following the scheme proposed in VOKEN [27], which extrapolates cross-modal alignments to language-only data by contextually mapping language tokens to the related images. In addition to masked language modeling, VOKEN proposed a voken classification task: given a set of tokens with masks, the model is asked to predict the best-matching image (the voken) for each tokens. The pre-training loss can be given as:

$$\mathcal{L} = -\lambda_1 \sum_{w_j \in \hat{s}} \log q_j(w_j | \check{s}) - \lambda_2 \sum_{w_j \in \hat{s}} \log p_j(v(w_j; s) | \check{s}) \quad (2.4)$$

Here s is the token set, \hat{s} is the masked tokens, and \check{s} is the unmasked tokens. The q_j and p_j represent the conditional probability distribution of the j -th token given the token w_j and voken $v(w_j; s)$ respectively, and λ_1 and λ_2 are the balance factor of the masked language modeling task and the voken-classification task. The cross-modal classification task enables the model to learn the matching between the tokens from the language corpus (e.g., wiki) and its most-related images from the image set (e.g., MSCOCO).

Step 2: Imagination-augmented Fine-tuning We use GLUE [14] and SWAG [15] as the downstream datasets in the following sections. Our proposed iACE learns to minimize the cross-entropy loss below:

$$\mathcal{L}_{Imagine} = - \sum_{j=1}^{|D|} \sum_{k=1}^K y_k \log p_k(d_j(\mathbf{t}; \mathbf{v})|D) \quad (2.5)$$

where j denotes the j -th data sample in dataset D , and K as the class number. The p_k represents the conditional probability distribution of d_j . During fine-tuning, the visually-supervised Transformer language model only relied on the textual input to make predictions. The loss are computed as:

$$\mathcal{L}_{Lang} = - \sum_{j=1}^{|D|} \sum_{k=1}^K y_k \log p_k(d_j(\mathbf{t})|D) \quad (2.6)$$

Notice that we use MSE loss for the regression task. The imagination-augmented loss and pure-language based loss are summed up with a balance factor λ in a jointly training schema as:

$$\mathcal{L} = \lambda \mathcal{L}_{Imagine} + (1 - \lambda) \mathcal{L}_{Lang} \quad (2.7)$$

We use Adam Optimizer with a learning rate $1e - 4$ for the GLUE benchmark and $2e - 5$ for the SWAG dataset. We discuss more details in Section 2.4.

2.4 Experiments

2.4.1 Experimental Setup

Datasets & Metric We conduct experiments to evaluate the performance of our proposed method over SST-2 [62], QNLI [63], QQP [64], MultiNLI [65], MRPC [66], STS-

B [67] from GLUE [14] Benchmark, and SWAG [15] dataset. We construct few-shot setting subsets by taking 0.1%, 0.3%, and 0.5% of training instances as the Extreme Few-shot Setting, and 1%, 3%, and 5% as the Normal Few-shot Setting. We train the model with the subsets and evaluate its performance on the complete development set. We use accuracy as the default evaluation metric and compare such results in the following sections.

Baselines We choose BERT [24] and RoBERTa [26] as the base language models, and apply our iACE framework on top of their small and base architectures for comparison. A recent study proposes a visually-supervised language model VOKEN [27] that introduces visual supervision into language model pre-training by borrowing external knowledge from retrieved images of the tokens. In natural language understanding tasks, VOKEN achieved improvements over language-based baselines BERT and RoBERTa. Thus we also use VOKEN built upon these language-based models as a set of powerful baselines. In the following experiments, each model is first pre-trained with visual supervision introduced in Tan et al. [27] upon the four base models (BERT_{small}, BERT_{base}, RoBERTa_{small} and RoBERTa_{base}). Then the models will be fine-tuned on downstream tasks.

Notice that base models and VOKEN use pure-language training objectives during fine-tuning. Neither of them utilizes the visual signals inherent in the downstream language corpora. In contrast, our iACE explicitly introduces visual imagination supervisions into fine-tuning and inference stages.

Implementation Details We train RoBERTa with the same configurations as a robustly optimized pre-training approach based on BERT of the same size. BERT_{small} has 6 repeating layers, 512 hidden dimension. BERT_{base} has 12 repeating layers, 768 hidden

dimension.

The imagination of the texts is generated interactively by using VQGAN+CLIP, with 128×128 size, 500 iterations. We use pre-trained VQGAN (imagenet_{f16}) and CLIP (ViT-B/32). We leverage CLIP (ViT-B/32) as our language and vision model for premise and hypothesis, and imagination of them. The text and image dimension is 512. The dropout rate is set to 0.1. We use Cross-Entropy loss for our cross-modal classification. Each model was first pre-trained on 4 TITAN RX GPUs for 30 epochs with early stopping and a batch size of 32 and a sequence length of 126. The optimizer used is Adam with a learning rate of $2e - 4$ and a weight decay of 0.01. The models are then fine-tuned on GLUE benchmark and SWAG dataset for 3 epochs with 32 batch size. We adopt the joint training strategy for our proposed iACE and visually supervised transformer during fine-tuning. The learning rate of the Adam optimizer is set as $1e - 4$ and $2e - 5$ for GLUE and SWAG, respectively.

2.4.2 Few-shot Learning Results

We claim that introducing imagination into language processing helps the existing language-based system tackle the low-resource situation. Thus, the automatically generated imagination helps reduce the human effort to annotate textual data. To verify this, we define two situations, a normal few-shot setting, and an extreme few-shot setting. For the normal few-shot setting, we keep 1%, 3%, and 5% of the training dataset for each task in GLUE Benchmark. For the extreme few-shot setting, we keep a lower number of the training dataset, which is reduced to 0.1%, 0.3%, and 0.5% of the training dataset. We train the models with the same configuration under these two settings and compare them with visually supervised transformer baselines to confirm the benefit that our proposed iACE brings to the few-shot situation.

	SST-2			QNLI			QQP			MNLI		
	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%
Extreme Few-shot												
<i>VOKEN(Bert_{base})</i>	54.70	77.98	80.73	50.54	51.60	61.96	44.10	60.65	65.46	37.31	54.62	58.79
<i>iACE(Bert_{base})</i>	77.98	80.96	81.42	51.64	58.33	64.03	49.36	63.67	71.17	40.07	56.49	59.57
<i>VOKEN(Roberta_{base})</i>	70.99	71.10	77.86	54.37	62.23	65.78	62.32	67.25	70.18	48.59	49.76	58.23
<i>iACE(Roberta_{base})</i>	75.34	78.66	83.60	54.79	65.03	65.83	65.43	68.11	70.77	48.94	52.74	59.39
Normal Few-shot	1%	3%	5%	1%	3%	5%	1%	3%	5%	1%	3%	5%
<i>VOKEN(Bert_{base})</i>	81.40	86.01	84.75	64.17	77.36	80.19	72.55	78.37	80.50	60.45	62.73	72.35
<i>iACE(Bert_{base})</i>	82.45	87.04	86.47	65.09	79.54	80.52	74.31	78.69	80.52	62.15	70.43	73.73
<i>VOKEN(Roberta_{base})</i>	83.78	84.08	87.61	75.00	81.16	81.23	73.14	79.09	79.63	63.51	70.68	74.02
<i>iACE(Roberta_{base})</i>	83.83	84.63	89.11	79.35	81.41	81.65	73.72	79.38	79.81	65.66	70.76	74.10

Table 2.1: **Model-agnostic Improvement in Few-shot Setting.** iACE and VOKEN upon BERT and RoBERTa base size architecture are fine-tuned in Extreme Few-shot (0.1%, 0.3%, 0.5%) and Normal Few-shot setting (1%, 3%, 5%). For the few-shot setting, we use large and stable datasets from GLUE Benchmark. We compare accuracy on SST-2, QNLI, QQP, and MNLI and the average of accuracy and F1 score on QQP. **BEST** results are highlighted.

Results of the few-shot setting are reported in Table 2.1. Following Tan et al. [27], we only report the four largest and stable tasks in GLUE for the model-agnostic comparison. We report the accuracy for SST-2, QNLI, MNLI. For QQP and MRPC, we report the average of F1 and accuracy. For SWAG, we report the correlation. We observe that the imagination information remarkably helps with both the normal few-shot curriculum and extreme few-shot curriculum. We assume the imagination-augmented fine-tuning successfully transfers the language understanding from the large-scale vision and language model. Thus iACE achieves consistent performance gain and shows great superiority of generalization and transferring ability.

2.4.3 Ablation Studies

We conduct ablation studies over both the method side and data side to validate their contribution to our proposed iACE.

Base Model	Method	SST-2			QNLI			QQP			MNLi			ALL
		0.1%	1.0%	3.0%	0.1%	1.0%	3.0%	0.1%	1.0%	3.0%	0.1%	1.0%	3.0%	Avg.
BERT _{base}	Direction	49.01	79.59	87.15	51.31	52.55	66.90	56.74	31.58	31.59	32.73	61.54	70.72	55.95
BERT _{base}	Unify	48.96	77.98	86.92	50.54	52.02	67.20	55.29	56.93	79.09	39.05	63.29	70.86	62.34
BERT _{base}	iACE	77.98	82.45	87.04	51.64	65.09	79.54	49.36	74.31	78.69	40.07	62.15	70.43	68.23
RoBERTa _{base}	Direction	72.71	80.38	84.63	54.91	74.68	78.58	61.57	74.68	31.59	32.95	61.96	70.62	64.94
RoBERTa _{base}	Unify	75.11	80.04	88.07	53.62	74.64	78.47	64.94	74.85	76.84	51.12	65.42	70.74	71.15
RoBERTa _{base}	iACE	75.34	83.83	84.63	54.79	79.35	81.41	65.43	73.72	79.38	48.94	65.66	70.76	71.93

Table 2.2: **Method Design Ablation in Few-shot Setting.** We compare the results of two variants over 0.1%, 1.0%, 3.0% of SST-2, QNLI, QQP and MNLi dataset. Details of *Direction* and *Unify* are illustrated in Section 2.4.3.

Method Design Ablation Two method variants of our imagination-augmented encoder are built as baselines to validate the importance of our bi-directional cross-modal imagination design in iACE. The variants are built upon RoBERTa_{base} and BERT_{base} base models. Specifically, we develop variant *Direction* and *Unify*. *Direction* represent alignment between text input and imagination into a directional embedding as $\text{FUSE}(\mathbf{t}_{sen1} - \mathbf{i}_{sen1}, \mathbf{t}_{sen2} - \mathbf{i}_{sen2})$. *Unify* encode the text and imagination, considering the direction from vision to language by encoding as $\text{FUSE}(\mathbf{t}_{sent1}, \mathbf{t}_{sent2}, \mathbf{i}_{sent1}, \mathbf{i}_{sent2})$. While *iACE* consider direction from visoin to language and language to vision by encoding as the combination of $\text{FUSE}(\mathbf{t}_{sent1}, \mathbf{i}_{sent2})$ and $\text{FUSE}(\mathbf{i}_{sent1}, \mathbf{t}_{sent2})$. As shown in Table 2.2, our bi-directional imagination and language learning achieve stable and best average performance. These results indicate that our bi-directional imagination method design obtain generalization and transferring ability. We assume iACE benefits from both learning from language to vision and learning from vision to language simultaneously.

Imagination Composition Ablation The composition of the imagination is essential for the performance. To further study the importance of full imagination, we ablate the data side by constructing a textual-only model denoted as *Textual Only*, a visual-only imagination denoted as *Visual Only* and a single directional imagination input denoted as *Visual+Textual*. *Visual Only* and *Visual+Textual* represent the imagination model

Base Model	Composition	Extreme Few-shot (0.1%)				Normal Few-shot (3.0%)				ALL
		SST-2	QNLI	QQP	MNLI	SST-2	QNLI	QQP	MNLI	Avg.
BERT _{base}	Textual-Only	49.08	50.54	55.48	38.82	87.50	67.05	77.42	71.00	62.11
BERT _{base}	Visual-Only	59.97	50.56	49.01	39.05	86.81	67.23	79.06	70.80	62.81
BERT _{base}	Visual+Textual (VT)	53.89	50.54	49.15	38.83	87.04	66.81	79.16	70.77	62.02
BERT _{base}	Bi-directional VT	77.98	51.64	49.36	40.07	87.04	79.54	78.69	70.43	66.84
RoBERTa _{base}	Textual-Only	75.57	53.85	64.96	35.28	84.07	78.51	75.76	51.48	64.93
RoBERTa _{base}	Visual-Only	75.11	54.18	65.01	47.22	84.17	79.88	76.88	70.56	69.12
RoBERTa _{base}	Visual+Textual (VT)	74.20	53.98	65.43	47.35	83.94	79.96	76.87	70.73	69.05
RoBERTa _{base}	Bi-directional VT	75.34	54.79	65.43	48.94	84.63	81.41	79.38	70.76	70.08

Table 2.3: **Imagination Composition Ablation in Few-shot Setting.** *Bi-directional VT* represents the full input for iACE. More details about *Textual Only*, *Visual Only* and *Visual+Textual* are illustrated in Section 2.4.3.

use visual pairs $(\mathbf{i}_{sent1}, \mathbf{i}_{sent2})$ and one direction visual and textual pairs $(\mathbf{i}_{sent1}, \mathbf{t}_{sent2})$ as input respectively. Our full approach use *Bi-directional VT* which takes $(\mathbf{i}_{sent1}, \mathbf{t}_{sent2})$ and $(\mathbf{t}_{sent1}, \mathbf{i}_{sent2})$ as input.

Results are reported in Table 2.3 for Extreme Few-shot setting and normal few-shot setting. We observe *Bi-directional VT* data input achieve the most stable and the best average performance. Results show the importance of bi-directional imagination from all the textual input to construct an imagination-augmented cross-modal encoder.

2.4.4 Model-agnostic Improvement

iACE is a model-agnostic training paradigm that could help existing models achieve consistent gain over GLUE and SWAG with both the few-shot setting and full data setting. To validate such model-agnostic effectiveness of our proposed novel paradigm in processing natural language, we compare the performance with two language models (BERT and RoBERTa) of two architectures ("6L/512H" and "12L/768H"), and a strong visually supervised pre-trained baseline VOKEN [27].

Table 2.4 shows the metric comparison on GLUE and SWAG. The base models are trained with a masked language model. The VOKEN model is pre-trained with a masked

Base Model	Method	SST-2	QNLI	QQP	MNLI	MRPC	STS-B	SWAG	Avg.
BERT _{small}	VOKEN	89.7	85.0	87.3	78.6	78.2	80.4	57.6	79.5
BERT _{small}	iACE	89.8	86.2	87.7	78.9	78.4	82.7	57.9	80.2
BERT _{base}	VOKEN	92.2	88.6	88.6	82.6	83.5	86.0	70.6	84.6
BERT _{base}	iACE	91.7	88.6	89.1	82.8	85.8	86.6	70.8	85.1
RoBERTa _{small}	VOKEN	87.8	85.1	85.3	76.5	78.5	78.6	53.6	77.9
RoBERTa _{small}	iACE	89.2	85.1	86.5	76.8	79.0	78.7	53.7	78.3
RoBERTa _{base}	VOKEN	90.5	89.2	87.8	81.0	87.0	86.9	68.5	84.4
RoBERTa _{base}	iACE	91.6	89.1	87.9	82.6	87.7	86.9	68.5	84.9

Table 2.4: **Model-agnostic Improvement in Full Data Setting.** Results of iACE and VOKEN upon BERT and RoBERTa of small(6L/512H) and base(12L/768H) architecture are reported. The models are fine-tuned over GLUE Benchmark and SWAG with access to the full dataset. **BEST** results are highlighted.

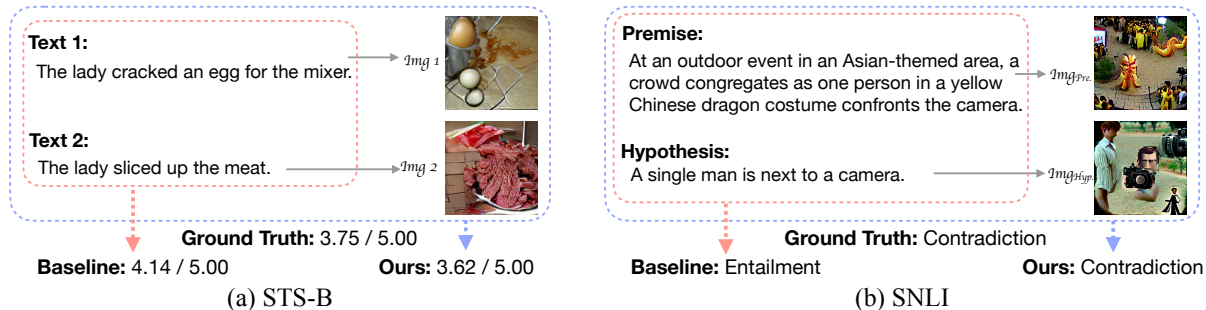


Figure 2.4: Case studies on the STS-B and SNLI tasks. The baseline models yield predictions solely based on the text input, while our approach takes both the text input and corresponding visualization into consideration. On both tasks, our iACE gives predictions that are more aligned with the ground truth.

language model with an additional voken-classification task as introduced visual supervision. iACE achieves model-agnostic improvement over the model that solely fine-tune based on textual information, including the pure-language-based model and visually supervised pre-trained model. The gain is consistently observed from different architectures of models.

2.4.5 Case Study

Figure 2.4 lists out our examples for the case study. We show the results from the natural language inference and sentence similarity task. We use examples from the STS-B

and SNLI datasets. Our contextual imagination describes the textual input as expected and provides an external prediction reference.

For example (a), given the structurally diversified sentence and low n -grams overlaps but high semantic similarity, we observe that the pure language-based model predicts the wrong label. While the imagination helps the model capture the semantic similarity between two textual inputs via comparing the cross-modal semantics with the imagination information. From example (b), we observe the pure language-based model predicts the wrong label based on the similar sentence structure and high n -grams overlaps. While the imagination helps the model capture the difference between the similar premise and hypothesis text.

Chapter 3

Imagination-Guided Open-Ended Text Generation

3.1 Introduction

One great resource human writers cherish is the ability of imagination, with which they render mental images about an actual or vicarious experience and link knowledge that would later make the writing more concrete, sensible, and intriguing. Cognitive studies show that visual imagery improves comprehension during language processing [3, 68, 69], and that mental imagery facilitates humans' written language expression at young ages [70].

When it comes to the study of Artificial Intelligence (AI), one classic challenge for AI systems is to generate informative and coherent text snippets. Open-ended text generation is such a task that provides an input context, and asks the model to generate a piece of text that is consistent with the context. This is the cornerstone of a wide range of downstream tasks such as text completion [9, 71], story generation [72, 73, 74, 75], and dialogue systems [76, 77, 78, 79, 80], and has received much attention throughout the

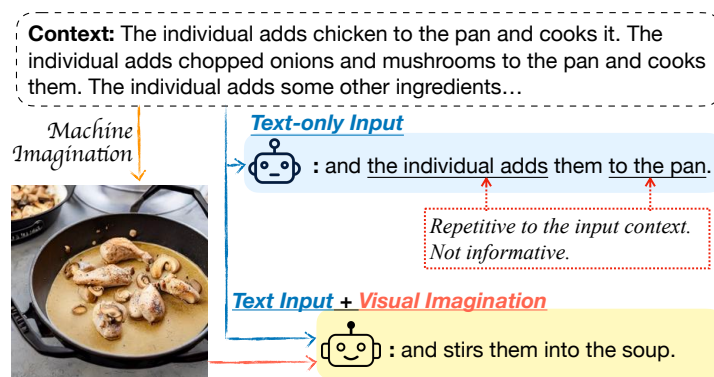


Figure 3.1: When performing open-ended text generation, the language models prompted with text-only input may generate repetitive or unilluminating contents, which is also known as degeneration. Hereby, we propose to use machine-generated images as additional visual supervision to guide the language models in generating more informative and coherent text with the given context.

years. Inspired by human writers’ common practice of creative visualization, we ask the following question: Can we endow machines with the same ability to construct a general picture of the context and use it as a blueprint to guide text generation?

Recent advances in text-to-image generation make it possible to visualize machine imaginations for a given context [4, 6, 60, 81, 82]. Moreover, this line of work shows great potential in utilizing textual information to guide image synthesis. It comes naturally that one may attempt to complete the loop by using visual supervision to guide text generation.

In this work, we propose using machine-generated images to guide the language model (LM) in open-ended text generation. More specifically, we visualize machine imagination for the input context by rendering images with StableDiffusion [4], a state-of-the-art text-to-image generator. The machine imagination acts as additional visual supervision to guide LMs in generating informative and coherent text in two ways. Firstly, the machine-generated images are introduced as the input to the LM in the form of the visual prefix. Secondly, we designed a contrastive training objective that enforces the generated text to be semantically similar to the visual supervision.

We conduct experiments on three open-ended text generation tasks, namely text completion, story generation, and concept-to-text generation. Extensive experiments in the few-shot settings show better or competitive performance to state-of-the-art baselines on both automatic metrics and human evaluation. Experiments with full-data settings show that introducing machine-generated visual supervision with our iNLG yields consistent improvements on various LM models including GPT-2 [9], BART [83], and T5 [84].

Our main contributions are as follows:

- We introduce a novel paradigm that leverages machine-generated images to guide open-ended text generation. This endows the machines with the ability of creative visualization that human writers often demonstrate.
- We distill the vision information from the pre-trained multimodal models and further construct visual prefixes to guide language models performing text generation with teacher forcing and contrastive objectives.
- Extensive experiments show the effectiveness of iNLG as a model-agnostic framework in open-ended text generation tasks, including text completion, story generation, and concept-to-text in both few-shot and full-data settings.

3.2 Related Work

Open-ended Conditional Text Generation is the task of generating a coherent portion of the text based on the given context. Recent advances in pre-trained models have pushed frontier in the open-ended conditional text generation, such as text completion [85, 86], story generation [72, 87, 88] and concept-to-text generation [89, 90]. Despite the success of large language models, text degeneration and semantic coverage still remain as two core technical challenges in few-shot open-ended text generation. To

improve the text coverage, StoryEndGen [71] leverages the knowledge graph to encode context sequentially. Fan et al. [72] and Yao et al. [88] plan the content (premise or keywords) first and then encourage the generation based on planned content. To mitigate the text degeneration, SimCTG [74] uses a contrastive training strategy to encourage the model to learn isotropic token embeddings. Similar to our approach, Wang et al. [91] generates a scene graph for each concept and combines them with text for the model input. Previous work has proposed to add visual information to LM by retrieving images from the Internet or large-scale image sets [92, 93, 94]. However, the retrieved images may fail to fully incorporate the context, which will misguide the LM from yielding contextually consistent predictions. Unlike prior work, our approach leverages images generated conditioning on the context to assist the text generation process.

Visually-aided NLP Recent work show the power of visual guidance in natural language processing, spanning from the language representation learning [11, 39, 43, 44, 45, 46, 47, 27], the downstream tasks [11, 28, 29, 32, 33, 34] and evaluation [13]. They either leverage visual information from an external vision-and-language corpus or obtain such visual knowledge from the large pre-trained model. In this line of work, imagination achieves promising performance in various NLP domains [11, 13, 48, 91]. Previous imagination-based work in NLP either study non-generation problems [11, 13] or utilize non-visual information [48, 91]. Our work explores the potential of generating visual imagination to improve open-ended text generation tasks.

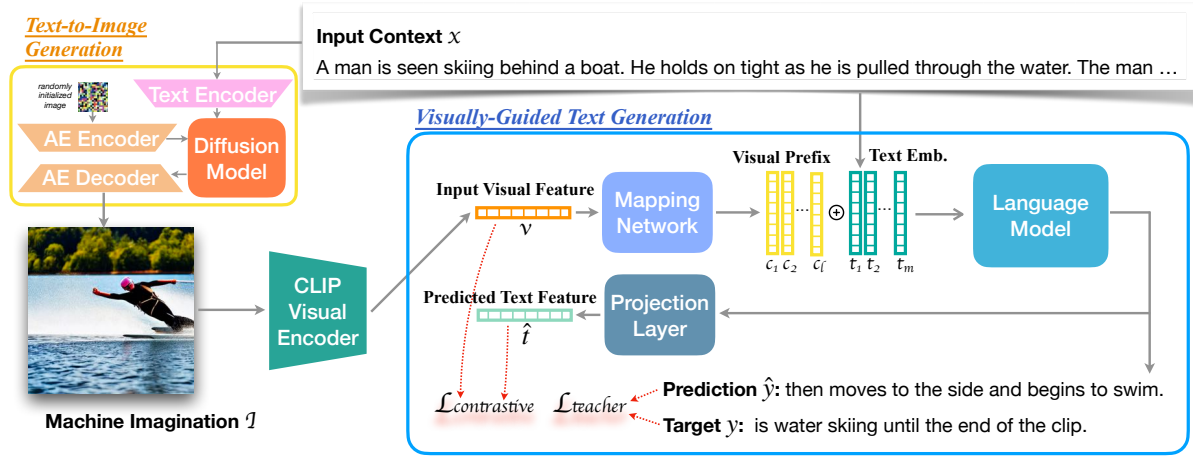


Figure 3.2: An overview of our iNLG. Given an input context x , we first visualize the context with the text-to-image generation model. Then we use the machine-generated image I as the additional visual supervision to guide the language model in open-ended text generation. The visual feature is provided as a source of input to the LM in the form of the visual prefix. Aside from the teacher forcing objective $\mathcal{L}_{teacher}$, we also enforce the LM to generate text that is semantically similar to the machine imagination with a contrastive training objective $\mathcal{L}_{contrastive}$.

3.3 Method

3.3.1 Overview

Open-ended text generation is a task that provides an input context, and asks the model to generate a piece of text that is consistent with the context.

This work mainly focused on introducing machine-rendered images to assist LM in performing open-ended text generation. More specifically, given the context x^i , we first use a text-to-image generator to illustrate an image I^i that depicts the input context. The LM is prompted with image I^i as the visual prefix along with the text context x^i , and will incorporate the multimodal input to generate the output text \hat{y}^i .

Figure 3.2 provides an overview of our iNLG framework, which mainly involves two modules. The first module is a text-to-image generator that takes in the input context and illustrates a descriptive image, which we also refer to as the machine imagination. The

second module is a visually-guided language model that utilizes the machine imagination as a source of input and also a supervision that encourages the LM to generate text that is semantically similar to the visual information.

3.3.2 Text-to-Image Rendering

In this work, we propose to use images generated conditioning on the context by the machines as additional visual information to the LM. The text-to-image generation backbone is StableDiffusion [4], which mainly consists of a text encoder, a diffusion model, and an autoencoder. The text encoder is from the frozen CLIP ViT-L/14 [23] and encodes the input text to textual embeddings. The diffusion model uses UNet [95] to provide noise estimation. The UNet is modified so as to attend to the input textual embeddings. The encoder of the pretrained autoencoder encodes images into the lower-resolution latent maps \mathbf{z}_T . At each step t , the diffusion model provides the noise estimation ϵ and modifies \mathbf{z}_t correspondingly. The decoder of the pretrained autoencoder takes the final noise-free latent map \mathbf{z} and generates the image prediction. StableDiffusion is trained with LAION-5B [96].

3.3.3 Visually Guided Text Generation

Visual Prefix Construction One can encode the visual information with the pre-trained visual models. However, such visual embedding may lie in a representation space different from the LM due to the discrepancy between models. One way of introducing features extracted by another network to the current model is through feature mapping [97]. With a dataset of image-text pairs $(\mathbf{I}', \mathbf{x}')$, we can pre-train a mapping network \mathcal{F} for a given LM in an image captioning formulation. More specifically, we encode \mathbf{I}' with the visual encoder $\text{Enc}_{\text{visual}}$ and receive its visual features \mathbf{v}' . Then we

apply the mapping network \mathcal{F} over \mathbf{v}' , and receive a sequence of l visual prefixes:

$$c'_1, c'_2, \dots, c'_l = \mathcal{F}(\mathbf{v}') = \mathcal{F}(\text{Enc}_{\text{visual}}(\mathbf{I}')) \quad (3.1)$$

We provide the list of visual prefix as input to the LM with the corresponding text \mathbf{x}' as the target output. Such a pre-training process enables \mathcal{F} to project visual features into the visual prefix that lies within the same embedding distributions as the LM. The mapping network is agnostic of the downstream task, and only depends on the visual source and the LM.

After generating a descriptive image \mathbf{I}^i for the input context \mathbf{x}^i , we use CLIP to encode \mathbf{I}^i and receive its visual features \mathbf{v}^i . We apply the pre-trained mapping network \mathcal{F} over \mathbf{v}^i , and receive the visual prefix \mathbf{c}^i of length l :

$$\mathbf{c}^i = \{c_1^i, c_2^i, \dots, c_l^i\} = \mathcal{F}(\text{CLIP}(\mathbf{I}^i)) \quad (3.2)$$

Visually-guided Language Modeling We use the visual information to guide text generation in two ways, reflected in the following two training objectives. Firstly, we directly introduce the machine-generated visual information as input to the LM. We concatenate the visual prefix \mathbf{c}^i and the text embeddings \mathbf{t}^i for the input context \mathbf{x}^i with m tokens. LM input can be denoted as $[\mathbf{c}^i; \mathbf{t}^i] = \{c_1^i, \dots, c_l^i, t_1^i, \dots, t_m^i\}$. With $\mathbf{y}^i = \{y_1^i, y_2^i, \dots, y_n^i\}$ denoting the target output of n tokens, and θ denoting the trainable parameters, we can list out the teacher forcing training objective as follows:

$$\mathcal{L}_{\text{teacher}} = - \sum_{j=1}^n \log p_{\theta}(y_j^i | \mathbf{c}^i; \mathbf{t}^i; \mathbf{y}_{<j}^i) \quad (3.3)$$

In addition, we design a contrastive objective to enforce the generated text to be

semantically similar to the input visual supervision with the InfoNCE loss [98, 99]:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\text{sim}(\mathbf{v}^i, \hat{\mathbf{t}}^i)/\tau)}{\sum_{j \neq i} \exp(\text{sim}(\mathbf{v}^i, \hat{\mathbf{t}}^j)/\tau)} \quad (3.4)$$

in which $\hat{\mathbf{t}}$ is the projected representation of the decoder’s last layer’s output, and can be viewed as the sentence-level representation of the generated text. Here $\text{sim}(\cdot, \cdot)$ first normalizes the two vectors, then compute their cosine similarity, and τ is the temperature.

3.3.4 Training & Inference

We first pre-train the mapping network on the pre-training dataset with the teacher-forcing objective. Such pre-training is agnostic of the downstream task, and only depends on the type of base LM.

When applying our iNLG on downstream tasks, we train the base LM with the teacher forcing objective for the first $N_{\text{no_contra}}$ epochs. Then, we introduce the contrastive objective and tune the base LM together with the mapping network and projection layer by minimizing the following loss \mathcal{L} . Here ep denotes the epoch and λ is the factor:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{teacher}}, & ep < N_{\text{no_contra}}, \\ \mathcal{L}_{\text{teacher}} + \lambda \mathcal{L}_{\text{contrastive}}, & ep > N_{\text{no_contra}}, \end{cases} \quad (3.5)$$

During inference, we provide the context and machine-generated image to the LM. We use beam search during decoding with a beam width of 10.

3.4 Experimental Setup

3.4.1 Tasks, Datasets, and Baselines

We apply our iNLG on three open-ended text generation setups: sentence completion, story generation, and concept-to-text generation. Table 3.1 shows examples for each task.

Task	Input Context	Target Output
Text Completion	Different people are interviewed on camera while several others are shown raking up the leaves. A man is seen sitting in his car and another puts his gloves on. The camera	pans over the raked up leaves while several others discuss their hard work.
Story Generation	Live Show. Tim was in his school’s play.	He was nervous about their first show. He almost dropped out. The show went smoothly. Tim was excited for his second show.
Concept-to-Text	grow, flower, pavement	Wild flower growing through crack in the tiled pavement.

Table 3.1: Exemplars of the input context and corresponding target output for three open-ended text generation task covered in this study, namely story generation, text completion, and concept-to-text generation.

Sentence Completion is a task of finishing the sentence in a commonsense inference scenario. We conduct experiments on the ActivityNet [100] subset¹ of HellaSwag [101], which is a benchmark for commonsense natural language inference that ask the model to predict the most likely follow-up among several choices given a specific context. We compare with StoryEndGen [71] which encodes the given context incrementally and attends to the one-hop knowledge graph retrieved from ConceptNet for the context tokens. We implement our iNLG on top of the GPT-2 [9], which by nature, can generate the follow-up for an arbitrary input in a zero-shot manner.

Story Generation requires the model to compose a story based on the given title or context. We conduct experiments on the widely used story generation benchmark ROCStories [102]. Each data item consists of a story title and a human-written five-

¹14740/982/2261 samples for train/validation/test.

sentence everyday life story that incorporates commonsense related to the title.² We provide the story title and the story’s first sentence as the input context, and ask the LM to predict the following four sentences. We consider the following methods as baselines: Action-Plan [72] first predicts the premise of a story with the convolutional LM [103], then use the fusion mechanism [104] to encourage a convolutional seq2seq model [105] to generate the story from the premise. Plan-and-Write [88] first plans a storyline that consists of keywords, then generate the story conditioned on the storyline. Its model structure is built upon GRU [106]. SimCTG [74] proposes a contrastive training objective that encourages the LM to learn discriminative and isotropic token representations, and is implemented on GPT-2 [9].

Concept-to-Text is a relatively more constrained conditional text generation task involving commonsense reasoning. This task provides a set of concepts as input, and requires the model to generate a piece of text that incorporates the concepts and describes an everyday scenario. We conduct experiments on the CommonGen [107] benchmark.³ We compare against the following models: KG-BART [89] encompasses the relations of concepts with the knowledge graph and augments the BART [83] encoder and decoder with graph representations. ModelAdapt [108] is built upon BART and removes the positional embedding in the encoder. Imagine-and-Verbalize (I&V) [91] predicts a scene graph for each set of concepts, and uses it as an additional input to the LM. In contrast to I&V, we directly visualize the concepts and use the machine-generated images as the auxiliary information to assist the concept-to-text generation.

²We use the split provided by Su et al. [93], which is based on the ROCStories Winter 2017 release and contains 49666/1500/1500 items for the train/validation/test sets.

³We use the in-house split provided by Wang et al. [91], which contains 65323/2066/4018 samples for train/validation/test.

3.4.2 Evaluation

Automatic For sentence completion and story generation, we follow previous work and evaluate the quality of the generated text from the aspect of model degeneration level (rep- n , diversity, distinct- n), text distribution divergence (MAUVE), and semantic similarity (BERTScore): (1) rep- $n = 1.0 - \frac{|\text{unique } n\text{-grams}|}{|\text{total } n\text{-grams}|}$ measures sequence level repetition by computing the portion of duplicate n -grams [109]. (2) diversity = $\prod_{n=2}^4 (1 - \text{rep-}n)$ measures the diversity of n -grams [93]. (3) distinct- $n = \frac{|\text{unique } n\text{-grams}|}{|\text{length of text}|}$ measures the portion of distinct n -grams in the text [110]. (4) MAUVE measures the learned distributions divergence between the generated text and human-written text [111],⁴ a low MAUVE indicates a great difference between the distributions of generated text and human text. (5) BERTScore assesses contextual text similarity between two pieces of texts by computing the cosine similarities between their tokens’ embeddings [112],⁵ a low BERTScore means the generated text is contextually different from the ground-truth.

For concept-to-text, following prior work, we report the metrics scores on BLEU [113], METEOR [114], CIDEr [115], SPICE [116], and BERTScore [112].

Human We also set up a human evaluation as a complementary evaluation beyond the automatic metrics. We select 100 samples from the test set for sentence completion and story generation and perform the head-to-head comparison between the text snippets generated by our iNLG and the baseline models. We invite human annotators to compare the text quality from the following three independent aspects: (1) *Coherence*: Which snippet is more semantically consistent with the context, and follows the logic of the context more naturally. (2) *Fluency*: Which snippet is more fluent in English. (3) *Informativeness*: Which snippet contains more interesting content, and describes

⁴We report MAUVE with gpt2-large as the base model.

⁵We report BERTScore with roberta-large as base model.

the scenes that are more likely to happen in real life. Three human judges rate each comparison.

3.4.3 Implementation Details

We use StableDiffusion-v1-1 [4] to render a 512x512 image from the context, and use CLIP ViT/B-32 to extract features offline. The mapping network is an 8-layer Transformer, and the visual prefix length is 20. For the sentence completion and story generation tasks, the mapping network is pre-trained on the MSCOCO [117] dataset. For the concept-to-text task, the mapping network is pre-trained on VIST [118].⁶ We pre-train the mapping network for 5 epochs with a batch size of 128. Results are reported on three repeat runs.

3.5 Result and Analysis

3.5.1 Few-Shot Learning Results

Open-ended text generation is a broad topic with flexible and inexhaustible setups, many of which have low resources. Collecting annotations is often extremely expensive and time-consuming. Therefore, we first report few-shot results to check if our iNLG can rapidly adapt to new task setups with a few examples, which is more practical in real-life.

More specifically, we report few-shot open-ended text generation results with 1% of the training data. For sentence completion and story generation tasks, the base LM is GPT2-base [9]. For concept-to-text, we test it with BART-base [83] as the base LM.

⁶CommonGen is built upon image and video captioning datasets including MSCOCO. To avoid data leakage, we choose to pre-train the mapping network on VIST, which is not revealed to CommonGen.

Task	*	Setting	rep-2 ↓	rep-3 ↓	rep-4 ↓	diversity ↑	distinct-2 ↑	MAUVE↑	BERTScore↑
Sentence Completion	0	Human	0.45	0.05	0.01	99.50	77.32	-	-
	1	GPT2 <i>no finetune</i> [9]	6.71	6.87	10.13	78.07	74.83	44.19	22.57
	2	StoryEndGen [71]	39.53	35.11	39.30	34.12	44.57	0.45	-47.29
	3	GPT2 <i>text-only finetune</i>	4.20	4.03	5.53	86.85	75.14	49.45	24.13
	4	GPT2 +iNLG	2.43	2.61	3.57	91.63	75.92	60.30	24.25
Story Generation	5	Human	1.76	0.38	0.15	97.71	56.34	-	-
	6	GPT2 <i>no finetune</i>	37.65	22.76	21.92	45.67	43.42	0.43	-7.77
	7	Action-Plan [72]	52.05	35.58	28.11	26.97	21.43	0.41	-18.32
	8	Plan-and-Write [88]	45.22	32.86	23.34	30.71	20.83	0.41	-37.35
	9	SimCTG [74]	28.72	24.02	20.61	43.00	42.06	0.43	18.01
	10	GPT2 <i>text-only finetune</i>	25.41	18.51	14.41	52.10	46.60	9.10	21.23
	11	GPT2 +iNLG	10.73	5.64	3.42	81.36	51.91	35.94	23.03

Table 3.2: Generation quality scores for few-shot text completion on the ActivityNet and few-shot story generation on ROCStories. “Human” shows the human performance and “GPT2 *no finetune*” denotes the vanilla GPT2 model without tuning. All the other listed models are trained with 1% of the training data. “+iNLG” denotes introducing machine-generated images on top of the base LM.

Sentence Completion As shown in Table 3.2, StoryEndGen (#2) suffers from degeneration with the highest rep- n and the lowest diversity. Training with only 1% of the training data improves GPT2’s performance on all metrics (#3 vs. #1). Under the same few-shot setting, adding additional machine-generated images with our iNLG (#4) further alleviate model degeneration. The improvement on MAUVE also indicates that introducing visual input can aid GPT2 in generating text that is more similar to the human-written ones.

Story Generation As shown in Table 3.2, for the story generation task that requires the LM to compose longer text, we see the vanilla GPT2 without tuning suffering from more severe degeneration compared to rendering a sentence ending (#6 vs. #1). The high rep- n scores indicate that the two non-Transformer-based baselines Action-Plan (#7) and Plan-and-Write (#8) stammer with repetitive tokens, which greatly differs from the human-written text (leads to low MAUVE) and does not have concrete meanings (leads to low BERTScore). The models based on GPT-2 (#9-#10) yield more complete sentences with concrete meanings (BERTScore gets higher). However, they keep repeating the

* Setting	B-4	M.	CIDEr	SPICE	BertS.
1 BART-base <i>text-only finetune</i>	20.72	25.47	114.49	24.58	59.76
2 +KG [89]	15.26	24.44	98.53	23.13	52.76
3 +Adapt [108]	23.11	25.96	123.44	25.14	61.53
4 +I&V [91]	24.50	25.89	119.61	25.59	57.29
5 +iNLG	25.07	26.48	127.93	26.32	63.37

Table 3.3: Automatic metrics scores for few-shot concept-to-text generation on CommonGen with 1% of the training data. All listed models are implemented on BART-base. “+KG” adds knowledge graph, “+Adapt” applies model adaption, “+I&V” adds scene graph, and “+iNLG” introduces machine-generated images as input. B-4: BLEU-4; M.: METEOR; BertS.: BERTScore.

same sentence, which is still quite different from human language (MAUVE remains low). Applying iNLG to GPT-2 leads to minor degeneration and has the best performance on all metrics (#11).

Concept-to-Text Table 3.3 shows that knowledge graph information may not be fully exploited under the few-shot setting (#2), while removing the information of relative positions between input concepts helps the LM write better sentences (#3). Introducing machine-generated images can improve the base LM’s performance on concept-to-text generation (#5 vs. #1). While both I&V and our iNLG involve machine “imagination”, we provide such information in different forms (scene graphs vs. images). Comparing #4 and #5, our iNLG outperforms I&V with BART-base as the base LM. This suggests that the additional information introduced by I&V and iNLG is complementary.

Human Evaluation Table 3.4 lists out human evaluation results on text completion and story generation. Our iNLG outperforms the compared baselines on all three criteria in the model-level head-to-head comparisons. This further verifies the effectiveness of our iNLG in generating fluent and informative text snippets that better align with the given context.

Task	Models	Coherence			Fluency			Informativeness		
		Win(%)	Tie(%)	Lose(%)	Win(%)	Tie(%)	Lose(%)	Win(%)	Tie(%)	Lose(%)
Sentence Completion	Ours vs. StoryEndGen	51.67	20.33	28.00	44.67	19.33	36.00	41.33	18.33	40.33
	Ours vs. GPT2 <i>no finetune</i>	51.00	22.67	26.33	45.00	22.33	32.67	41.00	21.00	38.00
	Ours vs. GPT2 <i>text-only finetune</i>	58.00	24.33	17.67	43.33	18.67	38.00	42.33	21.67	36.00
Story Generation	Ours vs. Action-Plan	51.00	24.67	24.33	54.67	16.33	29.00	52.00	15.00	33.00
	Ours vs. Plan-and-Write	45.33	25.67	29.00	53.00	16.67	30.33	54.67	17.00	28.33
	Ours vs. SimCTG	42.00	27.67	30.33	40.33	25.67	34.00	43.33	18.33	38.33
	Ours vs. GPT2 <i>no finetune</i>	43.33	24.33	32.33	43.67	20.33	36.00	44.67	19.00	36.33
	Ours vs. GPT2 <i>text-only finetune</i>	39.33	26.67	34.00	38.67	26.67	34.67	44.33	22.67	33.00

Table 3.4: Human evaluation results for the sentence completion task and the story generation task. The scores indicate the percentage of win, tie or lose when comparing our iNLG with the baseline models.

3.5.2 Model-Agnostic Improvement

We further report open-ended text generation results with various base LM when trained with the full set of data. For concept-to-text, we experiment with BART-base/large [83] and T5-base/large [84]. For sentence completion and story generation, we record results on GPT2-base/large [9]. As shown in Table 3.5, introducing machine-generated visual supervision with our iNLG leads to model-agnostic improvements over text-only finetuning. This holds true for all the listed base LM with different architectures and verifies that our iNLG is a model-agnostic framework.

3.5.3 Performance Analysis

Source of Image We first perform an ablation study to understand how the source of visual information affects our iNLG framework. We compare retrieved/generated images from four sources: (1) the first returned result by Yahoo Image Search;⁷ (2) images rendered by VQGAN+CLIP [60];⁸ (3) images rendered by OFA [82],⁹ and (4) images rendered by StableDiffusion [4], with which we report the main results.

As shown in Figure 3.3(a), the images generated by machines act as a more effective

⁷<https://images.search.yahoo.com/>

⁸<https://github.com/nerdyrodent/VQGAN-CLIP>

⁹<https://github.com/OFA-Sys/OFA>

Base LM	Setting	Metrics				
<i>Concept-to-Text</i>		B-4↑	MET.↑	CIDEr↑	SPICE↑	BertS.↑
BART-base	text-only	30.32	31.35	158.92	31.22	68.50
	+iNLG	30.60	31.44	160.63	31.42	69.02
BART-large	text-only	32.38	33.06	169.69	33.01	70.33
	+iNLG	32.76	33.17	171.47	33.35	70.79
T5-base	text-only	30.39	30.87	163.67	32.77	70.03
	+iNLG	31.09	31.18	165.52	32.81	70.35
T5-large	text-only	34.13	32.91	175.67	34.30	72.44
	+iNLG	34.50	33.87	177.65	35.48	72.70
<i>Sentence Completion</i>		rep-4↓	div.↑	dist-2↑	MAUVE↑	BertS.↑
GPT2-base	text-only	4.20	87.46	72.87	61.42	29.84
	+iNLG	3.95	89.33	74.09	64.01	30.10
GPT2-large	text-only	1.77	96.54	76.74	87.81	31.66
	+iNLG	2.05	95.90	76.80	89.11	32.15
<i>Story Generation</i>		rep-4↓	div.↑	dist-2↑	MAUVE↑	BertS.↑
GPT2-base	text-only	7.83	68.42	49.53	33.13	28.81
	+iNLG	6.80	71.17	49.92	38.86	29.13
GPT2-large	text-only	1.02	91.91	54.17	82.81	31.86
	+iNLG	0.85	92.51	54.54	87.83	32.03

Table 3.5: Automatic metric scores when trained with the full set of data with ablations of the base LM. Introducing our iNLG leads to model-agnostic improvements across the board. B-4: BLEU-4; MET.: METEOR; BertS.: BERTScore; div.: diversity; dist-2: distinct-2.

supervision than the retrieved images. This validates our motivation of introducing machine-generated images over retrieved ones to guide LM in performing text generation. Among the three text-to-image generators, VQGAN+CLIP is slightly inferior to the other two, while StableDiffusion and OFA have mixed performance. Images generated by StableDiffusion rank first on CommonGen, while images rendered with OFA score slightly higher on ActivityNet. Figure 3.3(b) reports the average image rendering time, where StableDiffusion is $10\times$ faster when rendering images than the other two.

Contrastive Training We examine the effect of the contrastive training objective on CommonGen, and the results are presented in Figure 3.4. We notice that introducing

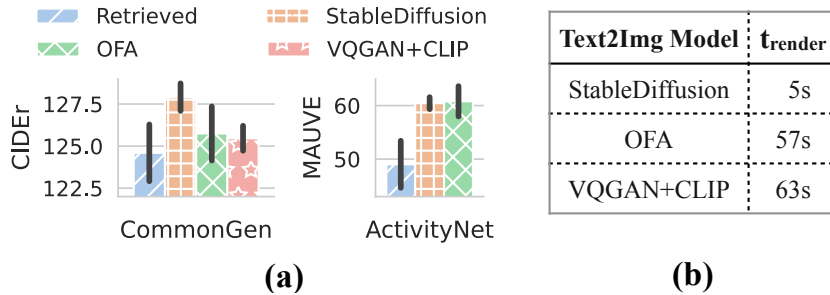


Figure 3.3: (a) iNLG’s performance on CommonGen and ActivityNet with visual supervisions retrieved from the web or generated by machines. Scores are reported with error bars. (b) Average time to render an image on TITAN RTX with each text-to-image generator.

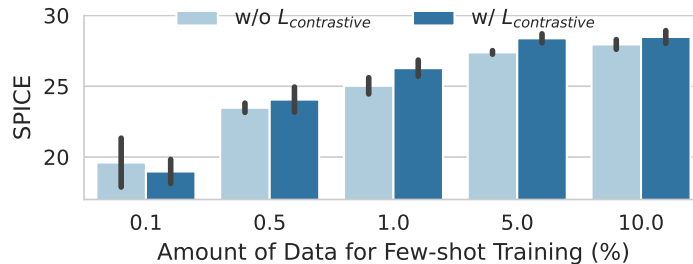


Figure 3.4: Performance of applying our iNLG on BART-base for few-shot concept-to-text with ablated training objective $\mathcal{L}_{\text{contrastive}}$ on various few-shot settings. Scores are reported with error bars.

$\mathcal{L}_{\text{contrastive}}$ improves iNLG’s performance on 4 out of 5 listed few-shot setups, which suggests that our contrastive training objective generally can assist the LM in composing open-ended text snippets. One exception is in the extreme few-shot setting with only 0.1% of training data, where the amount of data is insufficient to let the LM form a decent representation. In this case, enforcing the sentence representation to be similar to the visual supervision with $\mathcal{L}_{\text{contrastive}}$ might misguide the LM.

Mapping Network & Visual Prefix We discuss the effects of different types of mapping networks and various visual prefix lengths. Aside from the 8-layer Transformer we used in the main experiments, we also tried a simple Multi-Layer Perceptron (MLP) with two fully-connected layers. As shown in Figure 3.5, the Transformer-based mapping

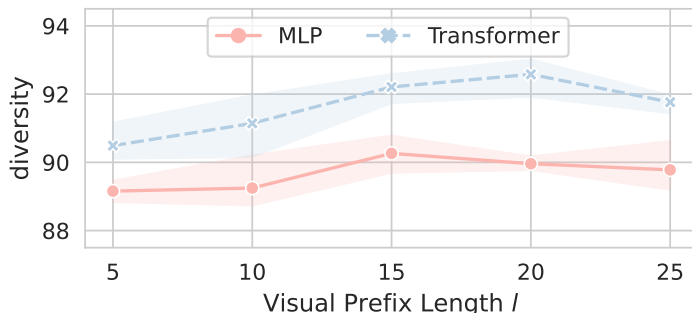


Figure 3.5: Performance of our iNLG on few-shot sentence completion with various visual prefix lengths and with MLP and Transformer as mapping network. Scores are reported with error bands.

Tune LM	Pretrain Map.	Tune Map.	diversity \uparrow	MAUVE \uparrow
\times	\times	\times	15.52	0.47
\times	\times	\checkmark	78.20	33.79
\times	\checkmark	\times	27.06	1.83
\times	\checkmark	\checkmark	76.36	25.15
\checkmark	\times	\times	87.45	48.06
\checkmark	\times	\checkmark	88.68	51.81
\checkmark	\checkmark	\times	89.05	55.61
\checkmark	\checkmark	\checkmark	92.68	60.62

Table 3.6: Performance of our iNLG on few-shot sentence completion with ablated settings on whether to tune the LM, pretrain the mapping network (Pretrain Map.) and tune the mapping network (Tune Map.).

network outperforms MLP on all listed l . MLP has the best performance when visual prefix length $l = 15$, while the Transformer-based mapping network scores highest when $l = 20$.

Model Weight Tuning Table 3.6 compares the influence of pre-training/tuning the weights of different modules of our iNLG. Generally speaking, tuning the weights during training outperforms freezing the weights, which applies to both the base LM and the mapping network. In addition, considering our few-show setup, pre-training the mapping network also helps our iNLG gain better performances. The best combination is applying the pre-trained mapping network, and tuning it together with the base LM on the few-

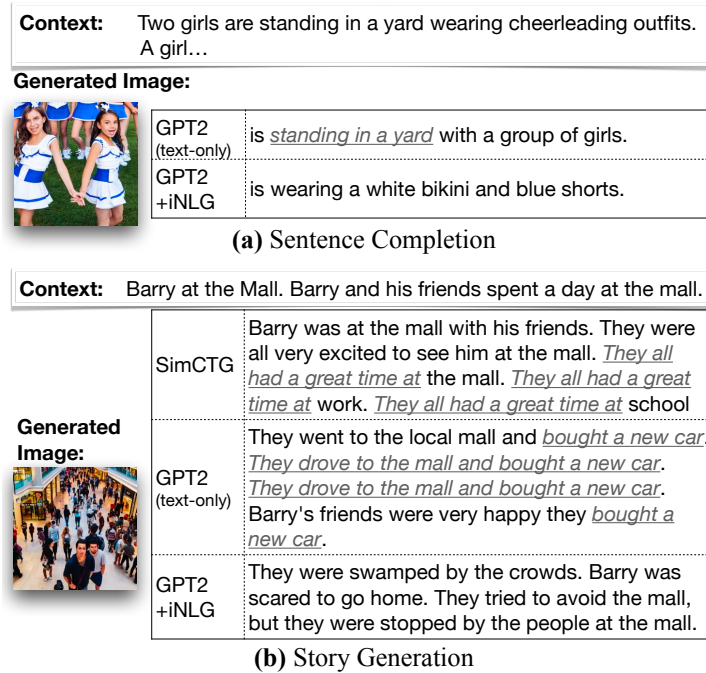


Figure 3.6: Sentence ending and stories generated by GPT2-based methods tuned with 1% of the training data. *Repetitive contents* are underlined. The sentence ending and story written by our iNLG is coherent with the context, related to the machine-generated image, and has minor degeneration.

shot downstream task.

Showcase Figure 3.6 provides two showcases on few-shot sentence completion and story generation to compare our iNLG with the GPT2-based baselines. SimCTG and GPT2 tuned with text-only corpus rendering repeated segments, either copying from the input context, or simply repeating themselves. In comparison, our iNLG has minor degeneration and writes coherent sentence endings or stories with more creative details in both tasks.

Chapter 4

An Imagination-Based Automatic Evaluation Metric for Natural Language Generation

4.1 Introduction

A major challenge for natural language generation (NLG) is to design an automatic evaluation metric that can align well with human judgments. To this end, many approaches have been investigated. Metrics that base on matching mechanisms such as BLEU [113], METEOR [114], CIDEr [115], have been widely adopted in the field. Edit-distance based metrics, such as CharacTER [119], WMD [120], SMD [121], have also been explored. Recently, BERTScore [112] and BLEURT [122] attempt to leverage BERT [24] to compare text embedding similarities, which correlates better with human judgments than previous methods. These automatic evaluation metrics make use of textual information from various angles extensively.

But what happens in our minds when we read, comprehend, and evaluate text? Re-

search [1, 123] has found that, unlike commonly designed automatic evaluation methods that compare the generated candidates with the references on the text domain only, humans, in contrast, leverage visual imagination and trigger neural activation in vision-related brain areas when reading text. Cognitive studies show that visual imagery improves comprehension during language processing [3, 68, 69]. Inspired by this imagination-based multi-modal mechanism in human text comprehension, we ask a critical research question: *can machines create a visual picture of any underlying sentence, and use their imaginations to improve natural language understanding?* The advances of recent pre-trained vision-language models such as CLIP [23] provide an excellent opportunity for us to utilize the learned image-text representations. This enables us to explore the possibility of incorporating multi-modal information into NLG evaluation.

In this work, we propose IMAGINE, an imagination-based automatic evaluation metric for text generation. Specifically, we first use the state-of-the-art text-to-image generator StableDiffusion [4] to visualize machine imagination from sentences, which is to generate descriptive images for the candidate text and the references. Then we receive the IMAGINE scores by computing two sets of similarity scores with the pre-trained CLIP model [23]: the visual similarity of the generated images, and the cross-modal similarity between the text and the generated image. Figure 4.1 shows an example.

To understand the role the machine-generated images play in NLG evaluation, we conduct a series of experiments with IMAGINE on multiple NLG tasks and datasets, including machine translation, text summarization, and sentence completion for open-ended text generation, aiming to answer the following questions:

1. *How influential is IMAGINE in NLG evaluation in terms of correlations with human judgments? Can it provide additional reference information on top of existing metrics?*

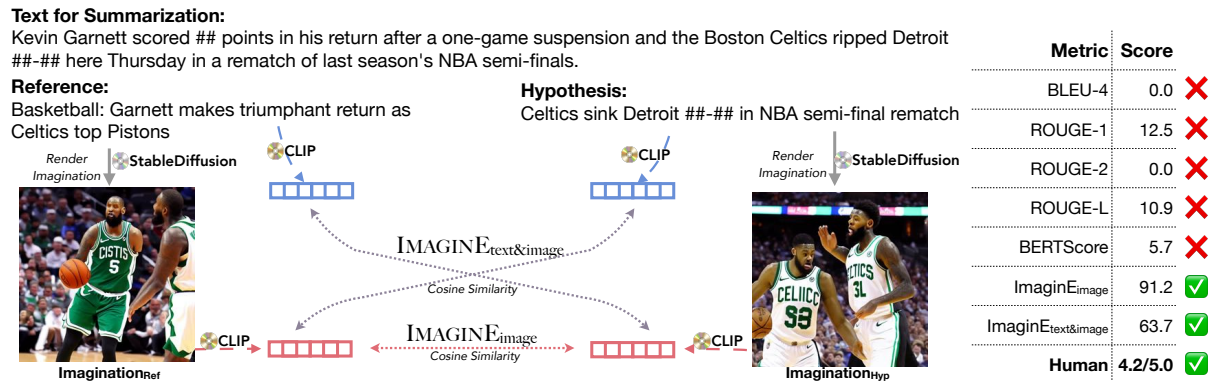


Figure 4.1: An evaluation example on GigaWord for text summarization. IMAGINE visualizes machine imagination with StableDiffusion [4] and extracts textual and visual representations with CLIP [23]. While traditional evaluation metrics for natural language generation rely on n -grams matching or textual embeddings comparison, IMAGINE incorporates machine-generated images into the evaluation process and enhances the understanding of the text snippet as a whole through the integration of multi-modal information.

2. *What are the applicable scenarios of introducing IMAGINE to NLG evaluation? When and why do machine-generated images help?*
3. *What are the potentials and limitations of introducing machine-generated images with IMAGINE to NLG evaluation?*

Experimental results show that IMAGINE can serve as a complementary evaluation metric to text-based ones, and adding IMAGINE scores to existing metrics surprisingly improves most of the popular metrics' correlations with human performance on various text generation tasks. This holds for both reference-based evaluation and reference-free evaluation. We further conduct comprehensive quantitative analyses with case studies to verify its effectiveness. Overall, IMAGINE displays great potential in introducing multi-modal information into NLG evaluation.

4.2 Related Work

Automatic Metrics for Natural Language Generation Common practices for NLG evaluation compare the generated hypothesis text with the annotated references. Metric performance is conventionally evaluated by its correlation with human judgments. Existing automatic evaluation metric calculations are mainly based on three mechanisms: n -grams overlap, edit distance, and embedding matching. BLEU [113], ROUGE- n [124], METEOR [114] and CIDEr [115] are a few widely used n -gram based metrics for text generation tasks. Another direction is based on edit distance [119, 125, 126, 127, 128], where they calculate the edit distance between the two text snippets with different optimizations. Embedding-based metrics [121, 120, 129, 130, 131] evaluate text quality using word and sentence embeddings, and more recently, with the help of BERT [112, 122].

Multi-Modal Automatic Metrics Aside from previous text-only metrics, some metrics utilize pre-trained multi-modal models and introduce visual features on top of text references for NLG evaluation. TIGEr [132] computes the text-image grounding scores with pre-trained SCAN [133]. ViLBERTScore-F [134] relies on pre-trained ViLBERT [45] to extract image-conditioned embeddings for the text. The CLIPScore [135] proposes a metric for image captioning by directly comparing images with captions using CLIP [23]. Our method differs in that we use visual picture generation as embodied imagination and apply our metric to various text-to-text generation tasks.

Mental Imagery The debate between pictorialists and propositionalists about how imagery information is stored in the human brain is still an open question in the neuroscience and psychology community [136]. We follow the views from pictorialists that information can be stored in a depictive and pictorial format in addition to language-like forms [137, 138]. In pictorialists’ model, mental imagery is constructed in the “visual

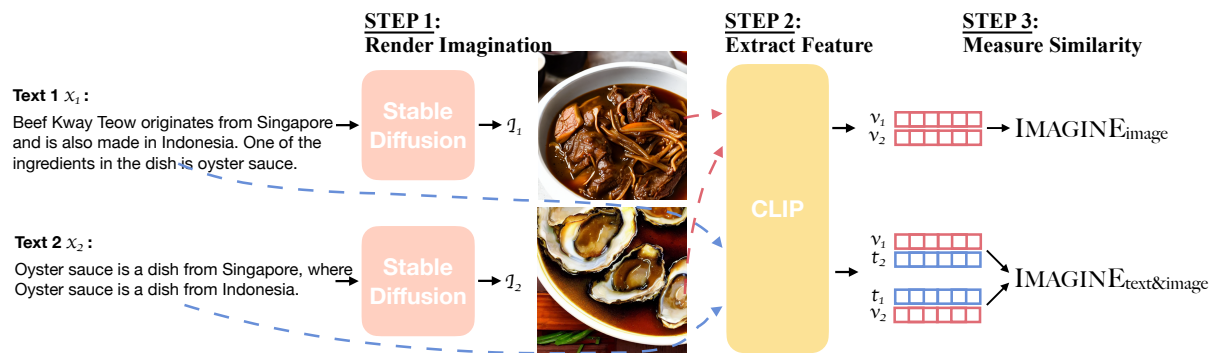


Figure 4.2: Illustration of the computation process of the IMAGINE metric. Given the two pieces of text for comparison, x_1 and x_2 , we render the machine imagination by generating two images I_1 and I_2 with the pre-trained StableDiffusion [4]. We extract features of the input text and corresponding generated images with CLIP [23]. We receive two variants of IMAGINE by computing the cosine similarity of the extracted features, in which $IMAGINE_{image}$ measures mono-modal similarities on the visual side, while $IMAGINE_{text\&image}$ conducts cross-modal matching.

buffer” either from the retinal image in seeing or from a long-term memory store of “deep representations” in the brain. Our image generation method is to mimic the generation of deep representations in machines, with the help of recent powerful text-to-image models. Inspired by empirical studies from cognitive science that visual imagination improves human text comprehension [1, 3, 68, 69, 139], we are interested in exploring if one can draw similar conclusions from automatic text evaluations by machines.

4.3 ImaginE

This section describes how our IMAGINE metric evaluates the similarity between two pieces of text with the help of machine imagination. Figure 4.2 provides an overview of our method.

4.3.1 Model Details

CLIP We use the cross-modal retrieval model, CLIP [23], for our evaluation purposes. CLIP jointly trains an image encoder and a text encoder to predict the correct pairing of image-text pairs with InfoNCE [98] on 400M image-text pairs gathered from the web. We utilize the CLIP-ViT-B/32 variant, which consists of a 12-layer, 8-head Transformer text encoder with a hidden size of 512, and a Vision Transformer (ViT) [61, 140] image encoder adopting the BERT-base configuration and using a 32×32 input patch size. Both the text and image representations are normalized and projected into the multi-modal space before computing pairing likelihood through cosine similarity.

StableDiffusion We perform text-to-image generation with StableDiffusion [4], which is a denoising diffusion probabilistic model [141]. The model comprises three key components: a text encoder, a diffusion model, and an autoencoder. The text encoder, adopted from the frozen CLIP-ViT-L/14 [23], is utilized to encode the input text into textual embeddings. The diffusion model, which leverages UNet [95] for noise estimation, is modified to attend to the input textual embeddings. We conduct experiments with StableDiffusion-v1-1, which was trained with LAION [96], using 256×256 images for pre-training, followed by 512×512 images for fine-tuning.

4.3.2 Imagine Similarity Score

In our proposed approach, as depicted in Figure 4.2, the computation of IMAGINE consists of three sequential steps. Firstly, the StableDiffusion model [4] is utilized to generate descriptive images, referred to as machine imagination, from the two text snippets being compared. Secondly, both the text snippets and the generated images are encoded using the CLIP model [23]. Finally, IMAGINE is calculated by computing the

cosine similarities of the resulting text and visual features, both in a mono-modal and cross-modal manner.

Step 1: Render Imagination For each image, StableDiffusion randomly initializes a latent matrix \mathbf{H} from the standard normal distribution and uses the encoder of the pre-trained autoencoder to encode \mathbf{H} into the lower-resolution latent map \mathbf{z}_T (T is the total inference steps). At each step t , the diffusion model estimates the noise, ϵ , and subtracts it from \mathbf{z}_t . The decoder of the pretrained autoencoder takes the final noise-free latent map \mathbf{z} and generates the image prediction \mathbf{I} of size 512×512 .

Step 2: Extract Feature In the previous step, we generate the corresponding images \mathbf{I}_1 and \mathbf{I}_2 for the pair of text \mathbf{x}_1 and \mathbf{x}_2 for comparison with the text-to-image synthesis backbone. Then we pass the machine-generated images \mathbf{I}_1 and \mathbf{I}_2 and the input text \mathbf{x}_1 and \mathbf{x}_2 through corresponding CLIP encoders to receive the visual representations \mathbf{v}_1 , \mathbf{v}_2 , and the textual representation \mathbf{t}_1 , \mathbf{t}_2 .

Step 3: Measure Similarity With $\text{sim}(\cdot, \cdot)$ denoting the process of first normalizing the two vectors, then computing their cosine similarity, we compute two types of similarity scores for IMAGINE with the extracted textual and visual features:

- (1) IMAGINE_{image} computes the visual representation similarity between \mathbf{v}_1 and \mathbf{v}_2 :

$$\text{IMAGINE}_{image} = \mathcal{F}(\text{sim}(\mathbf{v}_1, \mathbf{v}_2)) \quad (4.1)$$

- (2) $\text{IMAGINE}_{text\&image}$ ($\text{IMAGINE}_{t\&i}$) takes both the text and the generated image into consideration, and conducts cross-modal comparisons between $(\mathbf{t}_1, \mathbf{v}_2)$, as well as $(\mathbf{t}_2, \mathbf{v}_1)$:

$$\text{IMAGINE}_{t\&i} = \mathcal{F}\left(\frac{\text{sim}(\mathbf{t}_1, \mathbf{v}_2) + \text{sim}(\mathbf{t}_2, \mathbf{v}_1)}{2}\right) \quad (4.2)$$

The cosine similarity between the text and image representations theoretically has a range of $[-1, 1]$. However, in practice, the IMAGINE similarity scores tend to cluster within a more narrow interval $[l, h]$. Following Hessel et al. [135], we use a linear function \mathcal{F} to stretch the similarity score distribution to the range of $[0, 1]$, which is also the score range for most of the automatic metrics covered in this study. Eq. (4.3) shows how we re-scale the similarity score s into s' .

$$s' = \frac{s - l}{h - l},$$

$$[l, h] = \begin{cases} [0.1, 1.0], & \text{for IMAGINE}_{image}, \\ [0.1, 0.4], & \text{for IMAGINE}_{text\&image}. \end{cases} \quad (4.3)$$

4.3.3 Integration with Existing Metrics

The IMAGINE similarity scores can serve as standalone automatic metrics. Additionally, IMAGINE can be incorporated as an extension to existing metrics, as it offers multimodal references and addresses the limitations of current text-only evaluations that only compare tokens or text embeddings. This mimics the human process of comprehending text, where both text and visual imagination are utilized. The integration of IMAGINE with other automatic metrics is straightforward, achieved by summing the IMAGINE similarity score with the other automatic metric’s score for each example:

$$metric_score' += \text{IMAGINE}_{similarity_score} \quad (4.4)$$

4.4 Experimental Setup

4.4.1 Tasks, Datasets, and Models

We evaluate our approach on three popular natural language generation tasks: machine translation, abstractive text summarization, and open-ended text generation.

Machine Translation We use Fairseq [142] to generate English translation from German on IWSLT’14 [143] and WMT’19 [144] datasets.

Abstractive Text Summarization We use the implementation of Li et al. [145] to generate summarization on DUC2004¹ and use ProphetNet [146] for generation on Gigaword.² Both datasets are built upon news articles.

Open-ended Text Generation We perform experiments on the ActivityNet [100] subset of HellaSwag [101], which is a benchmark for commonsense natural language inference that ask the model to predict the most likely follow-up among several choices given a specific context. The dataset is derived from ActivityNet video captions and we use it for the task of sentence completion, where the model is given a context and asked to complete the sentence. The predicted sentence endings generated by StoryEndGen [71] and GPT-2 [9] are collected and used in the following evaluation.

4.4.2 Automatic Metrics

Machine Translation & Summarization In the evaluation of machine translation and text summarization tasks, it is a common practice to compare the predicted text with the reference. Adhering to previous studies, we present results using reference-based

¹<https://duc.nist.gov/duc2004/>

²<https://catalog.ldc.upenn.edu/LDC2011T07>

metrics. For machine translation, we present scores using BLEU- n ($n=1,2,3,4$) [113], METEOR[114], and CIDEr [115]. Meanwhile, for text summarization, we present ROUGE- n ($n=1,2$) [124] precision scores. Additionally, we report the scores of ROUGE-L [124], BERTScore [112], and BLEURT [122] for both tasks.

Open-ended Text Generation In the context of open-ended text generation, where the number of possible answers for a given scenario can be inexhaustible, evaluating the quality of generated text through a comparison with a fixed set of references is challenging. To address this issue, previous studies have proposed to utilize reference-free metrics to evaluate the quality of the generated text. In this work, we experiment with the following reference-free metrics which assess model degeneration: (1) $\text{div-}n = \frac{|\text{unique } n\text{-grams}|}{|\text{total } n\text{-grams}|}$ measures sequence level repetition by computing the portion of duplicate n -grams ($n=2,3,4$) [109]. (2) $\text{diversity} = \prod_{n=2}^4 \text{rep-}n$ measures the diversity of n -grams [74], and assesses the model degeneration. (3) $\text{distinct-}n = \frac{|\text{unique } n\text{-grams}|}{|\text{length of text}|}$ measures the portion of distinct n -grams (here $n=2$) in the text [110]. In addition, we report results on BERTScore [112] and BLEURT [122] for comparison of contextual similarity.

4.4.3 Human Evaluation

We invite Amazon Mechanical Turk³ annotators to evaluate the quality of the generated text. Due to cost constraints, when conducting human evaluation, we randomly sample 1,000 test examples for each dataset, except for DUC2004 which has 500 examples in the test set. Each example is evaluated by three human judges using a 5-point Likert scale, which assessed the fluency, grammar correctness, and factual consistency of the generated text with the reference text. The overall human assessment score is calculated as the mean of the scores obtained from the three aspects. We compute the Pearson

³<https://www.mturk.com/>

Metric	IWSLT'14			WMT'19		
	Original	+IE _{image}	+IE _{text&image}	Original	+IE _{image}	+IE _{text&image}
BLEU-1	21.47	21.38±1.53	21.86±0.82	13.74	14.71±1.19	16.40±0.73
BLEU-2	20.82	21.17±1.45	21.53±0.68	12.50	12.93±1.13	15.11±0.64
BLEU-3	19.17	19.88±1.39	20.31±0.62	11.31	12.07±1.09	13.90±0.58
BLEU-4	17.60	18.57±1.36	19.08±0.60	9.10	9.15±1.06	11.84±0.54
METEOR	20.60	21.44±1.54	21.30±0.99	13.47	14.77±1.33	16.80±0.91
ROUGE	20.55	20.69±1.54	21.26±0.80	11.40	11.58±1.16	14.34±0.68
CIDEr	21.98	22.12±0.24	22.25±0.07	11.82	11.86±0.18	12.05±0.07
BERTScore	23.95	24.02±1.41	24.09±0.65	17.01	17.08±1.22	18.88±0.78
BLEURT	22.93	22.99±0.64	23.40±0.41	18.81	19.36±0.82	19.59±0.37

Table 4.1: The effect of applying our IMAGINE similarities on automatic metrics for machine translation, reflected in the Pearson correlation with human judgments. The image generation process is conducted over five different random seeds for each piece of text. We report the mean and standard deviation of the repeated runs. IE: IMAGINE.

correlation [147] between the human scores and the scores obtained from the automatic metrics, and the results are reported as a multiple of 100 for clarity.

4.5 Results and Analysis

4.5.1 Main Results

Machine Translation Table 4.1 presents the results of the system-level Pearson correlation with human judges when extending the IMAGINE similarity metric to various existing automatic natural language generation (NLG) metrics on the IWSLT'14 and WMT'19 German-to-English datasets. The results demonstrate that the addition of both IMAGINE_{image} and IMAGINE_{text&image} improves the Pearson correlation for all metrics listed. Among the two variants, the mean of IMAGINE_{text&image} consistently performs better on both datasets. It is observed that there is a more substantial variance in IMAGINE_{image}, which is attributed to the difference in the images generated by the StableDiffusion model [4] due to varying random seed and initialization values. As a result,

Metric	DUC2004			GigaWord		
	Original	+IE _{image}	+IE _{text&image}	Original	+IE _{image}	+IE _{text&image}
ROUGE-1	13.66	16.77 ±1.31	13.45±0.80	12.90	17.52 ±0.73	16.78±0.66
ROUGE-2	9.74	15.71 ±1.65	11.19±1.08	7.75	14.26 ±0.83	13.33±0.77
ROUGE-L	13.14	16.35 ±1.47	13.17±0.95	14.31	17.44 ±0.77	16.78±0.70
BERTScore	19.44	20.60 ±1.29	20.26±0.78	19.59	20.47 ±0.64	20.10±0.57
BLEURT	23.59	25.20 ±0.72	24.46±0.42	20.23	21.08 ±0.39	20.74±0.35

Table 4.2: The effect of applying our IMAGINE similarities on automatic metrics for text summarization, reflected in the Pearson correlation with human judgments. The image generation process is conducted over five different random seeds for each piece of text. We report the mean and standard deviation of the repeated runs. IE: IMAGINE.

IMAGINE_{image}, which compares two machine-generated images, has a higher standard deviation compared to IMAGINE_{text&image}.

Abstractive Text Summarization The results in Table 4.2 demonstrate the system-level Pearson correlation with human judges when incorporating our IMAGINE similarity into existing automatic NLG metrics on the DUC2004 and Gigaword datasets. In alignment with the observations made in the machine translation task, the addition of both IMAGINE_{image} and IMAGINE_{text&image} results in an improvement in Pearson correlation across all metrics. On the two summarization datasets, we notice that the correlation after incorporating IMAGINE_{image} exhibits higher mean values along with larger variances compared to the correlation with IMAGINE_{text&image}.

Open-ended Text Generation For the sentence completion task, we conduct evaluations in two setups. In the reference-based evaluation, we compare the predicted sentence ending with the ground-truth ending provided in the dataset. In reference-free evaluation, we compare the predicted sentence ending with the input context. This setup is designed to assess the coherence of the prediction with the input context, as it is hypothesized that a high-quality prediction for open-ended text generation should be consistent

Metric	Reference-based			Reference-free		
	Original	+IE _{image}	+IE _{text&image}	Original	+IE _{image}	+IE _{text&image}
div-2	27.21	28.01±0.49	28.08±0.34	27.21	26.51±0.42	27.29±0.58
div-3	26.80	27.67±0.49	27.78±0.35	26.80	26.17±0.43	26.98±0.59
div-4	26.20	27.14±0.48	27.28±0.36	26.20	25.71±0.44	26.55±0.60
diversity	27.40	28.19±0.41	28.23±0.30	27.40	26.89±0.36	27.55±0.50
distinct-2	26.72	27.76±0.56	27.90±0.40	26.72	25.54±0.48	26.49±0.66
BERTScore	23.47	25.92±0.50	25.43±0.36	25.10	23.47±0.56	25.26±0.78
BLEURT	19.99	22.47±0.83	21.55±0.72	18.70	19.67±0.88	20.56±1.25

Table 4.3: The effect of applying our IMAGINE similarities on ActivityNet for open-ended text generation, reflected in the Pearson correlation with human judgments. In the “Reference-based” setting, we compare the predictions with the references, while in the “Reference-free” setting, we compare the predictions with the input contexts. The image generation process is conducted over five different random seeds for each piece of text. We report the mean and standard deviation of the repeated runs. IE: IMAGINE.


with the input context.

The results of extending our IMAGINE similarity metric to existing automatic NLG metrics for the sentence completion task on the ActivityNet dataset are shown in Table 4.3. In the reference-based setting, both IMAGINE variants demonstrate improvement over the listed metrics and exhibit comparable performances. In the reference-free setting, the introduction of $\text{IMAGINE}_{\text{text\&image}}$ continues to enhance the Pearson correlation, while the implementation of $\text{IMAGINE}_{\text{image}}$ results in a decrease in correlation. One possible reason for the decline in correlation when $\text{IMAGINE}_{\text{image}}$ is used in the reference-free setting of the sentence completion task on ActivityNet (which is comprised of video captions) is that, despite the requirement for the predicted continuation to be coherent with the given context, the visual representation of the context and continued text may differ greatly in this scenario (e.g., due to a plot twist in the video). Consequently, direct comparison of images through $\text{IMAGINE}_{\text{image}}$ may result in a decrease in correlation. However, the inherent coherence between the input text and the continued text may be captured through cross-modal comparison, which may explain why $\text{IMAGINE}_{\text{text\&image}}$

Src.: Also entschied ich mich eines tages den filialleiter zu besuchen, und ich fragte den leiter, "funktioniert dieses modell, dass sie den menschen all diese möglichkeiten bieten wirklich?"

Ref.: So I one day decided to pay a visit to the manager, and I asked the manager, "is this model of offering people all this choice really working?"

Hyp.: So I decided to visit the filialler one day, and I asked the ladder, "does this model work that you really offer to the people all these possibilities?"



Metric	Score
BLEU-1	69.70
ROUGE-L	50.00
BERTScore	58.88
BLEURT	55.73
ImaginE _{image&text}	23.85

Figure 4.3: A case study on IWSLT'14 German-to-English translation with images rendered by StableDiffusion-v2-1. Src.: input source text. Ref.: reference text. Hyp.: generated hypothesis text.

still improves the correlation for the listed metrics.

4.5.2 Performance Analysis

Why is ImaginE helpful? As shown in Tables 4.1 to 4.3, the incorporation of certain variants of IMAGINE improves the correlation between the reference-based and reference-free metrics and human scores in the majority of cases. This indicates the usefulness of extending text-only metrics with multi-modal knowledge. However, how do these machine imaginations actually help text understanding and evaluation? In this section, we further explore how and why IMAGINE works. We first provide a case study to show the uniqueness of IMAGINE over text-based metrics, then systematically analyze the effectiveness of our method from different perspectives.

Case Study Figure 4.3 shows an example in which IMAGINE effectively detects the dissimilarity in keywords between two text snippets. Despite the similarity in sentence structure between the reference and hypothesis, the crucial distinction lies in the inclusion of the terms “manager” and “ladder”. While traditional automatic metrics that rely

Metric	Original	+IE _i (dVAE)	+IE _i (BigGAN)	+IE _i (VQ-GAN)	+IE _i (SD)
ROUGE-1	13.7	15.9 ± 0.9	15.7 ± 1.0	15.9 ± 0.8	16.8 ± 1.3
ROUGE-2	9.7	14.9 ± 1.2	14.6 ± 1.3	14.9 ± 1.0	15.7 ± 1.7
ROUGE-L	13.1	16.0 ± 1.0	15.8 ± 1.1	16.0 ± 0.9	16.4 ± 1.5

Table 4.4: The Pearson correlations with human judges when using IMAGINE_{image} (IE_i) to augment ROUGE-1/2 and ROUGE-L on DUC2004. We compute four sets of IMAGINE_{image} similarity scores (mean±std) with dVAE, BigGAN, VQGAN, and StableDiffusion (SD).

on n -grams matching (BLEU, ROUGE) or textual embedding comparison (BERTScore, BLEURT) may exhibit high scores, the quality of the generated text remains questionable. In contrast, IMAGINE generates distinctive images and exhibits a relatively low cross-modal similarity score, which aligns with human perception.

Sensitivity to Different Image Generation Backbones In previous sections, we utilize StableDiffusion [4] as the image generation backbone for IMAGINE . Here, we examine the influence of the image generation backbone on the evaluation performance of IMAGINE by conducting experiments on the DUC2004 dataset for summarization and comparing StableDiffusion with three alternative models: dVAE [81], BigGAN [148], and VQGAN [59]. The results, as shown in Table 4.4, indicate comparable performance of IMAGINE_{image} with dVAE and VQGAN, both of which outperform BigGAN across all metrics. StableDiffusion achieves the highest mean value, but also displays the largest variance among the models. These findings highlight the significance of considering the image generation architecture when evaluating text, as it can result in varying machine-generated images and affect the final evaluation outcomes.

Reliability of Machine-Generated Images The reliability of IMAGINE ’s visualization capability is further evaluated on the Flickr30k Entities dataset [149], which consists of annotated image captions. We randomly sample 100 captions and use the four gen-

	dVAE	BigGAN	VQGAN	StableDiffusion
Entity Recall	88.8%	41.2%	87.2%	94.1%

Table 4.5: Entity recall rate on the visualizations for Flickr30k captions. We report results for images generated by dVAE, BigGAN, VQGAN, and StableDiffusion.

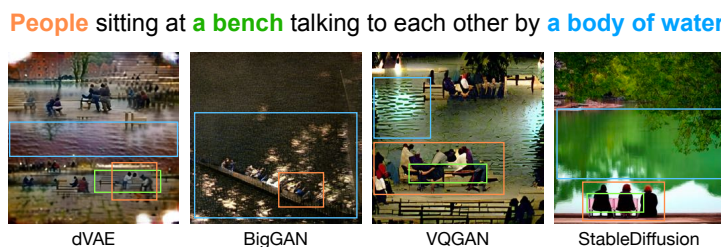


Figure 4.4: An example caption from Flickr30k Entities, and images rendered by dVAE, BigGAN, VQGAN and StableDiffusion. The bounding boxes point to the visualizations of the entities marked in the same color.

erative backbones to render images. We present the captions and generated images to human annotators, and ask them to indicate if the entities mentioned in the captions are visually represented. The results, in terms of entity recall rates, are presented in Table 4.5. A higher recall rate indicates that the text-to-image generator is more capable of visualizing the content described in the text. The results show that StableDiffusion has the highest entity recall rate of approximately 94%, followed closely by dVAE and VQGAN. In contrast, BigGAN has the lowest recall rate of around 41%. An example of entity recall for a set of images generated by the four generative backbones is shown in Figure 4.4.

Syntax Importance to Machine-Generated Images We evaluate the significance of different syntax tokens in the image generation process using the DUC2004 summarization dataset. We utilized the Stanza [150] part-of-speech (POS) tagger to parse the text and created ablated examples by masking out a token of a specific syntax tag.⁴ The visual similarity of the images generated from the ablated examples is then compared to

⁴We report Universal POS tags in this study: <https://universaldependencies.org/u/pos/>

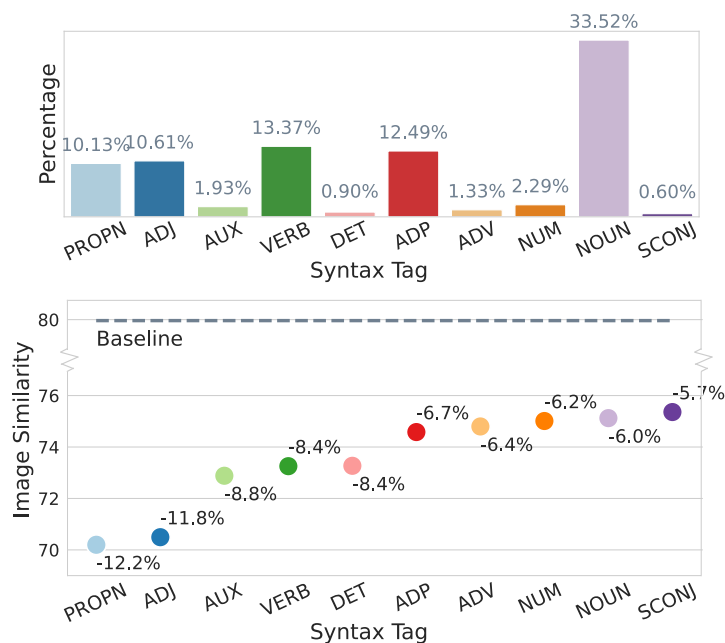


Figure 4.5: The influence on visualization when masking tokens of different syntax tags. Upper: The occurrence frequency of each syntax tag in DUC2004. Lower: The relative image similarity decrease after masking each syntax tag. Baseline: The average intra-group pairwise image similarity. The top-10 syntax tags that have the most significant impact on visualization are listed here.

the visualization of the original text. The results, as reported in Figure 4.5, indicated that the removal of PROPN and ADJ tags has a significant impact on the visualization results, resulting in a 12% decrease in visual similarity. Conversely, removing NOUN tokens has a comparatively smaller effect. The most frequent NOUN, PROPN, and ADJ tokens in the DUC2004 dataset were listed in Table 4.6. For DUC2004 built upon new clusters, PROPN and ADJ tokens cover concrete concepts such as nations, corporations, and celebrities, while NOUN tokens involve more abstract concepts such as government, party, and right. For this particular dataset, our IMAGINE approach pays more attention to PROPN and ADJ tokens that are easier to visualize by nature.

Which Imagine Variant to Report? From Tables 4.1 to 4.3, we can see a mixed trend of performance between the two IMAGINE variants. In general, $\text{IMAGINE}_{\text{text}\&\text{image}}$

POS Tag	10 Most Frequent Tokens
NOUN	president, minister, government, space, party, station, budget, game, right, arrest
PROPN	U.S., Clinton, China, Korea, Gaza, Microsoft, Congo, Israel, Livingston, Lebanon
ADJ	new, prime, Russian, international, Asian, possible, Cambodian, first, human, economic

Table 4.6: The most frequent NOUN, PROPN, and ADJ tokens in DUC2004.

	IWSLT'14	WMT'19	DUC2004	GigaWord	AN(w/ ref)	AN(w/o ref)
IE_i	19.1±1.5	13.8±1.7	10.6±1.5	15.9±1.1	18.9±1.5	16.8±1.9
$IE_{t&i}$	18.0±1.5	12.9±1.8	9.6±1.6	15.3±1.1	18.4±1.6	18.2±1.8

Table 4.7: The Pearson correlation between IMAGINE variants and human assessments on each dataset. Here we use $IMAGINE_{image}$ (IE_i) and $IMAGINE_{text\&image}$ ($IE_{t\&i}$) as two individual metrics. AN: ActivityNet, “w/ ref”: reference-based, “w/o ref”: reference-free.

has smaller variances among repeated runs. Nevertheless, we would still suggest reporting both IMAGINE variants since they conduct comparisons from different aspects, with $IMAGINE_{image}$ comparing similarity within the visual modality, while $IMAGINE_{text\&image}$ compares cross-modal similarity.

Imagine as a Standalone Metric Table 4.7 presents the Pearson correlation with human evaluations on each dataset when utilizing the two IMAGINE variants as standalone metrics. The results reveal that both IMAGINE variants demonstrate a lower correlation compared to other metrics as reported in Tables 4.1 to 4.3. Additionally, the scores produced by IMAGINE are not determinate, given the probabilistic nature of text-to-image models that generate various images with different random seeds. Hence, IMAGINE may not be an optimal choice as a standalone metric. Nonetheless, it is important to emphasize the capability of IMAGINE in introducing multimodal aspects to traditional text-only metrics. In this study, integrating IMAGINE with text-only metrics

leads to an improvement in the Pearson correlation with human evaluations. Future work may explore alternative methods of integrating multimodal information in text evaluation.

Part II

LLM for Vision and Multimodal Studies

Chapter 5

Integrating GPT-k for Efficient Editing in Text-to-Image Generation

5.1 Introduction

The task of text-to-image (T2I) generation, which involves generating images from natural language descriptions, holds significant potential to create new avenues and job opportunities for content creators while also providing insights into the grounding of natural language in the visual world through the application of generative AI. A number of models have demonstrated exceptional performance in terms of image quality, such as StableDiffusion [4], Midjourney [5], Imagen [6], and DALLE-2 [7]. Despite the popularity of T2I generation and the ability of these models to generate impressive images, the difficulty of “prompt-engineering” – which is writing accurate prompts to describe the desired image in this scenario – still remains a significant challenge. Users often need to edit the prompt for several rounds before arriving at a satisfactory image, which makes the process of T2I generation time-consuming and laborious (and expensive for commercialized models). Figure 5.1 shows the `#edits` distribution in the editing traces

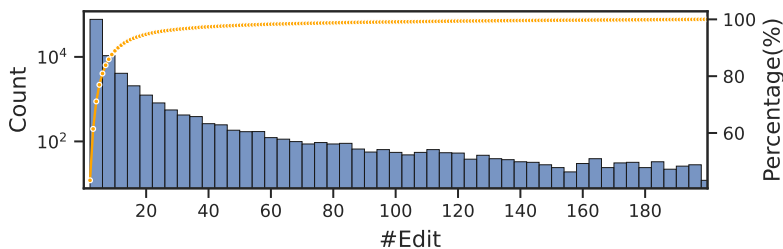


Figure 5.1: The histogram plots `#edit` per trace in DiffusionDB [151] while the lineplot shows the cumulative percentage of traces less than certain `#edit`. The y-axis on the left is on log-scale. On average, there are 6.9 edits being made in each user editing trace.

made by StableDiffusion Discord server users [151]. This phenomenon highlights the need to improve the efficiency and effectiveness of prompting T2I models.

Recently, large language models (LLMs) such as GPT- k [8, 9, 10] have demonstrated impressive abilities in text generation. This leads to the natural question of whether these models can be utilized to improve the T2I prompting process [152]. However, LLMs and T2I models have different architectures and are often trained on different modalities, which makes it challenging to integrate them seamlessly.

In this paper, we conduct a series of experiments and analyses on user editing traces on StableDiffusion,¹ and attempt to modify the T2I prompts with eight GPT- k models. The primary objective of our research is to examine the potential of modifying T2I prompts with GPT- k models. More specifically, we aim to investigate the common edits made by humans and by GPT- k , as well as the performance of prompting T2I generation with GPT- k models. Additionally, we aim to identify and investigate possible factors that might influence the performance of GPT- k in T2I generation tasks.

Through our experiments, we observe that:

- While GPT- k models tend to focus more on inserting modifiers, human users have a greater tendency towards replacing words and phrases, which may include spon-

¹Our experiments are conducted upon StableDiffusion since it is a wide-adopted open-source large text-to-image generative model with SoTA performance at the time of this research around Dec. 2022.

taneous changes to the subject matter of the prompt.

- Modifying the T2I prompt with GPT- k models may not necessarily result in a direct increase in similarity to the final target image in the editing trace. Instead, the edit suggested by GPT- k may be closely related to intermediate editing steps, and the percentage of remaining edits may decrease by 20-30% if the edit suggested by GPT- k is adopted.
- These findings suggest that instead of attempting to predict spontaneous changes made by human users on the primary subject matter, GPT- k models are more effective in improving prompts by rewriting and performing edits related to modifiers adjustment.

5.2 Research Questions and Settings

To investigate the potential of modifying T2I prompts with GPT- k models, we conduct a series of experiments and analysis, aiming to answer the following questions:

1. *To what extent can GPT- k models help prompt text-to-image generation?*
2. *What are the common types of edits made by humans and by GPT- k models?*
3. *What are the factors that may influence GPT- k prompting text-to-image generation?*

We describe the dataset, models and evaluation metrics used in this study below.

Dataset DiffusionDB-2M [151] scrapes 2M groups of user prompts, hyperparameters and images generated by StableDiffusion [4] from the official Stable Diffusion Discord server. We group the prompts by anonymized `user_id`, then cluster the user prompts

GPT- k	Model Name	#Parameters
GPT-2 [9]	gpt2-base	117M
	gpt2-medium	345M
	gpt2-large	774M
	gpt2-xl	1.2B
GPT-3 [8]	text-ada-001	350M
	text-babbage-001	1.3B
	text-curie-001	6.7B
GPT-3.5 [10]	text-davinci-003	175B

Table 5.1: The names and corresponding parameter sizes of the GPT- k models covered in our study.

into *traces of edits* based on the prompt contents. More specifically, we encode prompts with Universal Sentence Encoder [153], then cluster upon the prompt embeddings with DBSCAN [154]. The order of edits within each trace is determined by the timestamps. We receive 100k traces of edits, the mean `#edit` per trace is 6.9, with a standard deviation of 16.1. Figure 5.1 plots the `#edit` in the clustered traces, which shows a long-tail distribution with about 5% of the traces having more than 20 edits. Thus, in the following experiments, the evaluation results were reported on traces with at most 20 edits.

GPT- k & Text-to-Image Models Table 5.1 lists the names and parameter sizes of the eight GPT- k models we covered in this study. We conduct T2I experiments with the open-source StableDiffusion-v1-4 [4], which is used to render images in DiffusionDB [151].

Annotations For each trace of length n , we denote the prompts as $(\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n)$, and the generated images as $(\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_n)$. We refer to the GPT- k -modified prompt as \mathbf{t}' , and refer to the image rendered from the modified prompt as \mathbf{i}' .

Metrics We use the cosine similarity of CLIP-ViT/B-32 [23] visual features to evaluate the similarity between images. Let \mathbf{i}_k denote the image in the original edit trace that is most similar to \mathbf{i}' , we define the Relative Number of Edits (RNE) as $\frac{n-k}{n-1}$. The RNE

metric reflects the relative position of the edit in the original trace that is most similar to the edit suggested by GPT- k and also represents the percentage of remaining edits after the edit suggested by GPT- k is performed.

5.3 Prompting T2I w/ GPT- k

In the following experiments, we only reveal the initial prompts in the user editing traces to GPT- k , and ask the models to perform *one* round of edit.

Procedure We split the 100k trace of edits into two parts, with 30k traces used for evaluations and the remaining 70k serving as heldout set. For each of the listed GPT- k models, we provide the first prompt \mathbf{t}_1 in each evaluation trace to the model, and let GPT- k generate the modified prompt \mathbf{t}' .

GPT-2 models are finetuned with the prompts in the heldout traces for two epochs with a learning rate of $5e - 5$ and a batch size of 128. For GPT-3/3.5 models, an in-context learning approach is adopted. Following previous studies [22, 155], supporting examples for in-context learning are selected by comparing the similarity of the prompt text features and retrieving $(\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_n)$ pairs from the top- m most similar traces. Modified prompts are generated through 16-shot in-context learning with $m=16$.

After receiving the GPT- k -modified prompt \mathbf{t}' , we provide it to StableDiffusion to generate image \mathbf{i}' . The effectiveness of GPT- k in prompting T2I generation is evaluated by comparing the similarity of the generated image \mathbf{i}' to images in the original trace. The results reported in the following sections are the mean of three repeated runs.

Automatic Evaluation The CLIP cosine similarity scores, as listed in Table 5.2, are used to evaluate image similarity. Two model-agnostic baselines are established for

Model	$\mathbf{i}_1-\mathbf{i}_n$	$\mathbf{i}_{n-1}-\mathbf{i}_n$	$\mathbf{i}'-\mathbf{i}_n$	$\mathbf{i}'-\mathbf{i}_{MS}$	RNE(%)
gpt2-base			69.58	80.16	75.28
gpt2-medium			69.63	80.39	78.28
gpt2-large			69.70	80.50	78.88
gpt2-xl	71.72	74.82	69.57	80.37	75.25
gpt3-ada			66.95	76.37	69.43
gpt3-babbage			68.79	78.81	72.33
gpt3-curie			68.45	78.28	71.41
gpt3.5-davinci			68.79	78.09	69.22

Table 5.2: The CLIP cosine similarity scores between images and the relative number of edits (RNE). Here, \mathbf{i}_1 , \mathbf{i}_{n-1} , \mathbf{i}_n denotes the first, last-but-one, and last image in the trace of edits; \mathbf{i}' is the image generated from the modified prompt, and \mathbf{i}_{MS} is the image that is most similar to \mathbf{i}' with regard to CLIP cosine similarity. RNE scores suggest a 20-30% decrease in the percentage of remaining edits if adopting edits suggested by GPT- k .

comparison: the similarity between the first and last images ($\mathbf{i}_1-\mathbf{i}_n$), and the similarity between the last-but-one and last images ($\mathbf{i}_{n-1}-\mathbf{i}_n$).

We denote \mathbf{i}_{MS} as the image in the original trace that is most similar to the generated image \mathbf{i}' (has the highest CLIP cosine similarity score). Examining results in Table 5.2, we notice that the image \mathbf{i}' generated from the modified prompt may not be directly similar to the final target \mathbf{i}_n , as $(\mathbf{i}'-\mathbf{i}_n)$ is lower than the baselines. However, it appears that \mathbf{i}' may be related to the intermediate steps in the editing trace, as evidenced by the significantly higher similarity between \mathbf{i}' and \mathbf{i}_{MS} compared to the baselines. RNE scores show that, \mathbf{i}' is most similar to images in the first one-third of the trace, and that the percentage of remaining edits decreases by 20-30% if the edit suggested by GPT- k is adopted.

Human Evaluation For each editing trace, we present MTurk annotators with the initial prompt and image (\mathbf{t}_1 , \mathbf{i}_1), and two candidate edits: (1) the GPT- k -modified prompt, \mathbf{t}' ; (2) the human edit, \mathbf{t}_{MS} , from the original editing trace. Here, \mathbf{t}_{MS} is the corresponding prompt to \mathbf{i}_{MS} , which has the highest CLIP cosine similarity with \mathbf{i}' . The

	Effectiveness			Likelihood		
	Win(%)	Tie(%)	Lose(%)	Win(%)	Tie(%)	Lose(%)
<code>gpt2-xl</code>	38.57	29.77	31.66	38.99	22.01	38.99
<code>gpt3-curie</code>	40.89	21.33	37.78	39.33	21.56	39.11
<code>gpt3.5-davinci</code>	39.58	21.67	38.75	38.13	25.00	36.87

Table 5.3: Human evaluation results of head-to-head comparison between edits suggested by GPT- k and human-made edits. We evaluate the effectiveness of the edit and the likelihood of this edit being adopted by humans. “Win” and “Tie” indicate that GPT- k -suggested edits are better than or comparable to human edits.

annotators are then asked to decide which edit was more effective and more likely to be adopted by human editors, t' or t_{MS} . We evaluate each listed GPT- k model with 200 traces. For each trace, three annotators are invited to provide their judgments.

As shown in Table 5.3, the three GPT- k models all show tight wins against the human edits regarding both effectiveness and likelihood of being adopted. This verifies that the edits made by GPT- k models are similar or comparable to the intermediate steps in the human editing trace.

5.4 Human’s Common Edits vs. GPT- k ’s

Upon examination of the user editing traces, we identify four types of edits commonly made by humans: (1) **Insert**: adding descriptive terms such as modifiers to the prompt to specify the style, artistic technique, camera view, lighting, etc; (2) **Delete**: removing certain terms; (3) **Swap**: changing the order of certain terms in the prompt; (4) **Replace**: changing the modifiers or the main subject of the prompt. Table 5.4 presents an excerpt from an editing trace and provides examples of the four types of common edits, including `insert`, `delete`, `swap`, and `replace`.

To count the frequency of edits, we first remove punctuation marks and stopwords using NLTK [156]. We then utilize the SequenceMatcher² to compare the adjacent prompts

²<https://docs.python.org/3/library/difflib.html>




User Input Prompt	Generated Image
circular ornated ceiling highly detailed	
photo of an ornated circular ceiling, full of paintings of angels, centered symmetrical, highly detailed	
SWAP: “circular ornated” → “ornated circular” INSERTION: “full of painting of angels”, “centered symmetrical”	
ornate marble and gold wall, full of paintings of angels, highly detailed	
REPLACE: “ceiling” → “marble and gold wall” DELETION: removed “centered symmetrical”	

Table 5.4: Common types of edits.

in the trace and detect the edit type. Table 5.5 lists the frequency of common edits made by humans and by GPT- k models. We notice a discrepancy between the distribution of human edits and the ones made by GPT- k . Nearly half of human edits pertain to **replace**, followed by **insert** and **delete**. GPT-2 models, due to their autoregressive training nature, have a tendency towards continual generation, resulting in a majority of edits being **insert**. While GPT-3/3.5 also undergoes autoregressive training, its extensive training data and enlarged model size empower it for emergent ability. Unlike GPT-2, which solely receives the specific prompt requiring editing during testing, GPT-3/3.5 is also provided with additional editing instances as supportive exemplars for in-context learning. Consequently, while GPT-2 indeed adheres to its autoregressive inference manner, the emergent capability of GPT-3/3.5 enables it to attempt other edit types that simulate human-like editing behavior. For GPT-3, as the model size increases, we see an increase in the frequency of **insert** and a decrease in **replace**. For GPT-3.5, the most frequent edit is **insert**, followed by **replace**; while **delete** and **swap** are relatively rare.

		Insert	Delete	Swap	Replace
Human	-	29.9%	21.1%	2.0%	47.0%
GPT- k	gpt2-base	95.4%	0.0%	0.0%	4.6%
	gpt2-medium	95.6%	0.1%	0.0%	4.4%
	gpt2-large	95.0%	0.1%	0.0%	4.9%
	gpt2-xl	95.9%	0.0%	0.0%	4.1%
	gpt3-ada	36.5%	14.9%	2.0%	46.6%
	gpt3-babbage	39.6%	18.8%	3.3%	38.3%
	gpt3-curie	42.9%	17.7%	4.1%	35.3%
	gpt3.5-davinci	68.5%	3.9%	2.3%	25.3%

Table 5.5: The distribution of the common types of edits made by human and by GPT- k models.

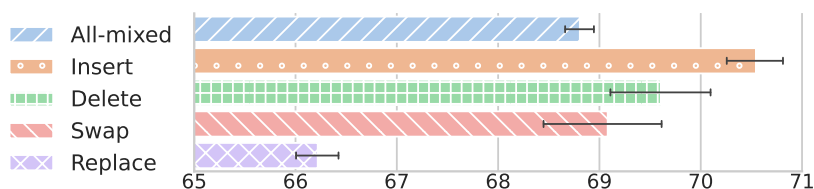


Figure 5.2: The CLIP cosine similarity scores between the image generated from the GPT-3.5-modified prompt and the last image. Results are reported on $\mathcal{S}_{\text{eval}}$ (All-mixed), $\mathcal{S}_{\text{insert}}$, $\mathcal{S}_{\text{delete}}$, $\mathcal{S}_{\text{swap}}$ and $\mathcal{S}_{\text{replace}}$.

5.5 Ablations & Analyses

Effects of Edit Types To investigate the effects of each individual edit type e_i , we create four subsets $\mathcal{S}_{\text{insert}}$, $\mathcal{S}_{\text{delete}}$, $\mathcal{S}_{\text{swap}}$ and $\mathcal{S}_{\text{replace}}$ from the evaluation set $\mathcal{S}_{\text{eval}}$. Each subset \mathcal{S}_{e_i} , comprises of traces that meet the criteria of “if and only if edit e_i is applied on the first prompt can we receive the last prompt.” For each edit type e_i , the image similarity between the image generated from the modified prompt and the last image for traces in \mathcal{S}_{e_i} is calculated and compared to the baseline results obtained from the complete $\mathcal{S}_{\text{eval}}$ that mixes all types of edits.

Figure 5.2 illustrates the impact of different edit types on image similarity. The CLIP cosine similarities of traces that solely consist of **insert**, **delete**, and **swap** edits are higher or comparable to the **all-mixed** baseline. This suggests that GPT- k performs better at adding, removing, and reordering modifiers. Conversely, we observe that

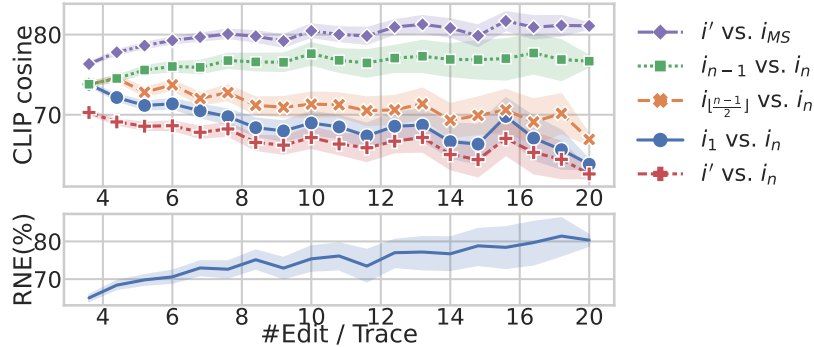


Figure 5.3: The CLIP cosine similarity (upper) and RNE (lower) on traces with `#edit` ranging from 2 to 20. Prompts for T2I generation are modified with GPT-3.5. Here, \mathbf{i}_1 , $\mathbf{i}_{\lfloor \frac{n-1}{2} \rfloor}$, \mathbf{i}_{n-1} , \mathbf{i}_n denotes the first, middle, last-but-one, and last image in the trace of edits; \mathbf{i}' is the image generated from the modified prompt, and \mathbf{i}_{MS} is the image that is most similar to \mathbf{i}' .

replace edits lead to lower image similarities. This is likely due to the fact that the replace edit sometimes results in a change of the subject matter, which can drastically alter the painting. It is worth noting that shifting the primary subject matter of the painting is a relatively spontaneous action made by humans. The vast number of potential replacements makes it particularly difficult for GPT- k to accurately select the desired edit.

Effects of #Edits Figure 5.3 depicts how the CLIP cosine similarity changes with the `#edit` in the trace. As `#edit` increases, the similarity between the last image \mathbf{i}_n and those at the beginning or middle of the trace (\mathbf{i}_1 or $\mathbf{i}_{\lfloor \frac{n-1}{2} \rfloor}$) decreases. This trend may be attributed to the higher likelihood of changing primary subject matters in longer traces. Meanwhile, the similarity between the last-but-one and the last images (\mathbf{i}_{n-1} vs. \mathbf{i}_n) remains consistent, suggesting that the majority of edits made towards the end of the prompt editing process are minor in nature. As the trace of edits gets longer, the similarity between the image \mathbf{i}' generated from the modified prompt and the most similar image \mathbf{i}_{MS} in the trace remains constant, while the RNE metric gradually increases to

approximately 80%. This indicates that the modified prompt is related to the early edits in the trace. This aligns with our previous findings, which suggest that GPT- k is proficient in rewriting prompts and adjusting modifiers, but may struggle to predict spontaneous changes in the main subject matter of the painting.

Chapter 6

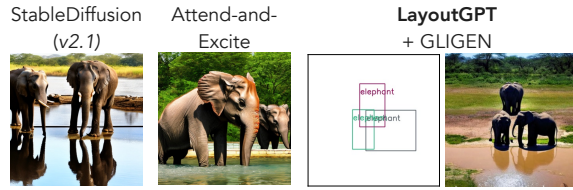
Compositional Visual Planning and Generation with Large Language Models

6.1 Introduction

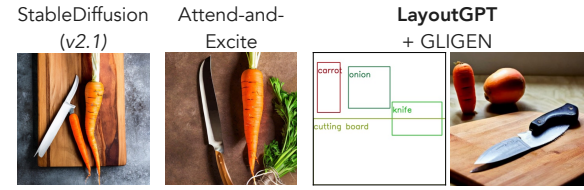
Can Large Language Models (LLMs) comprehend visual concepts and generate plausible arrangements in visual spaces? Recently, LLMs have shown significant advancement in various reasoning skills [157, 158] that remain challenging to visual generative models. For instance, text-to-image generation (T2I) models suffer from generating objects with specified counts, positions, and attributes [159, 160]. 3D scene synthesis models face challenges in preserving furniture within pre-defined room sizes [161]. Addressing these issues necessitates the development of compositional skills that effectively arrange components in a coherent manner, accurately reflecting object specifications and interactions.

Visual layout is an essential symbolic representation that has been widely studied as it reflects the compositions of a visual space [162, 163, 164, 165]. For instance, layout

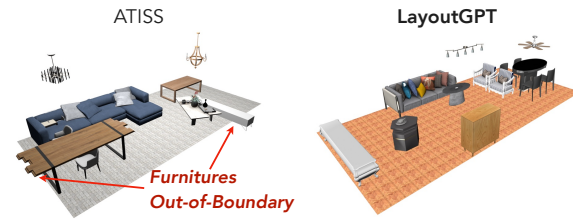
[2D Numerical Reasoning] *There are three elephants standing beside a pool of water.*



[2D Spatial Reasoning] *A carrot and some onion next to a knife on a cutting board.*



[3D Living Room] *Room Type: Living Room;*
Room Size: 7.7m x 3.6m



[3D Bedroom] *Room Type: Bedroom;*
Room Size: 3.0m x 4.8m

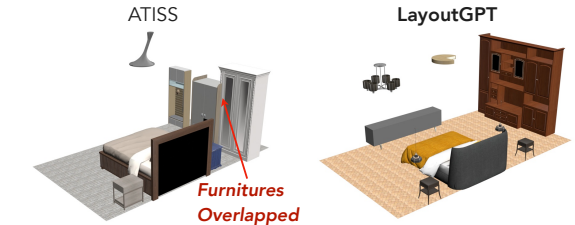


Figure 6.1: Generated layouts from LayoutGPT in 2D images and 3D indoor scenes. LayoutGPT can serve as a visual planner to reflect challenging numerical and spatial concepts in visual spaces.

generation models [166, 167, 168, 169, 165] can be combined with region-controlled image generation methods [170, 171] to improve image compositionality [172]. But unlike LLMs, these models are restricted to discrete categories or have limited reasoning skills for complicated text conditions. Recently, LLMs like ChatGPT [10], are adopted as a centralized module of frameworks or systems where multiple foundational computer vision models are integrated. Through defined action items or API calls, LLMs can interact with visual generative models to extend the systems' capability into image generation tasks [173].

Despite the advancement, existing approaches that involve the collaboration between LLMs and image generation models are either limited to executing the latter through program generation or using LLMs for language data augmentation for image editing [174]. Current LLM-based systems fail to improve the compositional faithfulness of a generated image by simply using T2I models through API calls. While one could additionally integrate models that synthesize images with the guidance of layouts [170, 171],

keypoints [170], or sketches [175, 176], users still have to create fine-grained inputs on their own, leading to extra efforts and degraded efficiency compared to pure language instructions.

To address these challenges, we introduce **LayoutGPT**, a training-free approach that injects visual commonsense into LLMs and enables them to generate desirable layouts based on text conditions. Despite being trained without any image data, LLMs can learn visual commonsense through in-context demonstrations and then apply the knowledge to infer visual planning for novel samples. Specifically, we observe that representing image layouts is highly compatible with how style sheet language formats images on a webpage. Therefore, as LLMs are trained with program data, constructing layouts as structured programs may enhance LLMs’ ability to “imagine” object locations from merely language tokens. Our programs not only enable stable and consistent output structures but also strengthen LLMs’ understanding of the visual concepts behind each individual attribute value. When combined with a region-controlled image generation model [170], LayoutGPT outperforms existing methods by 20-40% and achieves comparable performance as human users in generating plausible image layouts and obtaining images with the correct object counts or spatial relations.

In addition, we extend LayoutGPT from 2D layout planning to 3D indoor scene synthesis. With a slight expansion of the style attributes, LayoutGPT can understand challenging 3D concepts such as depth, furniture sizes, and practical and coherent furniture arrangements for different types of rooms. We show that LayoutGPT performs comparably to a state-of-the-art (SOTA) supervised method. Our experimental results suggest that LLMs have the potential to handle more complicated visual inputs. Our contribution can be summarized as the following points:

- We propose LayoutGPT, a program-guided method to adopt LLMs for layout-

based visual planning in multiple domains. LayoutGPT addresses the *inherent* multimodal reasoning skills of LLMs and can improve end-user efficiency.

- We propose Numerical and Spatial Reasoning (NSR-1K) benchmark that includes prompts characterizing counting and positional relations for text-to-image generation.
- Experimental results show that LayoutGPT effectively improves counting and spatial relations faithfulness in 2D image generation and achieves strong performance in 3D indoor scene synthesis. Our experiments suggest that the reasoning power of LLMs can be leveraged for visual generation and handling more complicated visual representations.

6.2 Related Work

Image Layout Generation Layout generation has been an important task for automatic graphical design for various scenarios, including indoor scenes [177, 178], document layouts [179, 180, 181], and graphical user interface [182]. Previous work has proposed various types of models that need to be trained from scratch before generating layouts. LayoutGAN [169] is a GAN-based framework to generate both class and geometric labels of wireframe boxes for a fixed number of scene elements. LayoutVAE [167] generates image layouts conditioned on an input object label set. Transformer-based methods are proposed to enhance flexibility in the layout generation process. For instance, LayoutTransformer [166] adopts self-attention to learn contextual relations between elements and achieve layout completion based on a partial layout input. BLT [168] proposes a hierarchical sampling policy so that any coordinate values can be modified at the sampling stage to enable flexible and controlled generation. However, existing methods are

restricted to class labels and fail to reason over numerical and spatial concepts in text conditions. In contrast, LayoutGPT can convert challenging textual concepts to 2D layouts and generate free-form, detailed descriptions for each region.

Compositional Image Generation Recent studies have shown that text-to-image generation (T2I) models suffer from compositional issues such as missing objects, incorrect spatial relations, and incorrect attributes [160, 183]. StructureDiffusion [159] proposes to adjust text embeddings by utilizing prior knowledge from linguistic structures. Attend-and-Excite [184] optimizes attention regions so that objects attend on separate regions. Another line of work strives to introduce extra conditions as inputs. For example, ReCo [171], GLIGEN [170], and Layout-Guidance [185] can generate images based on bounding box inputs and regional captions. Wu et al. [172] combines a layout generator and a region-controlled method to achieve accurate generation results. While we focus on layout generation, we also employ layout-to-image models to generate final images and show the effectiveness of LayoutGPT.

Indoor Scene Synthesis Indoor scene synthesis aims at generating reasonable furniture layouts in a 3D space that satisfies room functionality. Early work adopting autoregressive models requires supervision of 2D bounding boxes and other visual maps [177]. Later, SceneFormer [186] proposes to apply a set of transformers to add furniture to scenes. While previous work adopts separate models to predict different object attributes, ATISS [187] demonstrates that a single transformer model can generate more realistic arrangements while being more efficient. In this work, we investigate leveraging LLMs to achieve scene synthesis without any fine-tuning.

LLMs for Vision Language inputs have been an essential part of many vision language tasks [188, 189, 190, 191]. With the strong generalization ability of contemporary

LLMs, recent work attempts to adapt the power of LLMs on multimodal tasks [192, 193]. For instance, multimodal chain-of-thought [194] trained a model to incorporate visual inputs as rationales for question answering. Koh et al. [195] proposes to learn translation parameters to map embeddings between visual and language domains such that an LLM can ground on both modalities. VisProg [196] and ViperGPT [197] use LLMs to design modular pseudocode instructions or executable Python programs to achieve visual reasoning. LLMScore [198] leverages LLMs to evaluate text-to-image models. Visual ChatGPT [173] proposes a prompt manager that supports the execution of various image generation models. In this work, we directly involve LLMs in the generation process by leveraging LLMs to design visual layouts through in-context learning and structured representations.

6.3 Method

6.3.1 Overview

Given a condition \mathcal{C} , the goal of layout generation is to predict a set of tuples $\mathcal{O} = \{\mathbf{o}_j | j = 1, 2, \dots, n\}$ where each tuple \mathbf{o}_j denotes the layout information of a 2D or 3D bounding box of object j . In image planning, \mathcal{C} is the input text prompt, \mathbf{o}_j consists of a category c_j , bounding box location $\mathbf{t}_j = (x_j, y_j) \in \mathbb{R}^2$ and bounding box size $\mathbf{s}_j = (w_j, h_j) \in \mathbb{R}^2$, i.e. $\mathbf{o}_j = (c_j, \mathbf{t}_j, \mathbf{s}_j)$. Similarly, in 3D scene synthesis, \mathcal{C} specifies the room type and room size, \mathbf{o}_j consists of category c_j , location $\mathbf{t}_j \in \mathbb{R}^3$, size $\mathbf{s}_j \in \mathbb{R}^3$, and orientation $\mathbf{r}_j \in \mathbb{R}$, i.e. $\mathbf{o}_j = (c_j, \mathbf{t}_j, \mathbf{s}_j, \mathbf{r}_j)$. While c_j can be modeled as a discrete value, our method directly predicts the category text.

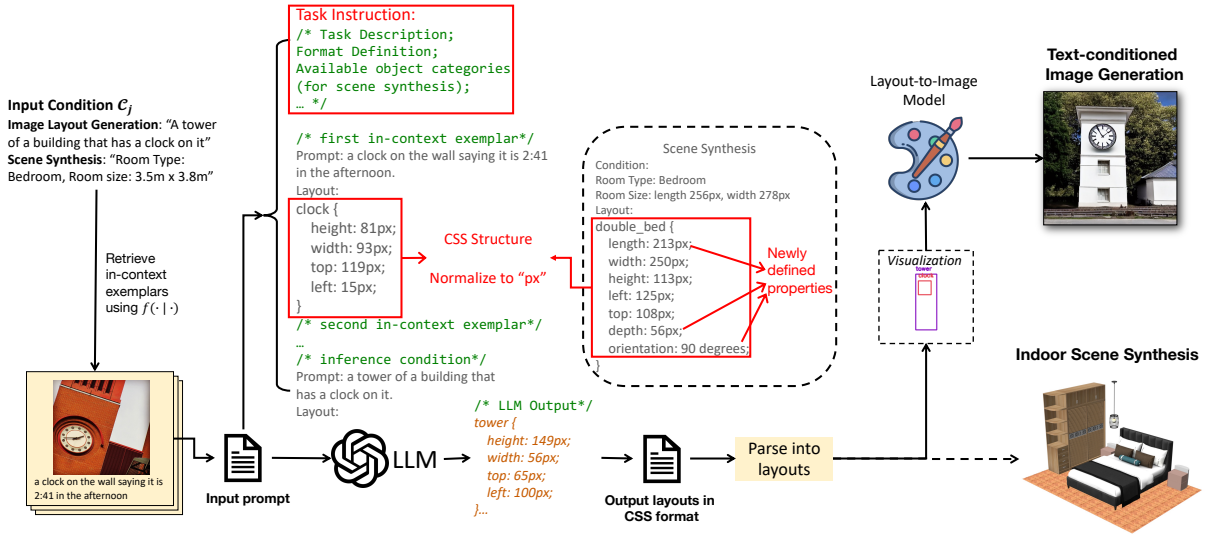


Figure 6.2: The overview process of our LayoutGPT framework performing 2D layout planning for text-conditioned image generation or 3D layout planning for scene synthesis.

6.3.2 LayoutGPT Prompt Construction

As is shown in Figure 6.2, LayoutGPT prompts consist of three main components: **task instructions**, and in-context exemplars in **CSS structures** with **normalization**.

CSS Structures In autoregressive layout generation, \mathbf{o}_j is usually modeled as a plain sequence of values, i.e. $(c_1, x_1, y_1, w_1, h_1, c_2, x_2, \dots)$ [166, 168]. However, such a sequence can be challenging for LLMs to understand due to underspecified meaning of each value. Therefore, we seek a structured format that specifies the physical meaning of each value for LLMs to interpret spatial knowledge. We realize that image layouts are highly similar to how CSS (short for Cascading Style Sheets) formats the layout of a webpage and defines various properties of the `img` tag in HTML. For instance, x_j, y_j corresponds to the standard properties `left` and `top`, while w_j, h_j corresponds to `width` and `height` in CSS. As LLMs like GPT- k are trained with code snippets, formatting image/scene layouts in CSS structures potentially enhances the LLMs' interpretation of the spatial

meaning behind each value. Therefore, as is shown in Figure 6.2, we place category name c_j as the selector and map other attribute values into the declaration section following standard CSS styles.

Task Instructions & Normalization Similar to previous work in improving the prompting ability of LLMs [10, 199, 200], we prepend task instructions to the prompt to specify the task goal, define the standard format, unit for values, etc. Besides, as the common length unit of CSS is pixels (px), we normalize each property value based on a fixed scalar and rescale the value to a maximum of 256px. As will be shown in later sections (Sec. 6.4.4 & 6.5.4), all three components play important roles in injecting visual commonsense into LLMs and improving generation accuracy.

6.3.3 In-Context Exemplars Selection

Following previous work [22, 155], we select supporting demonstration exemplars for in-context learning based on retrieval results. Given a test condition \mathcal{C}_j and a support set of demonstrations $\mathcal{D} = \{(\mathcal{C}_k, \mathbf{o}_k) | k = 1, 2, \dots\}$, we define a function $f(\mathcal{C}_k, \mathcal{C}_j) \in \mathbb{R}$ that measures the distances between two conditions. For 2D text-conditioned image layout generation, we adopt the CLIP [23] model to extract text features of \mathcal{C}_j (usually a caption) and the image feature of \mathcal{C}_k and measure the cosine similarity between them. For the 3D scene synthesis task where each room has length rl and width rw , we measure distance with $f(\mathcal{C}_k, \mathcal{C}_j) = \|rl_k - rl_j\|^2 + \|rw_k - rw_j\|^2$. We select supporting demonstrations with the top- k least distance measures and construct them as exemplars following the CSS structure in Figure 6.2. These supporting examples are provided to GPT-3.5/4 in reverse order, with the most similar example presented last.

6.3.4 Image and Scene Generation

For text-conditioned image synthesis, we utilize a layout-to-image generation model to generate images based on the generated layouts. As for each object layout in 3D scene synthesis, we retrieve a 3D object based on the predicted category, location, orientation, and size following Paschalidou et al. [187]. We directly render the scene with the retrieved 3D objects. See Sec. 6.4 & Sec. 6.5 for more details.

6.4 LayoutGPT for Text-Conditioned Image Synthesis

In this section, we provide an extensive evaluation of LayoutGPT for 2D text-to-image (T2I) synthesis and compare it with SOTA T2I models/systems. An ablation study is conducted to demonstrate the effect of individual components from LayoutGPT. We also showcase qualitative results and application scenarios of our method.

6.4.1 Experiment Setup

6.4.1.1 Datasets & Benchmarks

To evaluate the generations in terms of specified counts and spatial locations, we propose NSR-1K, a benchmark that includes template-based and human-written (natural) prompts from MSCOCO [117]. Table 6.1 summarizes our dataset statistics with examples. For template-based prompts, we apply a set of filters to obtain images with only 1-2 types of object and then create prompts based on object categories and bounding box information. As for natural prompts, we extract COCO captions with keywords to suit the task of numerical reasoning (e.g. “four”) or spatial reasoning (e.g. “on top of”) and

Task	Type	Example Prompt	# Train	# Val	# Test
T2I Numerical Reasoning	Single Category	“There are <i>two giraffes</i> in the photo.”	14890	-	114
	Two Categories	“ <i>Three potted plants</i> with <i>one vase</i> in the picture.”	7402	-	197
	Comparison	“A picture of <i>three cars</i> with a few fire hydrants, the number of cars is <i>more than that</i> of fire hydrants.”	7402	-	100
T2I Spatial Reasoning	Natural	“A fenced in pasture with <i>four horses</i> standing around eating grass.”	9004	-	351
	Two Categories	“A dog <i>to the right of</i> a bench.”	360	-	199
	Natural	“A black cat laying <i>on top of</i> a bed <i>next to</i> pillows.”	378	-	84

Table 6.1: Dataset statistics and examples of the NSR-1K benchmark for image layout planning and text-to-image (T2I) generation with an emphasis on numerical and spatial reasoning.

ensure that all objects from the bounding box annotations are mentioned in the caption to avoid hallucination. Each prompt from NSR-1K is guaranteed to have a corresponding ground truth image and layout annotations.

We rely on the MSCOCO annotations to create NSR-1K with ground-truth layout annotations. Note that each image in COCO is paired with a set of captions and a set of bounding box annotations.

Numerical Reasoning We primarily focus on the competence of T2I models to count accurately, i.e., generate the correct number of objects as indicated in the input text prompt. The prompts for this evaluation encompass object counts ranging from 1 to 5. To design the template-based T2I prompts, we initially sample possible object combinations within an image based on the bounding box annotations. We only use the bounding box annotation of an image when there are at most two types of objects within the image. As a result, the template-based prompts consist of three distinct types: (1) *Single Category*, wherein the prompt references only one category of objects in varying numbers; (2) *Two Categories*, wherein the prompt references two categories of distinct objects in varying numbers; and (3) *Comparison*, wherein the prompt references two categories of distinct objects but specifies the number of only one type of object, while the number of the other type is indicated indirectly through comparison terms including

“fewer than”, “equal number of”, and “more than”. As for natural prompts, we select COCO captions containing one of the numerical keywords from “one” to “five” and filter out those with bounding box categories that are not mentioned to avoid hallucination.

Spatial Reasoning We challenge LLMs with prompts that describe the positional relations of two or more objects. Our spatial reasoning prompts consist of template-based prompts and natural prompts from COCO. To construct template-based prompts, we first extract images with only two ground-truth bounding boxes that belong to two different categories. Following the definitions from PaintSkill [201], we ensure the spatial relation of the two boxes belong to (**left**, **right**, **above**, **below**). Specifically, given two objects A, B , their bounding box centers $(x_A, y_A), (x_B, y_B)$ and the Euclidean distance d between two centers, we define their spatial relation $\text{Rel}(A, B)$ as:

$$\text{Rel}(A, B) = \begin{cases} B \text{ above } A & \text{if } \frac{y_B - y_A}{d} \geq \sin(\pi/4) \\ B \text{ below } A & \text{if } \frac{y_B - y_A}{d} \leq \sin(-\pi/4) \\ B \text{ on the left of } A & \text{if } \frac{x_B - x_A}{d} < \cos(3\pi/4) \\ B \text{ on the right of } A & \text{if } \frac{x_B - x_A}{d} > \cos(\pi/4) \end{cases} \quad (6.1)$$

The definition basically dissects a circle centered at A equally into four sectors that each represent a spatial relation. While the definition may not stand for all camera viewpoints, it allows us to mainly focus on the **front view** of the scene. Then, we utilize the category labels and the pre-defined relations to form a prompt, as is shown in Table 6.1. As for the natural COCO prompts, we select prompts that contain one of the key phrases (**the left/right of**, **on top of**, **under/below**) and ensure that the bounding box annotations align with our definition.

6.4.1.2 Evaluation Metrics

To evaluate generated layouts, we report precision, recall, and accuracy based on generated bounding box counts and spatial positions [202, 203]. For spatial reasoning, each prompt falls into one of the four types of relations ($\{left, right, top, below\}$) and we use the bounding box center for evaluation following PaintSkills [201]. To evaluate generated images, we first obtain bounding boxes from GLIP [204] detection results and then compute average accuracy based on the bounding box counts or spatial relations. We also report CLIP cosine similarity between text prompts and generated images for reference.

We denote the set of n object categories in the ground truth annotation as $\mathcal{C}_{GT} = c_1, c_2, \dots, c_n$, where $x_{c_1}, x_{c_2}, \dots, x_{c_n}$ represent the number of objects for each category. Additionally, we denote the set of m object categories mentioned in GPT- k 's layout prediction as $\mathcal{C}_{pred} = c'_1, c'_2, \dots, c'_m$, where $x'_{c'_1}, x'_{c'_2}, \dots, x'_{c'_m}$ represent the number of objects for each category accordingly. If a category c_i is not mentioned in \mathcal{C}_{pred} , then x'_{c_i} is assigned a value of 0, and vice versa.

The numerical reasoning ability of GPT- k on layout planning is assessed using the following metrics: (1) *precision*: calculated as $\frac{\sum_{k=1}^n \min(x_{c_k}, x'_{c_k})}{\sum_{k=1}^m x'_{c_k}}$, is an indication of the percentage of predicted objects that exist in the groundtruth; (2) *recall*: calculated as $\frac{\sum_{k=1}^n \min(x_{c_k}, x'_{c_k})}{\sum_{k=1}^n x_{c_k}}$, indicates the percentage of ground-truth objects that are covered in the prediction; (3) *accuracy*: In the ‘‘comparison’’ subtask, an accuracy score of 1 is achieved when the predicted relation, whether it is an inequality or equality, between the two objects is accurately determined. For all other numerical subtasks, accuracy equals to 1 if the predicted categories and object numbers precisely match the ground truth. In other cases, the accuracy is 0. Figure 6.3 shows an example of how we compute the *precision* and *recall*. The *accuracy* for this single example is 0 since the predicted object

Categories	c_i	cat	bed	pillow
Ground Truth	x_{c_i}	2	1	2
Prediction	x'_{c_i}	1	0	3

$$precision = \frac{\sum \min(x_{c_i}, x'_{c_i})}{\sum x'_{c_i}} = \frac{1+0+2}{1+0+3} = 75\%$$

$$recall = \frac{\sum \min(x_{c_i}, x'_{c_i})}{\sum x_{c_i}} = \frac{1+0+2}{2+1+2} = 60\%$$

Figure 6.3: An closeup example of how we compute the layout automatic evaluation metrics for numerical reasoning.

distribution does not match the ground truth in every category.

For spatial reasoning, we evaluate spatial accuracy based on the LLM-generated layouts and GLIP-based layouts. We adopt GLIP [204] finetuned on COCO to detect involved objects from the generated images and obtain the bounding boxes. For both types of layouts, we categorize the spatial relation based on the above definition and compute the percentage of predicted layouts with the correct spatial relation. For all evaluation benchmarks, we measure the CLIP similarity, which is the cosine similarity between the generated image feature and the corresponding prompt feature.

6.4.1.3 Baselines

As we consider both layout evaluation and image evaluation, we compare Layout-GPT with **end-to-end T2I models** (Stable Diffusion [4], Attend-and-Excite [184])¹ and **two-stage systems** that generate layouts first and then apply GLIGEN [170] as the layout-to-image model. We also evaluate ground truth layouts and human-drawn layouts as the theoretical upper bounds. The human-drawn layouts are collected through crowdsourcing, in which we specifically ask human annotators to draw layouts given text prompts. We slightly modify LayoutTransformer [166] as a baseline for supervised conditional layout generation.

¹Attend-and-Excite uses Stable Diffusion (SD) as the generative backbone. For both end-to-end T2I models, we report results on SD v1.4 and SD v2.1.

Methods	Numerical Reasoning					Spatial Reasoning			
	Layout Eval.			Image Eval.		Layout Eval.		Image Eval.	
	Precision	Recall	Accuracy	Acc. (GLIP)	CLIP Sim.	Accuracy	Acc. (GLIP)	CLIP Sim.	
<i>Text \rightarrow Image</i>									
1	Stable Diffusion (v1.4) [4]	-	-	-	32.22	0.256	-	16.89	0.252
2	Stable Diffusion (v2.1)	-	-	-	42.44	0.256	-	17.81	0.256
3	Attend-and-Excite (SD v1.4) [184]	-	-	-	38.96	0.258	-	24.38	0.263
4	Attend-and-Excite (SD v2.1)	-	-	-	45.74	0.254	-	26.86	0.264
<i>Text \rightarrow Layout \rightarrow Image</i>									
5	LayoutTransformer [166]	75.70	61.69	22.26	40.55	0.247	6.36	28.13	0.241
6	LayoutGPT (GPT-3.5)	94.81	96.49	86.33	51.20	0.258	82.54	52.86	0.264
7	LayoutGPT (Codex)	90.19	88.29	72.02	46.64	0.254	74.63	45.58	0.262
8	LayoutGPT (GPT-3.5, chat)	81.84	85.47	75.51	54.40	0.261	85.87	56.75	0.268
9	LayoutGPT (GPT-4)	78.36	86.29	78.43	55.64	0.261	91.73	60.64	0.268
10	GT layouts	100.00	100.00	100.00	53.23	0.256	100.00	62.54	0.261
11	Human	99.26	96.52	92.56	56.07	0.258	91.17	51.94	0.258

Table 6.2: Comparison of our LayoutGPT with baseline methods in terms of counting and spatial correctness. Line 5-11 generates layout and adopts GLIGEN [170] for layout-guided image generation. “Human” (line 11) denotes layouts collected from human users given text prompts. Text in bold shows the best results of LayoutGPT.

6.4.2 Evaluation Results

Quantitative Results As shown in Table 6.2, among the variants of LayoutGPT (#6-#9), GPT-3.5 achieves the best performance in numerical reasoning while GPT-4 performs the best in generating correct spatial positions. LayoutGPT outperforms LayoutTransformer (#5) by large margins, proving the strong cross-modal reasoning skills of LLMs. As for image-level evaluation, LayoutGPT surpasses end-to-end T2I models (#1-#3) by 20-40% in GLIP-based accuracy and relatively 1-6% in CLIP similarity. Therefore, using layouts as an intermediate representation indeed leads to more reliable and faithful generation outcomes. In addition, LayoutGPT achieves similar layout accuracy as human users (numerical #6 vs. #11 (86.33% v.s. 92.56%); spatial #9 vs. #11 (91.73% v.s. 91.17%)), which implies its potential to spare users from drawing layouts manually. The discrepancy between layout accuracy and GLIP-based accuracy suggests that the bottleneck mainly stems from layout-guided image generation and GLIP grounding results.

In addition, LayoutGPT binds attributes to each object’s bounding box with 100%

Models	Attribute binding Accuracy (%)			
	Prompts w/ 2 objects	Prompts w/ 3 objects	Prompts w/ 4 objects	Overall
SD1.4	18.57	10.10	11.36	12.84
Attend-and-Excite	31.43	19.19	20.45	22.96
LayoutGPT + GLIGEN	22.86	19.19	14.77	18.68
LayoutGPT + ReCo [171]	40.00	37.37	34.09	36.96

Table 6.3: Color binding accuracy evaluated on prompts from HRS-Bench [183]. We follow the benchmark and use a hue-based classifier to identify the color of generated objects.

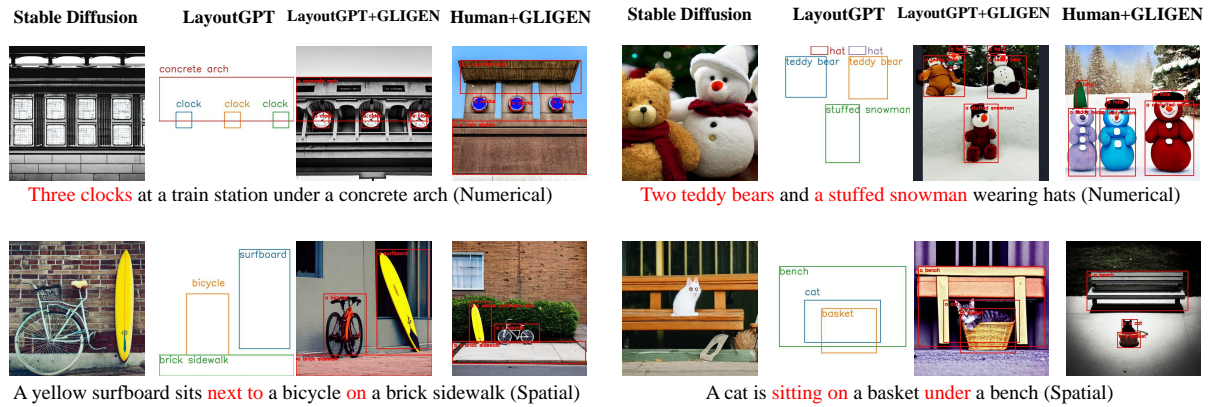


Figure 6.4: Qualitative comparison between Stable Diffusion, LayoutGPT, and human annotations regarding numerical (top row) and spatial reasoning (bottom row) skills.

accuracy on HRS [183] color prompts. We further evaluate the attribute correctness rate (accuracy) on the final generated images when combining LayoutGPT with GLIGEN/ReCo. As shown in Table 6.3, our system largely improves the color correctness over Stable Diffusion with multiple objects.

Qualitative results We show the qualitative results of LayoutGPT and baselines in Figure 6.4. LayoutGPT can understand visual commonsense such as the clock sizes at a train station (top left) or complex spatial relations between multiple objects (bottom right), while SD fails to generate correct numbers or positions. Besides, LayoutGPT demonstrates a similar layout design to human users (bottom left).

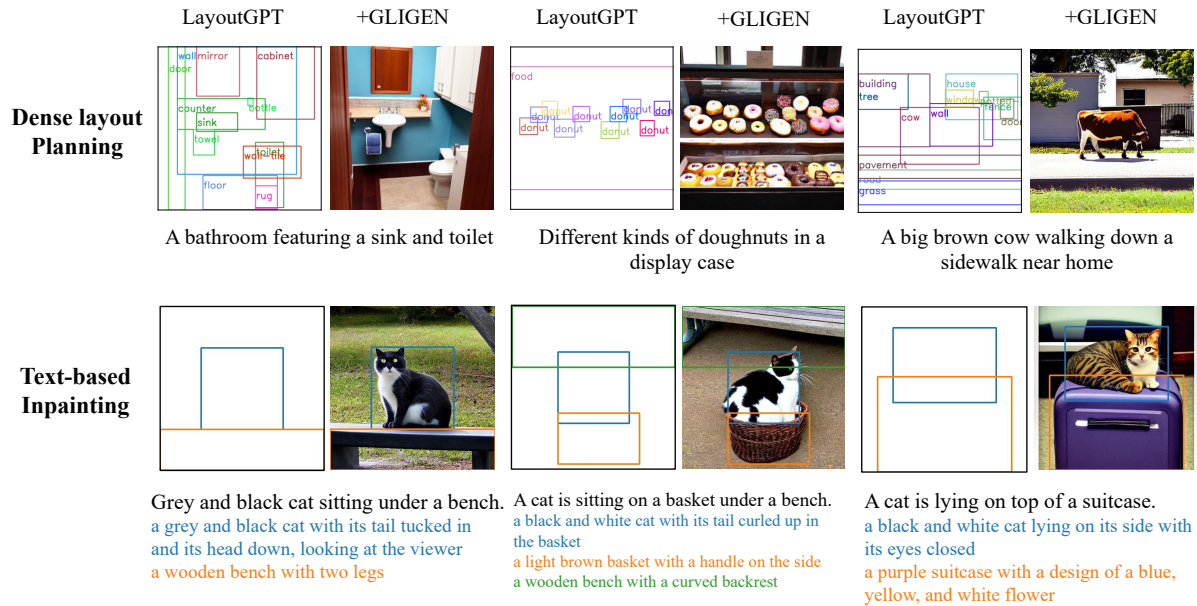


Figure 6.5: **Dense layout planning**: LayoutGPT can generate rich objects or categories in complex scenes for MSCOCO 2017 Panoptic prompts [117]. **Text-based inpainting**: LayoutGPT can generate free-form regional descriptions that are not mentioned in the global prompt.

6.4.3 Application Scenarios

By utilizing LLMs as layout generators, LayoutGPT can be applied to a diverse set of scenarios for accurate and creative image generation.

Dense Layout Planning In Figure 6.5 (top), we apply random in-context examples from COCO17 panoptic annotations with 6~15 bounding boxes per image. LayoutGPT can be applied to scenarios that imply numerous objects (e.g. different kinds of donuts) or various categories (e.g. bathroom or street view). Though only a few objects are mentioned in the prompts, LayoutGPT predicts layouts for the whole scene and imagines common objects that are usually visible in each scene.

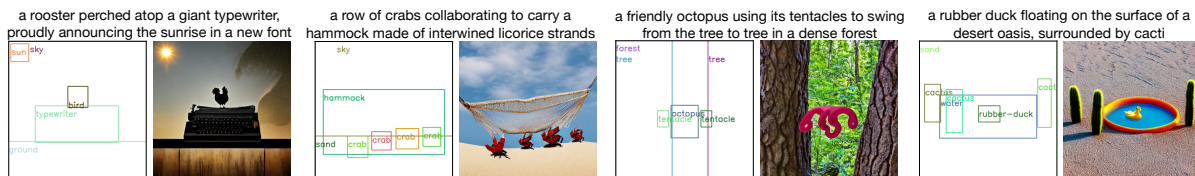


Figure 6.6: Qualitative examples of LayoutGPT’s performance on counterfactual prompts.

Text-based Inpainting In addition, the inherent language generation ability of LLMs enables our method to generate fine-grained regional descriptions from coarse global prompts (Figure 6.5 bottom). LayoutGPT can enrich the description of each object with details that are not mentioned in the prompt, producing suitable outputs for models like ReCo [171].

Counterfactual Scenarios We test LayoutGPT on counterfactual prompts provided by GPT-4 [205]. The in-context examples are randomly drawn from MSCOCO 2017[117], which greatly differs from the counterfactual prompts. As shown in Figure 6.6, LayoutGPT manages to generate reasonable layouts on these challenging prompts and handles the relationship between objects well.

6.4.4 Ablation Study

Component Analysis Table 6.4 presents the component analysis of our CSS-style prompt on spatial reasoning prompts. Comparisons between line 1–3 entails that the task instructions (#2) and CSS format (#3) effectively improve layout accuracy. Format in-context exemplars in CSS structures show a more significant effect on accuracy. Pairwise comparisons of line 5–7 support the argument that the CSS style is the most essential component. While solely applying normalization degrades accuracy in line 4, line 5&8 shows that it slightly improves the performance when combined with other components.

	w/ Instr.	w/ CSS	w/ Norm.	Layout-to-Image Model	Layout Eval	Image Eval	
					Acc.	Acc. (GLIP)	CLIP Sim
1				GLIGEN [170]	55.12	34.35	0.259
2	✓				78.23	47.92	0.263
3		✓			80.82	51.38	0.264
4			✓		44.10	26.43	0.257
5	✓	✓			81.84	52.08	0.264
6	✓		✓		73.36	44.88	0.262
7		✓	✓		76.61	47.56	0.263
8	✓	✓	✓		82.54	52.86	0.264
9	✓	✓	✓	Layout-Guidance [185]	82.54	31.02	0.258
10		GT layouts			100.00	33.92	0.257

Table 6.4: Ablation study of LayoutGPT (GPT-3.5) on spatial reasoning prompts. “w/ Instr.”: with prepended task instructions. “w/ CSS”: format in-context demonstrations in CSS style. “w/ Norm.”: normalizing attribute values to integers by a fixed size.

Model-Agnostic Property We show that LayoutGPT is agnostic to layout-guided image generation models in line 9–10 in Table 6.4. We feed the same generated layouts from LayoutGPT to Layout-Guidance [185] and compute image-level metrics. Compared to using ground truth layouts (#10), LayoutGPT (#9) shows a minor gap in GLIP-based accuracy and a comparable CLIP similarity score. The discrepancy in GLIP-based accuracy is similar to that in Table 6.2, implying that the layouts generated by our method are agnostic to the downstream model.

6.5 LayoutGPT for Indoor Scene Synthesis

6.5.1 Task Setup

Datasets & Benchmarks For indoor scene synthesis, we use an updated version of the 3D-FRONT dataset [206, 207] following ATISS [187]. After applying the same pre-processing operations, we end up with 4273 bedroom scenes and 841 scenes for the living room. We only use rectangular floor plans of the test set for evaluation since LayoutGPT is not compatible with irregular ones. Hence, we end up with 3397/453/423

Models	Bedrooms			Living Rooms		
	Out of bounds (\downarrow)	KL Div. (\downarrow)	FID (\downarrow)	Out of bounds (\downarrow)	KL Div. (\downarrow)	FID (\downarrow)
Random Scenes	11.16	0.0142	23.76	9.43	0.1239	79.61
ATISS*[166]	49.88	0.0113	30.02	83.02	0.1054	85.40
LayoutGPT (GPT-3.5)	43.26	0.0995	28.37	73.58	0.1405	76.34
LayoutGPT (GPT-3.5, chat)	57.21	0.0846	29.66	81.13	0.2077	89.40
LayoutGPT (GPT-4)	51.06	0.1417	29.88	64.15	0.1613	78.60

Table 6.5: Comparison of LayoutGPT with ATISS on indoor scene synthesis. “Random Scenes” means randomly sampling one training scene from the in-context demonstrations for each inference room sample. (* denotes results reproduced by us)

for train/val/test split of bedroom scenes and 690/98/53 for train/val/test split of living room scenes.

Evaluation Metrics We follow prior work [187] to report KL divergence between the furniture category distributions of predicted and ground truth scenes. We also render scene images from four camera angles for each scene and report FID scores [208]. In addition, we report out-of-bound rates, i.e. the percentage of scenes with furniture exceeding the floor plan boundary.

6.5.2 Evaluation Results

Quantitative Results The evaluation results are recorded in Table 6.5. We provide a random baseline for comparison denoted as “Random Scenes”, in which the scene is randomly sampled from the in-context exemplars for each inference run.²

For both bedrooms and living rooms planning, LayoutGPT attains lower out-of-bound rates than ATISS (bedrooms: 43.26% vs. 49.88%; living rooms: 64.16% vs. 83.02%), which verifies LayoutGPT’s spatial reasoning ability in 3D environments. In addition, LayoutGPT has lower FID compared to ATISS (bedrooms: 28.37 vs. 30.02; living rooms:

²Notice that while the scenes in “Random Scenes” are sampled from the training set, the out-of-boundary rate is larger than 0 since the 3D-FRONT dataset contains a small portion of scenes with out-of-bound furniture.

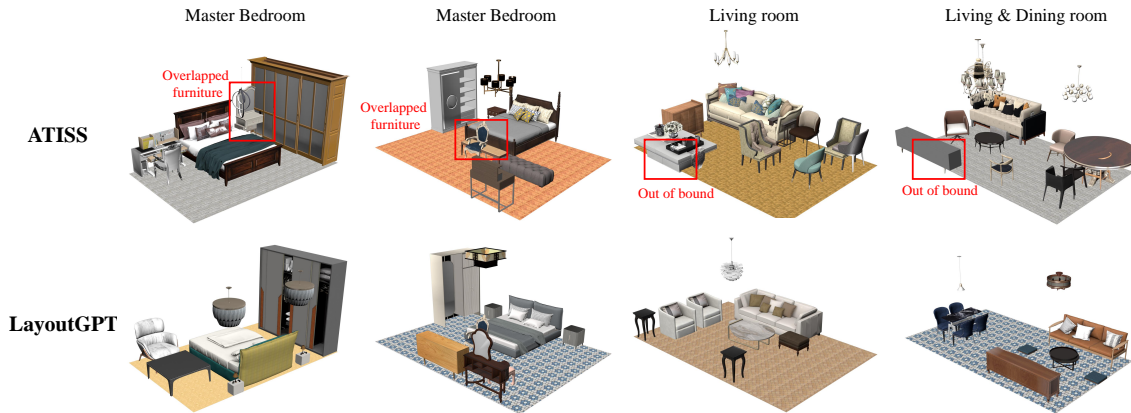


Figure 6.7: Visualization of LayoutGPT across different types of rooms with different floor plan sizes.

76.34 vs. 85.40), which indicates that the planned scene has higher quality. Noted here that the living room split contains much more objects on average (11 for living rooms vs. 5 in bedrooms) and is a low-resource split with only 690 training scenes. Therefore, while living rooms are challenging for both methods, LayoutGPT shows more significant improvement over ATISS as supervised methods tend to overfit in early epochs.

Meanwhile, ATISS performs better in terms of KL divergence, which means that the overall furniture distribution predicted by ATISS is closer to the test split. We observe that LayoutGPT tends to avoid furnitures that are extremely rarely seen in each scene (e.g. coffee tables for bedrooms) as these objects appear less frequently in the in-context demonstrations. The limited in-context demonstration size also restricts LayoutGPT to have a universal observation of the furniture distributions.

Qualitative Results As shown in Figure 6.7, LayoutGPT manages to understand common 3D concepts, such as “the pendant lamp should be suspended from the ceiling” and “nightstands should be placed by the headboard of the bed” (bottom row). When given a floor plan size for both living and dining rooms, LayoutGPT can also generate complicated 3D planning with dining tables and chairs on one side, and a sofa, a coffee

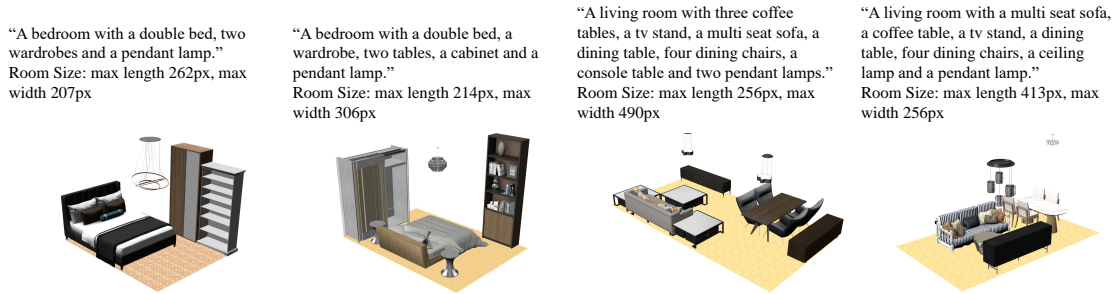


Figure 6.8: Generation of 3D scenes based on text captions that enumerate the furniture.

table, and a TV stand on the other side (bottom right).

6.5.3 Application Scenarios

Text-guided Synthesis : LayoutGPT can follow text captions to arrange furniture in the scene (see Figure 6.8). When the captions enumerate a complete list of furniture, LayoutGPT strictly follows the captions to generate the furniture and achieve a KL Div. value close to zero.

Partial Scene Completion : Thanks to the autoregressive decoding mechanism, LayoutGPT can complete a scene with partial arrangements such that the additional furniture remains coherent with the existing ones. Through in-context demonstrations, LayoutGPT learns critical (visual) commonsense such as visual symmetric (e.g. nightstands in Figure 6.9 (a)), positional relations (e.g. stool at the end of the bed in Figure 6.9 (b)), and room functions (e.g. desks and chairs in the dining area in Figure 6.9 (d)).

6.5.4 Ablation Study

Similar to Sec. 6.4.4, we study the effect of task instructions, CSS structure, and normalization on indoor scene synthesis (see Table 6.6). In contrast to our conclusion for 2D planning in Sec. 6.4.4, comparisons between line 1-4 show that normalization

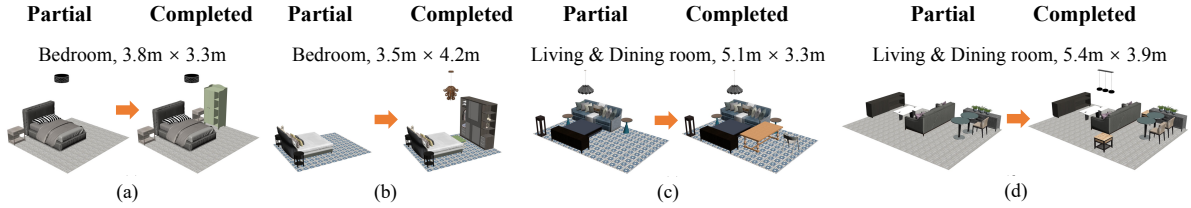


Figure 6.9: LayoutGPT can successfully complete a partial scene for different rooms. We provide three starting objects for bedrooms and seven objects for living rooms.

	w/ Instr.	w/ CSS	w/ Norm.	Out of Bound ↓	KL Div. ↓	FID ↓
1				55.32	0.1070	56.83
2	✓			54.85	0.1153	58.85
3		✓		51.77	0.0776	55.62
4			✓	46.57	0.1276	58.24
5	✓	✓		51.30	0.0741	57.64
6	✓		✓	46.81	0.0913	58.61
7		✓	✓	43.74	0.0848	57.70
8	✓	✓	✓	43.26	0.0995	56.66

Table 6.6: Ablation studies on LayoutGPT on the bedroom split for 3D indoor scene synthesis.

(#4) is the most critical component for suppressing the out-of-bound rate while the CSS structure is also effective. We observe that LLMs occasionally copy attribute values directly from in-context exemplars even though the room sizes are different. Therefore, normalizing all exemplars to the same scale can reduce the out-of-bound rate. CSS style facilitates LLMs to understand the physical meaning behind each attribute value and hence leads to almost the best result when combined with normalization (#7).

Chapter 7

Verbalization Embodiment of LLM Agents for Vision and Language Navigation

7.1 Introduction

Large language models (LLMs), which have shown impressive reasoning capabilities in traditional natural language processing tasks, are increasingly used as the reasoning engine of embodied agents for, e.g., household robots [209], video games [210] and indoor navigation [211]. These tasks are mostly based on simulations that either feature computer-generated images with a fixed set of displayable objects and textures, or are limited in scale and trajectory length. In this paper, we present a verbalization embodiment of an LLM agent (VELMA) for urban vision and language navigation in Street View. The unique challenge of this task is the combination of a large-scale environment derived from an actual road network, real-world panorama images with dense street scenes, and long navigation trajectories. The agent needs to ground its understanding of

```
Navigate to the described target location!
Action Space: forward, left, right, turn_around, stop
Navigation Instructions:
"Oriente yourself such that a blue bench is on your right, go to the
end of the block and make a right. Follow the park on your left and
make a right at the intersection. Pass the black fire hydrant on your
right and stop when you get to a gray door on the brown building."
Action Sequence:
There is a blue bench on your left.
1. turn_around
There is a blue bench on your right.
2. forward
There is a 3-way intersection.
3. right
4. forward
There is a park on your left.
5. forward
There is a park on your left.
6. forward
There is a 4-way intersection.
7. <next word prediction>
```

Figure 7.1: Prompt sequence used to utilize LLMs for VLN in Street View. Verbalized observations of the visual environment are in green and appended to the prompt at each step. Agent actions (blue) are acquired by LLM next word prediction. Highlighting of text for visual presentation only. Full navigation trajectories are, on average, 40 steps long.

the navigation instructions in the observable environment and reason about the next action to reach the target location. The navigation instructions are written by humans and include open-ended landmark references and directional indications intended to guide the agent along the desired path. In order to leverage the reasoning capabilities of LLMs, we use embodiment by verbalization, a workflow where the task, including the agent’s trajectory and visual observations of the environment, is verbalized, thus embodying the LLM via natural language. Figure 7.1 shows the verbalization at step 7 of the current trajectory for a given navigation instance. At each step, the LLM is prompted with the current text sequence in order to predict the next action. Then the predicted action is executed in the environment, and the new observations are verbalized and appended to the prompt. This is repeated until the agent eventually predicts to stop.

The main contributions of our work are as follows: (i) We introduce VELMA, to our

knowledge, the first LLM-based agent for urban VLN. (ii) We report few-shot results for the urban VLN task and achieve new state-of-the-art performance by finetuning our agent on the training set. (iii) We address and resolve limitations of the commonly used Touchdown environment [212], making it amenable for few-shot agents.

7.2 Related Work

Outdoor VLN Agent models for the outdoor/urban VLN task [212] commonly follow a sequence-to-sequence architecture where encoded text and image representations are fused for each decoder step [213, 214, 215, 216]. Other proposed agents employ pretrained vision and language transformers that are finetuned on task-specific data [217, 218]. Zhong et al. [219] represent the visual environment by symbols using semantic segmentation and extreme downsampling of panorama images, but their agent does not improve over previous success rates. Other work uses CLIP to score the presence of extracted landmarks at each panorama node in a graph and uses this information to plan a route for given navigation instructions [220]. Their non-urban environment has a graph with 300 nodes, and the navigation path is planned a priori with full access to all panorama images and landmark scores. In contrast, our agent is embodied and has to plan ad-hoc with access to directly observed information only.

Indoor VLN Indoor agents [221, 222, 223, 224, 225, 226, 227, 228, 229] are used for navigation datasets like R2R [230] and RxR [231] or ObjectNav [232, 233]. Khandelwal et al. [234] showed that using the CLIP encoder for image features improves performance for a range of vision and language tasks. Recently, Zhou et al. [211] introduced an LLM-based agent for R2R that incorporates image information by transcribing its entire content with an image-to-text model. This is feasible because the navigation trajectories

are only six steps on average compared to 40 steps in the urban VLN task considered in our work. Another notable indoor VLN agent uses CLIP to directly predict the next action by scoring the compatibility of the current sub-instruction with available waypoint images [235].

7.3 Urban VLN Environment

We use the Touchdown environment introduced by Chen et al. [212]. The environment is based on Google’s Street View and features 29,641 full-circle panorama images connected by a navigation graph. It covers the dense urban street network spanning lower Manhattan. The navigation graph is a directed graph $G = \langle V, E \rangle$ where each edge $\langle v, v' \rangle \in E$ is associated with $\alpha_{\langle v, v' \rangle}$ which is the heading direction from node v to node v' ranging from 0° to 360° . The agent state $s = (v, \alpha)$ is composed of its current position $v \in V$ and its heading direction α . The agent can move by executing an action $a \in \{\text{FORWARD}, \text{LEFT}, \text{RIGHT}, \text{STOP}\}$. The state transition function $s_{t+1} = \phi(a_t, s_t)$ defines the behavior of the agent executing an action. In Touchdown [212], the agent’s heading α_t at position v is restricted to align with the heading of an outgoing edge $\alpha_{\langle v, v' \rangle}$. In case of the **RIGHT** action, the new state s_{t+1} is $(v, \alpha_{\langle v, \mathbf{v}v \rangle})$ where $\mathbf{v}v$ is the neighboring node closest to the right of the agent’s current heading. In other words, the agent is rotated in place to the right until it *snaps* to the direction of an outgoing edge. Likewise, for the **LEFT** action. In the case of the **FORWARD** action, the agent moves along the edge $\langle v, v' \rangle$ according to its current heading direction $\alpha_t = \alpha_{\langle v, v' \rangle}$. The environment is then forced to automatically rotate the agent’s heading towards an outgoing edge: $\alpha_{t+1} = \alpha_{\langle v', v^* \rangle}$ where v^* is the neighbor node in the direction closest to the previous heading α_t .



Figure 7.2: The Touchdown environment introduced by Chen et al. [212] can require action sequences that are semantically inconsistent with the correct navigation instructions. In the depicted subgraph, the action sequence to move from node 1 to node 5 is to move **FORWARD** four times. The semantically correct sequence of actions would include a right turn in between. We fix the problem by modifying the environment behavior and selecting the desired direction at intersections in relation to all outgoing streets.

7.3.1 Alignment Inconsistencies in Touchdown

As described in Schumann et al. [214], the automatic rotation mentioned above can lead to generalization problems, e.g., when moving towards the flat side of a T-intersection. For example, if the agent is automatically rotated towards the right facing street and subsequently executes the **RIGHT** action, it rotates towards the direction it came from instead of clearing the intersection in the intended direction. The same problem also occurs at intersections with more than three directions. Figure 7.2 gives an illustrative example that shows the navigation graph at a 4-way intersection. Because the environment is derived from a real-world street layout, the nodes in the graph are not perfectly arranged as in an artificial grid world. In order to make a right turn at the intersection and to follow the route from v^1 to v^5 , one expects to use the action sequence `[FORWARD, FORWARD, RIGHT, FORWARD, FORWARD]`. However, when the agent reaches v^3 , it is automatically rotated towards the closest outgoing edge, in this case, $\langle v^3, v^4 \rangle$. This is because the rotation $20^\circ \rightarrow 50^\circ$ towards v_4 is shorter than the rotation $20^\circ \rightarrow 345^\circ$ towards

v_7 . As such, the required sequence of actions to go from v^1 to v^5 in Touchdown’s [212] environment is [FORWARD, FORWARD, FORWARD, FORWARD]. This is unpredictable and is not correctly aligned with “*turn right at the intersection*” instructions.¹ To alleviate this problem, Schumann et al. [214] explicitly feed the change of heading at each timestep as additional input to their model. This enables the agent to anticipate the unexpected rotation and to adapt to it. Because adding heading delta values to the text-based interface makes it convoluted and unnecessarily difficult for few-shot learning, we propose a more intuitive way to solve this ambiguity at intersections. We modify the state transition function ϕ such that the agent is not automatically rotated when moving FORWARD. This means the agent’s heading α_t is not automatically aligned with an outgoing edge. Instead, the direction is selected in relation to all outgoing edges. The agent at node v^3 in Figure 7.2 has the nodes v^6 , v^7 and v^4 in front. The forward direction is selected as the middle one of the three edges, the right direction as the right-most edge, and the left direction as the left-most edge. This means that executing the RIGHT action at position v^3 will now rotate the agent towards node v^4 and allows to use the semantically correct sequence of actions for the depicted route. The proposed modification solves the issue of inconsistent action sequences at intersections and allows to use agents that are not specifically trained in this environment.

7.3.2 Turning Around

We additionally introduce the TURN_AROUND action which lets the agent reverse its direction: $s_{t+1} = (v, \alpha_t - 180^\circ)$. In the unmodified environment, this is achieved using the LEFT or RIGHT action on regular street segments. The new action is better aligned with natural language verbalizations of direction reversal and promotes intuitive communication with the environment.

¹In the Appendix we show more examples for 3-way, 4-way and 5-way intersections.

Egocentric Spatial Reasoning	
1.	... turn so the orange barrier is on your left ...
2.	... a red truck in front of you ...
3.	... a playground on the far right corner ahead ...
Allocentric Spatial Reasoning	
4.	... green metal pole with pink flowers on top ...
5.	... building with columns around the windows ...
6.	... stop in between Chase and Dunkin' Donuts ...
Temporal Reasoning	
7.	... go straight until you see Chipotle and then ...
8.	... once you passed the underpass ...
9.	... stop when the park on your right ends ...
Other	
10.	... proceed straight through three intersections ...
11.	... you should see TD Bank on your left ...
12.	... if you see Dory Oyster Bar, you are too far ...

Table 7.1: Reasoning skills the embodied LLM agent must possess in order to successfully complete the navigation task. Each with three example snippets from the navigation instructions.

7.4 Navigation Task

The objective of the navigation task is to find the goal location by following the given navigation instructions. A navigation instance is defined by the initial state s_1 , target node \hat{v}_T , gold path $(\hat{v}_1, \hat{v}_2, \dots, \hat{v}_T)$ and navigation instructions text $n = (w_1, w_2, \dots, w_N)$. The agent starts at s_1 and predicts the next action a_1 based on the navigation instructions and current observations. These are the panorama image and number of outgoing edges at the current position. The environment processes the action and puts the agent into a new state: $s_2 = \phi(a_1, s_1)$. This is repeated until the agent predicts **STOP** at the presumed goal location. If the agent stops within one neighboring node of the target node, the navigation objective is considered accomplished.

7.4.1 Challenges

One main challenge to successfully follow the navigation instructions is to reliably detect landmarks in the panorama images along the route. The landmarks mentioned in the instructions are open-ended and can refer to any object or structure found in street scenes, including vegetation, building features, vehicle types, street signs, construction utilities, company logos and store names. The agent also needs to possess different types of reasoning, most importantly spatial reasoning to follow general directions, locate landmarks and evaluate stopping conditions. The agent also needs to understand the temporal aspect of the task and reason about the sequence of previous observations and actions. See Table 7.1 for example snippets from the navigation instructions.

7.4.2 Datasets

There are two datasets that provide navigation instructions for the environment described in Section 7.3: **Touchdown** [212] and **Map2seq** [236]. Each dataset includes around 10k navigation instances, and we utilize them in the more challenging *unseen* scenario introduced by Schumann et al. [214]. This means that generalization is crucial because the training routes are located in an area that is geographically separated from the area of development and test routes. The main difference between the two datasets is that Touchdown instructions were written by annotators who followed the route in Street View, while Map2seq instructions were written by annotators that saw a map of the route. The Map2seq navigation instructions were later validated to also be correct in Street View. Another difference is that the initial state in Map2seq orientates the agent towards the correct direction which leads to overall better task completion rates than for Touchdown instances.

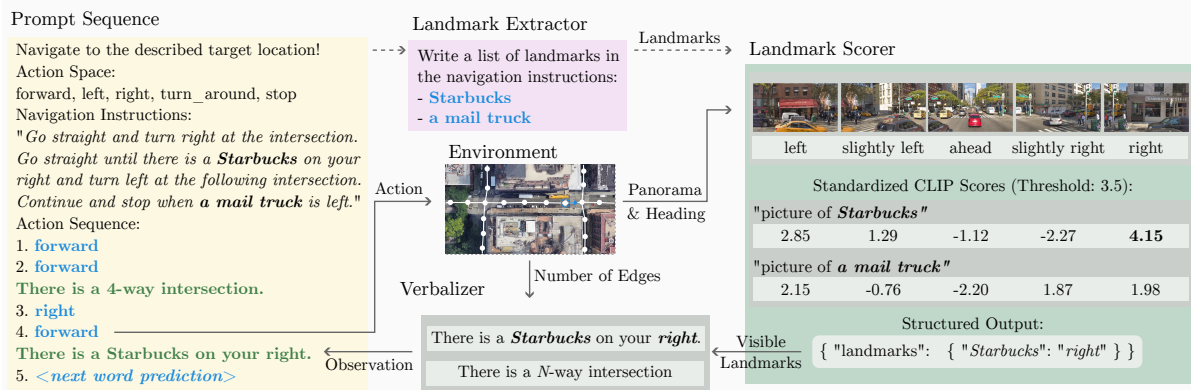


Figure 7.3: Overview of the proposed agent VELMA navigating in the Street View environment. The prompt sequence includes the task description, navigation instructions, and verbalized navigation trajectory up to the current timestep. The next action is decided by next word prediction utilizing an LLM and subsequently executed in the environment. This puts the agent into a new state, and the landmark scorer determines if an extracted landmark is visible in the current panorama view. The verbalizer takes this landmark information along with the information about a potential intersection and produces the current observations text. This text is then appended to the prompt sequence and again used to predict the next action. This process is repeated until the agent stops and the alleged target location.

7.5 LLM Agent

In this section, we propose the urban VLN agent that uses an LLM to reason about the next action. To this end, we verbalize the navigation task, especially the environment observations. The workflow includes the extraction of landmarks that are mentioned in the instructions and determining their visibility in the current panorama image. The verbalizer then integrates the visible landmarks and street intersections into an observation text phrase o_t at each step. The complete text prompt at timestep t is composed as follows:

$$x_t = [d^a, n, d^b, o_1, 1, a_1, o_2, 2, a_2, \dots, o_t, t], \tag{7.1}$$

where $[\]$ denotes string concatenation, d^a and d^b are part of the task description and n is the navigation instructions text. Punctuation and formatting are omitted in the notation

for brevity. Figure 7.3 shows a prompt sequence at $t = 8$ on the left. This formulation of the navigation task enables the agent to predict the next action by next word prediction:

$$a_t = \arg \max_{w \in A} P_{LLM}(w|x_t), \quad (7.2)$$

where A are the literals of the five defined actions and P_{LLM} is a black-box language model with no vision capabilities.

7.5.1 Landmark Extractor

Each navigation instructions text n mentions multiple landmarks for visual guidance. In order to determine if a mentioned landmark is visible in the current panorama view, we first have to extract them from the instructions text. For this, we create a single prompt that includes five in-context examples of navigation instructions paired with a list of landmarks (shown in the Appendix). It is used by the LLM to automatically generate the list of landmarks (l_1, l_2, \dots, l_L) mentioned in the given navigation instructions. The landmark extractor is depicted in the top middle of Figure 7.3 and executed before the navigation starts.

7.5.2 Landmark Scorer

At each step, the agent observes a panorama view p_v^α , defined by its current position v and heading direction α . The view is an 800x460 sized image cut from the panorama with 60° field-of-view. In order to determine if a landmark l_i is visible in the view, we employ a CLIP model [23] to embed the image and the caption: “*picture of l_i* ”. The similarity score of the two embeddings determines the visibility of the landmark. Because the scores can be biased towards certain types of landmarks, we standardize them using all views p_{train}^* of the ~20k panorama images in the training area. Recall that we operate

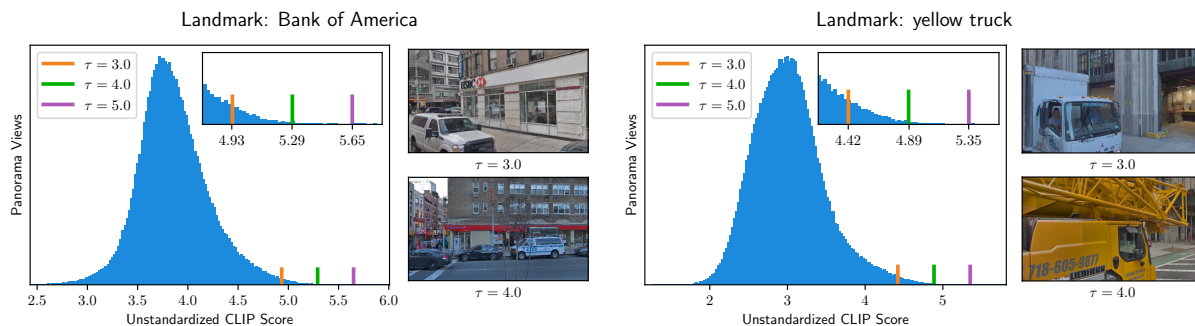


Figure 7.4: Distribution of CLIP scores between a landmark and panorama images in the training area. The CLIP score represents the semantic similarity of the panorama image and the text caption "picture of [landmark]". The distribution is used to standardize the score of the landmark and a novel panorama. The threshold τ is defined on the standardized score and used to determine the visibility of the landmark in the novel panorama image.

in the *unseen scenario* where the training area and evaluation area are geographically separated. The standardized score of a landmark is:

$$z(l, p_v^\alpha) = \frac{\text{CLIP}(l, p_v^\alpha) - \mu(C_l)}{\sigma(C_l)} \quad (7.3)$$

where $C_l = \{\text{CLIP}(l, p_{v'}^{\alpha'}) \mid p_{v'}^{\alpha'} \in p_{train}^*\}$.

If the standardized score is larger than the threshold τ , the landmark is classified as visible in the current view. The process does not require annotations and is completely unsupervised, allowing to score novel landmarks. The threshold is the only tunable parameter in the landmark scorer. Figure 7.4 shows the distribution of unstandardized CLIP scores and views at different threshold values for two example landmarks. While the views at $\tau = 4.0$ both show the correct landmark, the view at $\tau = 3.0$ for "Bank of America" shows an HSBC branch, and for "yellow truck" it shows a white truck. This suggests that the optimal threshold lies between the two values. As depicted on the right in Figure 7.3, the agent also evaluates views to the left and right of the current heading. Each panorama view direction $(p_v^{\alpha-90^\circ}, p_v^{\alpha-45^\circ}, p_v^\alpha, p_v^{\alpha+45^\circ}, p_v^{\alpha+90^\circ})$ is associated

with a string literal m valued *left*, *slightly left*, *ahead*, *slightly right* or *right*, respectively. A visible landmark l_i and the corresponding direction literal m_i are passed to the verbalizer. A full navigation trajectory includes around 200 image views (40 steps and 5 view directions per step) and each landmark is typically visible in only one or two views.

7.5.3 Verbalizer

The verbalizer is a template-based component that produces environment observations in text form. There are two types of environment observations. First, there are street intersections that are detected based on the number of outgoing edges $N(v)$ at the current node v in the navigation graph. If there are three or more outgoing edges at step t , the verbalizer encodes this information into the observation string o_t^e : “*There is a $[N(v)]$ -way intersection*”. Extracting this information directly from the navigation graph is akin to the junction type embedding used by the ORAR model [214] and is motivated by direction arrows displayed in the Street View GUI that human navigators used during data collection. The other type of observations are landmarks visible in the panorama view. The landmark name l_i and direction literal m_i are used to verbalize the observation o_t^l : “*There is $[l_i]$ on your $[m_i]$* ”. The complete observation is $o_t = [o_t^e, o_t^l]$, where the respective string is empty if no intersection or landmark is detected. The observation is appended to the prompt in Eq. (7.1) and used by the agent to decide the next action.

7.6 Experiments

We conducted experiments² to evaluate the navigation performance of the proposed LLM agent in finetuning and in-context learning settings. We used CLIP-ViT-bigG-14-laion2B-39B-b160k [96] as the CLIP model in the landmark scorer. We set the threshold $\tau = 3.5$ for all experiments. The threshold was selected by inspecting the distribution of CLIP scores (as in Figure 7.4) for a handful of landmarks. On purpose, we did not systematically tune it in order to not violate the premise of few-shot learning.

7.6.1 Landmark Extraction

We ran the landmark extractor once for all instances using GPT-3 [237] and used the same extracted landmarks in all experiments. On average, 2.7 landmarks were extracted from a navigation instructions text. Around 58% of the landmarks in the test sets are *novel*, i.e., they are not used in the training instances. In order to estimate the quality of the automatically extracted landmarks, we annotated 50 instances of each development set by hand. For Touchdown we calculated an F1-score of 96.3 (precision: 97.2, recall: 95.4) and the F1-score for Map2seq is 99.6 (precision: 100, recall: 99.3). This shows that GPT-3 reliably extracts landmarks from the instructions text and reusing them for all experiments is minimizing the inaccuracies introduced by this workflow step.

7.6.2 Metrics and Baseline

We use three metrics to measure navigation performance. The task completion (TC) rate is a binary metric that measures whether the agent successfully stopped within one neighboring node of the target location. Shortest-path distance (SPD) calculates the

²Project page: <https://velma.schumann.pub/> and code: <https://github.com/raphael-sch/VELMA>

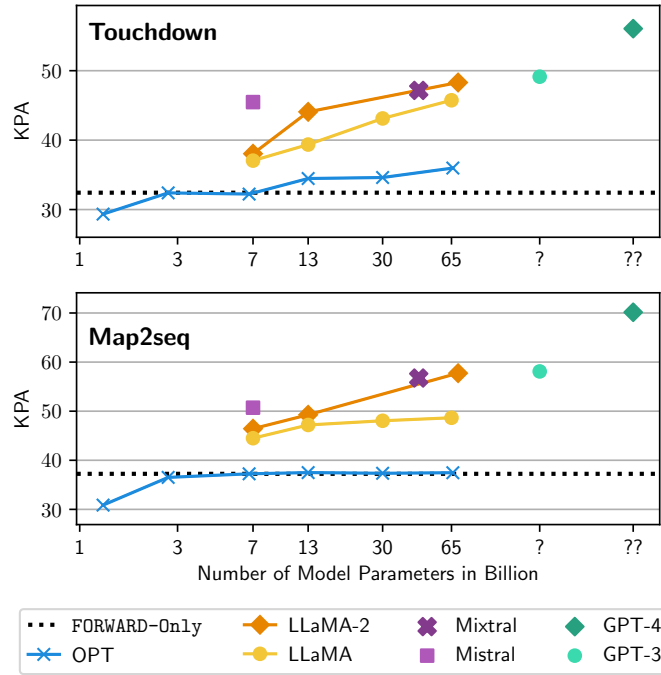


Figure 7.5: Key point accuracy (KPA) for 2-shot in-context learning of large language models with increasing parameter count. The FORWARD-Only baseline predicts walking forward until the average trajectory length is reached and performs better than predicting random directions.

shortest path length between the stopping location and goal location [212]. Key point accuracy (KPA) measures the ratio of correct decisions at key points. Key points include the initial step, intersections along the gold route, and the target location.

For baselines, we use the current state-of-the-art agent model for urban VLN called ORAR [214]. The model employs a seq-to-seq architecture where the encoder LSTM reads the navigation instructions text, and the multi-layer decoder LSTM receives image feature vectors of the current panorama view as additional input at each action decoding step. The ORAR model is a very strong baseline beating more sophisticated models like the VLN Transformer [218]. Because the environment modifications introduced in Section 7.3 spare the agents from learning specific irregularities, we additionally retrain ORAR in the improved environment for a fair comparison.

7.6.3 Few-Shot Learning Results

The proposed text-only interface allows us to use large language models as reasoners without updating their weights or fusing image representations. The prompt consists of a short task description and two in-context examples (2-shot). The examples are full text sequences along the gold route for randomly selected navigation instances in the training set. The two plots in Figure 7.5 show that performance scales with parameter count and varies across model families. The FORWARD-Only baseline reveals that OPT [238] can barely compete with a basic heuristic, even at a model size of 65 billion parameters. LLaMa [239] and especially LLaMa-2 [240] show promising navigation skills reaching 48.3 and 57.7 key point accuracy (KPA) on Touchdown and Map2seq, respectively. However, this KPA score only translates to task completion (TC) rates of 2.1 and 3.2, revealing that the model is not able to consistently predict correct actions throughout the whole navigation trajectory. Mistral-7b performs on par with a LLaMA-2 model twice its size, but also fails to score task completion rates significantly higher than 3. The only few-shot LLMs that achieve substantial TC rates are GPT-3, GPT-4 [205] and Mixtral [241]. As listed in Table 7.2, VELMA-GPT-4 achieves the best results for the 2-shot setting. It reaches 44% and 77% of the TC rate reported for the previous state-of-the-art model ORAR[♠]-ResNet which is a seq-to-seq model that has direct access to image features and was trained on the full training set. In contrast, the LLMs in our work act as a blind agent that solely relies on observation descriptions produced by the verbalizer. This is remarkable because LLMs are not explicitly trained to experience embodiment in a visual environment. This is emergent behavior unearthed by verbalizing the VLN task. We also observe that GPT-4 invokes the TURN_AROUND action in useful ways, e.g. to return a few steps when it notices that it went past the described goal location. This emphasizes the effectiveness of intuitive communication with the environment.

Model	Development Set						Test Set					
	Touchdown			Map2seq			Touchdown			Map2seq		
	SPD↓	KPA↑	TC↑	SPD↓	KPA↑	TC↑	SPD↓	KPA↑	TC↑	SPD↓	KPA↑	TC↑
ORAR-ResNet	20.0	-	15.4±2.2	11.9	-	27.6±1.8	20.8	-	14.9±1.2	13.0	-	30.3±1.8
ORAR [♣] -ResNet	16.5	64.0	22.6±0.6	10.3	74.4	29.9±1.7	17.4	62.3	19.1±1.0	10.9	74.7	32.5±1.4
ORAR [♣] -CLIP	17.5	63.7	21.5±0.9	10.0	75.3	32.8±1.5	17.0	63.4	20.0±0.1	10.5	75.1	34.0±0.5
2-Shot In-Context Learning												
VELMA-Mixtral	28.4	47.2	6.5	21.1	56.8	8.0	-	-	-	-	-	-
VELMA-GPT-3	22.2	49.1	6.8	19.1	58.1	9.2	-	-	-	-	-	-
VELMA-GPT-4	<u>21.8</u>	<u>56.1</u>	<u>10.0</u>	<u>12.8</u>	<u>70.1</u>	<u>23.1</u>	-	-	-	-	-	-
LLM Finetuning, full training set												
VELMA-FT	18.3	62.0	23.4±0.2	8.7	78.7	41.3±0.9	18.2	62.2	23.5±0.4	9.7	78.0	40.0±1.0
VELMA-RBL	15.5	63.6	26.0±0.6	8.3	79.5	45.3±0.5	16.0	62.8	26.4±1.7	8.3	79.6	47.5±0.7

Table 7.2: Results for the urban VLN task on Touchdown and Map2seq in the *unseen* scenario, meaning the training area is geographically separated from the area where development and test routes are located. ORAR-ResNet [214] is the previous best model and follows a seq-to-seq architecture that fuses text and image features during decoding. We retrained this model in our improved environment (ORAR[♣]-ResNet) and with the same OpenCLIP image embeddings (ORAR[♣]-CLIP) that we use in the landmark scorer. VELMA-GPT-3 and VELMA-GPT-4 models employ our proposed verbalization workflow and are prompted with two in-context examples. Due to cost and data leakage concerns, we evaluate the GPT models on the development sets only. VELMA-FT is LLaMa-7b finetuned on all training text sequences (around 6k for each dataset). The VELMA-RBL finetuning process is described in Section 7.6.4.1. All experiments are repeated three times with different random seeds (mean/std reported). Bold values are the nominal best results and underlined are best few-shot results.

7.6.4 Finetuning Results

To further explore the capabilities of the proposed LLM agent, we finetune LLaMa-7b on all training instances of the respective dataset, denoted by VELMA-FT in Table 7.2. Each training instance is the full text sequence that is produced by following the gold path. The visibility of landmarks is determined by the landmark scorer during training because gold annotations are not available. There are 6,770 training instances for Touchdown and 5,737 for Map2seq. We finetune for 20 epochs using LoRA [242] to adapt query, key and value projections of the attention layer as well as input and output projection of each transformer layer. The best model is selected by task completion on the development set. The resulting agent outperforms the previous state-of-the-art model ORAR*

by 10% and 16% relative TC rate. Comparing ORAR* which fuses image features at the vector level to VELMA-FT which finetunes on verbalizations of observations, shows that the text-based environment observations are less prone to overfitting.

7.6.4.1 Response-Based Learning

A navigation task is successfully completed if the agent stops at either the goal location or an adjacent neighboring node. Training the agent with teacher-forcing to exactly follow the gold route penalizes the agent for stopping one step short or one step past the target node, despite accomplishing the navigation objective. Furthermore, the agent can not learn to recover from incorrect decisions during inference. We thus train the agent to directly optimize the TC metric while also feeding it its own actions during training, called VELMA-RBL in Table 7.2. The procedure for VELMA-RBL is inspired by response-based learning [243] and imitation learning [244] and is outlined in Algorithm 1. The loss for an instance at training step j is either computed by teacher forcing the gold action sequence $\hat{\mathbf{a}}$, or by student forcing, determined by a mixing parameter λ . In student forcing, the actions decoded by the current model weights θ_j are executed instead of the gold actions. If this trajectory ends within one neighboring node of the target location, the predicted action sequence \mathbf{a}_j is considered correct and used as the reference to train the agent. If the agent stops at the wrong location, an oracle path is computed to provide the optimal counterfactual action at each step in the trajectory. In our case, the oracle’s optimal next action is computed as the shortest path to the goal location. We set $\lambda = 0.5$ to collect training losses in a batch evenly from both training strategies. Manually inspecting trajectories produced by the trained agent, we found improvements of following instructions that have stopping criteria like “*Stop a few steps before Y.*” or “*Stop at X. If you see Y you have gone too far.*”. In both cases, the agent learned to walk past the uncertain stopping location and to invoke the TURN_AROUND action in order

Algorithm 1 RBL Optimization of Task Completion

Require: mixing ratio λ , training step j , model weights θ_j , gold action sequence $\hat{\mathbf{a}}$, prompt x_1

```

if  $\text{random}(0, 1) < \lambda$  then
   $\mathbf{a}_{\theta_j} = \text{StudentForcing}(\theta_j, x_1)$ 
   $\mathbf{a}_j = \arg \max \mathbf{a}_{\theta_j}$ 
  if  $\text{TaskCompletion}(\mathbf{a}_j) = 1$  then
     $\text{loss}_j = \mathcal{L}_{CE}(\mathbf{a}_{\theta_j}, \mathbf{a}_j)$ 
  else
     $\mathbf{a}_j^* = \text{Oracle}_{\text{stepwise}}(\mathbf{a}_j)$ 
     $\text{loss}_j = \mathcal{L}_{CE}(\mathbf{a}_{\theta_j}, \mathbf{a}_j^*)$ 
  end if
else
   $\mathbf{a}_{\theta_j} = \text{TeacherForcing}(\theta_j, x_1, \hat{\mathbf{a}})$ 
   $\text{loss}_j = \mathcal{L}_{CE}(\mathbf{a}_{\theta_j}, \hat{\mathbf{a}})$ 
end if

```

Model	Touchdown		Map2seq	
	SPD↓	TC↑	SPD↓	TC↑
no image	27.4±0.5	14.7±0.5	9.7±0.2	35.2±0.9
CLIP	21.3±0.5	19.5±0.6	9.8±0.3	37.2±0.5
OpenCLIP	18.6±0.3	22.6±0.4	9.8±0.5	38.2±0.5

Table 7.3: Vision ablation on the development set. We finetune a separate LLaMa-7b model for each ablation. CLIP refers to `clip-vit-large-patch14` [23]. The OpenCLIP image model refers to `CLIP-ViT-bigG-14-laion2B-39B-b160k` [96].

to walk back once landmark Y appeared. The described training procedure leads to a significant increase of task completion rate by 2.9 and 7.5 for Touchdown and Map2seq, respectively. Overall, our contributions in this work amount to a relative increase of task completion by 77% and 57% over the previously reported state-of-the-art for urban VLN on the Touchdown and Map2seq datasets.

7.6.5 Image Ablation

In this section, we ablate the image model used by the landmark scorer. We finetune a LLaMa-7b model according to Section 7.6.4 and use CLIP [23], OpenCLIP [96] or no

image model in the landmark scorer. The latter case means that no landmark observation is passed to the prompt sequence. The results in Table 7.3 show that OpenCLIP is better suited for detecting landmarks in our navigation task than the original CLIP model. This is in line with better ImageNet results reported by the OpenCLIP authors and suggests that the agent can directly benefit from further improvements of CLIP models. Appending no landmarks to the prompt sequence further degrades performance, especially on Touchdown.

Part III

Towards Publicly-Available Vision-and-Language Studies

Chapter 8

Multimodal C4: An Open, Billion-scale Corpus of Images Interleaved With Text

8.1 Introduction

In-context learning [8] enables sequence models to adapt to new tasks without any parameter updates. By interleaving a few supervised examples in a prompt, few-shot learning can be formatted as a next-token prediction task, i.e., $x_1, y_1, x_2, y_2, \dots, x_n$ is input to predict \hat{y}_n . Some image+text models also support in-context learning via interleaving of images/text jointly. Prior experiments [22] suggest that performant multimodal in-context learning is dependent upon pretraining on similarly interleaved sequences of images and text (rather than single image/caption pairs). However, such a large-scale corpus has not been made publicly available.

To address this, we introduce Multimodal C4 (`mmc4`), a public, billion-scale image-

text dataset consisting of interleaved image/text sequences.¹ `mmc4` is constructed from public webpages contained in the cleaned English `c4` corpus. In addition to standard pre-processing steps like deduplication, NSFW removal, etc., we place images into sequences of sentences by treating each document as an instance of a bipartite linear assignment problem, with images being assigned to sentences (under the constraint that each sentence is assigned at most one image). We show that applying CLIP ViT-L/14 [23] to estimate bipartite weights in a zero-shot fashion results in state-of-the-art performance on intra-document alignment benchmarks, and then apply this process to 100M+ documents to construct `mmc4`. Apart from the full corpus, we have created two additional subsets: `mmc4-ff`, which removes images with detected faces, and `mmc4-core`, a more strictly filtered and downsized version of the corpus, serving as an initial corpus for developers.

We explore `mmc4`, showing that: 1) the text and images in the corpus span expected everyday topics like cooking and travel; 2) filters like NSFW/ad removal work with high accuracy; and 3) the resulting images are relevant to the associated documents, and often, appropriately aligned to the most-relevant individual sentence. We conclude by discussing initial use-cases of `mmc4`, including OpenFlamingo [20],² an open source version of Flamingo [22]. Initial ablations show that training on the sequences of `mmc4` enables few-shot, in-context adaptation to image captioning datasets.

8.2 Related Dataset Work

Most million/billion-scale, public multimodal pretraining datasets consist of images paired with their literal descriptions, e.g., LAION-2B [96], CC-12M [246], YFCC100M [247]. However, literal description is only one of many ways images can relate to text

¹`mmc4`'s datasheet [245] is available here.

²https://github.com/mlfoundations/open_flamingo

	# images	# docs	# tokens	Public?
M3W (Flamingo) [22]	185M	43M	-	×
Interleaved training data for CM3 [253]	25M	61M	223B	×
Interleaved training data for KOSMOS-1 [254]	≤ 355M	71M	-	×
Multimodal C4 (<code>mmc4</code>)	571M	101.2M	43B	✓
Multimodal C4 fewer-faces (<code>mmc4-ff</code>)	375M	77.7M	33B	✓
<code>mmc4</code> core (<code>mmc4-core</code>)	29.9M	7.3M	2.4B	✓
<code>mmc4</code> core fewer-faces (<code>mmc4-core-ff</code>)	22.4M	5.5M	1.8B	✓

Table 8.1: Comparison of `mmc4` with other interleaved image/text pretraining corpora. In addition to the full version of the dataset, we also release: 1) fewer-faces subsets, which aim to remove all depicted human faces; and 2) “core” subsets, result from more stringent filtering.

on the web [248]. `mmc4` aims to capture a broader range of these relationship types. Some web datasets collect multiple images for one text snippet (e.g., the Google Local Restaurant Reviews Dataset [249] with 4.4M images), or situate images in longer bodies of text (e.g., the Wikipedia-based Image Text Dataset [250] with 11.5M images), but do not directly cover multi-image/multi-sentence interleaving. Table 8.1 provides summary statistics of other large-scale interleaved pretraining datasets. `mmc4` contains more images than prior non-public datasets. Birhane et al. [251] highlight risks associated with web-scale multimodal data.

In addition to the detailed curation steps described in Section 8.3 and the considerations for data release outlined in Section 8.3.1, we are hopeful that the availability of `mmc4` can facilitate a more transparent and critical examination of interleaved corpora compared to previous privately held training sets. Models trained on `mmc4` inherit its risks; we selected the widely-adopted `c4` corpus as a starting point in part because there are existing auditing efforts on the text-only corpus, see Section 8.3 and Mei et al. [252] for more discussion of transparency.

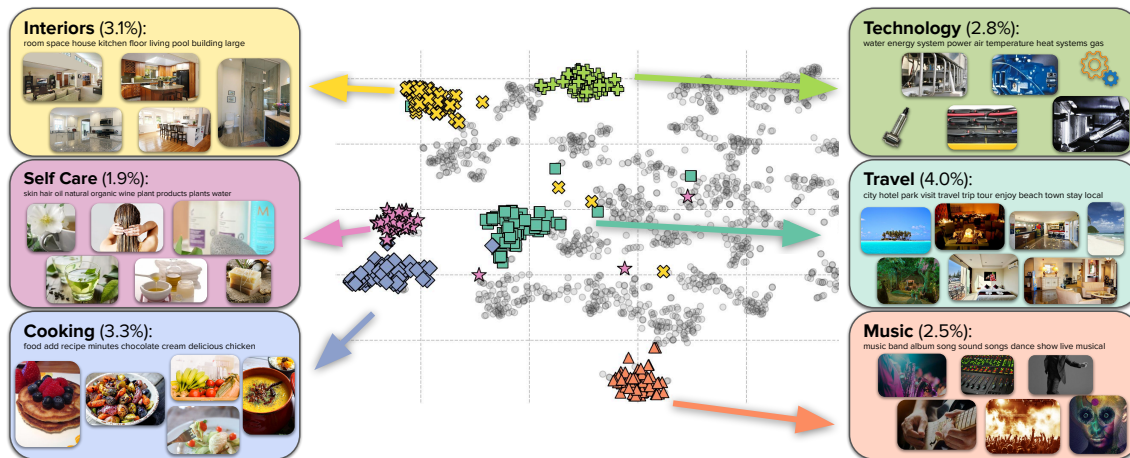


Figure 8.1: A T-SNE [255] projection of LDA [256] topic clusters from a random sample of 22K documents from `mmc4`; `mmc4` spans a variety of everyday topics, e.g., cooking, technology travel, etc. For 6 selected topics, we also show a sample of most-central images to the topic according to CLIP ViT-L/14 [23].

8.3 Data Curation Process

Initial data collection. Multimodal C4 is an expansion of the text-only `c4` dataset [84], which was created by taking the April 2019 snapshot from Common Crawl³ and applying several filters with the intention of retaining high-quality, natural English text. Each document in `c4` consists of the text scraped from one URL. The full `c4` dataset has 365M documents and 156B tokens, covering many domains [257]; it was first used to train T5 [84]. We built the `mmc4` dataset on top of `c4` because: 1) `c4` is a web-scale dataset widely adopted as a pre-training corpus [84, 258, 259, 260, 261]; 2) `c4` is constructed from web pages, which frequently contain multimedia content like images, which makes it a suitable basis for extending to a multimodal sequence version; and 3) `c4-en`,⁴ the specific underlying subset from which we construct `mmc4` has already been processed with several data-cleaning steps (including English-language identification by `langdetect`⁵ with at least 0.99 confidence; text deduplication removing duplicate three-sentence spans +

³<https://commoncrawl.org/>

⁴https://www.tensorflow.org/datasets/catalog/c4#c4en_default_config

⁵<https://pypi.org/project/langdetect/>

placeholder text like “lorem ipsum”; and removal of any document containing any word on the “List of Dirty, Naughty, Obscene or Otherwise Bad Words”).⁶ See Raffel et al. [84] for more information about the text-only c4. Importantly, by building on the popular text-only c4, prior text-only documentation efforts [257] can provide insight about potential biases and risks that could arise when training on our multimodal extension. We use the NLTK [156] sentence tokenizer to chunk each c4 document into a list of sentences.

Gathering images. We first retrieve the original webpages for each document in the c4-en dataset from the Common Crawl version 2019-18, which is the default version for c4. Next, we extract the URLs for downloadable images from the raw WAT files. We restrict the image extension to either png/jpeg/jpg, and exclude image URLs that contain the following tokens: {logo, button, icon, plugin, widget}. We attempt to download from these URLs, and resize images to a maximum dimension of 800px. We eliminate any c4 documents that do not contain valid, downloadable images at the time of collection (mid-to-late 2022). The starting point after this step is 115M documents and 1.37B images.

De-duplication+small resolution. We next run duplicate image detection using opennota’s findimagedupes⁷ which uses phash⁸ to identify visually similar images.⁹ We keep only one copy of an image if multiple versions are detected within the same document. We also remove images with more than 10 duplicates in a sample of 60K images. We discard images with a width or height smaller than 150px; this accounts for many small icons, e.g., navigation buttons. We discard images with an aspect ratio of greater

⁶<https://github.com/LDN00BW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words>

⁷<https://gitlab.com/opennota/findimagedupes>

⁸<http://www.phash.org/>

⁹We use a more aggressive de-duplication threshold of 5 compared to the default library setting of 0; this removes roughly 10M additional images. While some duplicates survive this process, we qualitatively found a threshold of 5 to be an appropriate balance of false positives/negatives.

	MSCOCO		Story-DII		Story-SIS		DII-Stress		RQA		DIY	
	AUC	p@1	AUC	p@1	AUC	p@1	AUC	p@1	AUC	p@1	AUC	p@1
Random	49.7	5.0	49.4	19.5	50.0	19.4	50.0	2.0	49.4	17.8	49.8	6.3
Hessel et al. (2019) [263]	98.7	91.0	82.6	70.5	68.5	50.5	95.3	65.5	69.3	47.3	61.8	22.5
Li et al. (2021) [264]	99.3	97.6	85.5	77.2	70.2	53.1	–	–	–	–	–	–
CLIP ViT-L/14 (Zero Shot)	99.4	95.7	92.8	93.9	79.1	73.3	98.7	93.0	80.7	70.7	74.0	57.6

Table 8.2: Performance on single document image-text benchmarks from Hessel et al. [263] (higher=better in all cases). Applying CLIP ViT-L/14 in a zero-shot fashion [23] produces better within-document alignments compared to prior methods which rely on fine-tuning.

than 2 or less than 0.5; this accounts for many banner-like ads. In a manual sample of 3.7K images that survive this (and the NSFW) filter, 91 images (2.5%) were identified as ads potentially unrelated to document contents.¹⁰



Discarding NSFW images. We employ strict NSFW image filtering, using DataComp’s [262] `dataset2metadata`¹¹ NSFW binary image classifier. The model is a 4-layer MLP, trained on the NSFW dataset introduced in LAION-2B [96]. This MLP takes as input image features extracted from OpenAI’s CLIP ViT-L/14 [23] and achieves 97.4% accuracy on the NSFW test set. We run this classifier on each image and discard cases with a model-predicted NSFW probability over 0.1, which removes approximately 10% of remaining images. Because the data distribution of the classifier and `mmc4` may be slightly different, we also conduct a spot check on images that are marked safe for work. In a manual sample of 3.7K images, we discovered zero NSFW images.

Aligning images and sentences. After collecting a set of images for each document, we now describe our intra-document alignment process to interleave the collected images

¹⁰The delineation between an “irrelevant advertisement” and a “relevant image” is inexact: for example, we discovered images advertising specific, small events, e.g., ones hosted by a fishing club within a city (this type of image was not included in this count). We later assess advertisement-ess in the context of the text of documents, rather than assessing based on the image alone.

¹¹<https://github.com/mlfoundations/dataset2metadata>

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 8.2: Two example image+text documents from `mmc4`. Following Flamingo [22], during training, images can be interleaved before or after their assigned sentences.

with the sentences. Given that the scope of the images and sentences may be different – the image set is collected from the whole webpage, while the sentence list is subject to preprocessing within the `c4` dataset and thus may not represent the complete content of the webpage – we did not rely on Document Object Model placements in the raw HTML to establish the alignment between images and sentences in each document. Instead, to associate each image with a sentence, we consider each document as an instance of a bipartite assignment problem [263, 265], and use CLIP ViT-L/14 compute pairwise similarities between all sentences/images on a single page. Then, we discard images without at least a 0.15 CLIP cosine similarity to at least one sentence in the document. Finally, we use [266] to compute a bipartite assignment of images to sentences, under the constraint that each sentence can only be assigned a single image.¹² Table 8.2 shows that this zero-shot application of CLIP ViT-L/14 for within-document matching surpasses prior competitive, fine-tuned methods on image-text alignment benchmarks from Hessel

¹²For documents with more images than sentences, after assigning an image to each sentence, we assign according to max similarity.

et al. [263] (we also distribute the raw intra-document similarity matrices with `mmc4` so alternate assignment methods can be explored). Figure 8.2 illustrates two example documents with the images interleaved before or after the assigned sentences.

8.3.1 Considerations for data release

`mmc4` contains all images that survive the previously described filters. In addition to the full version of the corpus, we construct two additional types of subsets.

8.3.1.1 Fewer Faces (`mmc4-ff`)

Like the text-only version of `c4`, `mmc4` may contain webpages with personal information that individuals had not explicitly intended to make available for model training. For an initial public release, we make a version of `mmc4` available, `mmc4-ff` (`ff` stands for “fewer faces”); similar to some prior image dataset curation efforts [267, 268], `mmc4-ff` aims to remove images containing detected faces.

Removing images with detected faces. To detect faces at billion-scale with the intent of removing them from the dataset, we first run RetinaFace [269]¹³ over a sample of 60K images with the default settings. This detector runs at a high resolution and would be computationally prohibitive to run in full precision for the whole corpus; it produces detailed localization information about the coordinates of each face in each image (which we discard). Using an 80/20 train/test split, we train a cross-validated logistic regression over CLIP ViT-L/14 features to predict whether or not RetinaFace detects a face: this classifier is several orders of magnitude faster compared to RetinaFace. This approximation performs well: we choose a confidence cutoff that achieves 95%

¹³As implemented by Serengil et al. [270, 271] available from <https://github.com/serengil/retinaface>.

recall¹⁴ for the label “RetinaFace detected any face” over the test set while preserving 65% of the original images.

Manual sample-based face image risk assessment. We performed a manual verification of face removal. In a random sample of 912 images that pass all filters including the “no faces” filter, 23 (2.5%) images arguably contain a mostly-un-obscured human face. In most cases (12/23), faces are very low resolution, e.g., a 150x150px image of a crowd of people from a distance, where each face accounts for 3x4 pixels, or are motion shots where the face is blurred. In one case, the face is Marilyn Monroe’s as depicted in art on a wall. In 6 cases, there is a plausibly identifiable face depicted: in 2 cases, these are models posing in ads; in 1 case, there is a low resolution image of politicians giving a speech; in 2 cases, the faces are obscured; in 1 case, a passerby was caught in the background of a city photograph and could feasibly be individually identified. Overall: the rate of unobscured, high-resolution, identifiable faces in `mmc4-ff` is low.

8.3.1.2 Core (`mmc4-core`)

Early conversations with some model developers revealed a desire to work with a smaller subset of the corpus as an initial step. We thus additionally release `core` versions of `mmc4` (and `mmc4-ff`), which apply even more stringent filtration criteria. The aim of `core` is to identify a “higher-precision” subset of documents that: 1) have a minimum/maximum number of sentences/images per document; 2) pass an even stricter deduplication step; and 3) have a higher image-text similarity. Hyperparameters¹⁵ are selected heuristically and are balanced to downsize the original corpus by an order of

¹⁴RetinaFace is not perfectly accurate, so selecting a more aggressive threshold (e.g., 99.99%) would not necessarily result in significantly fewer face-containing images removed.

¹⁵Min/max number of sentences: 4/40; min/max number of images 2/15; `findimagedupes` applied with a threshold of 10; documents are required to have at least 75% of image assignments have CLIP ViT-L/14 similarity of greater than 25.

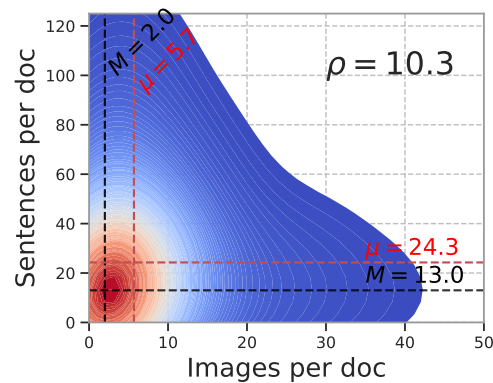


Figure 8.3: Distribution of images and sentences per document; the median document has 2 images/13 sentences. Documents with more sentences tend to have more images, but the correlation is weak (Spearman $\rho = 10.3$).

magnitude.

8.4 Exploring mmc4

Statistics. Table 8.1 gives basic summary statistics of `mmc4` (and fewer-faces/core subsets) compared to some other interleaved image/text corpora. Overall, the full version of `mmc4` is larger than prior non-public datasets across axes like number of images/number of documents. In addition, the various subsets of the corpus offer trade-offs between privacy, image/text similarity thresholds, etc. Figure 8.3 gives details about the mean/median number of images/sentences in each document (mean/median # sent.=2.0/5.7; # im = 13.0/24.3) based on a random sample of 22K documents.

Sources of documents & images. We trace back the top-level domains of documents (webpages) and images to better understand the origins of contents in `mmc4`. Figure 8.4 presents the top-20 top-level domains that host the highest number of documents and images in `mmc4`. The distribution of document sources in `mmc4` reveals a relatively uniform pattern, with 101.2M documents distributed across 6.0M unique domains. On average,

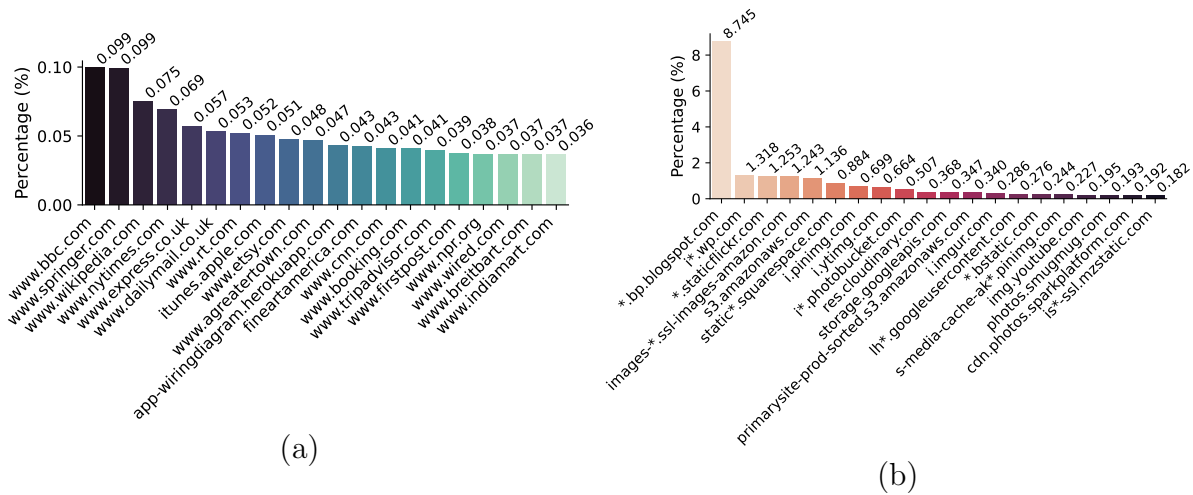


Figure 8.4: (a) Top-20 most frequent domains for `mmc4` documents. (b) Top-20 most frequent domains for images in `mmc4`.

each domain contains approximately 16.9 documents, with a median value of 2.0. The top 10% most frequently appeared domains account for 77% of all documents in `mmc4`. The documents are most commonly hosted on news media outlets (e.g., BBC, NY Times, Daily Express, Daily Mail), academic publication sites (e.g., Springer), online encyclopedias (e.g., Wikipedia), and e-commerce sites (e.g., iTunes, Etsy). Conversely, the sources of images in `mmc4` exhibit a higher level of clustering. The 571.4M images are hosted on 4.9M domains, with each domain having an average of 116.0 images and a median value of 7.0 images. The top 10% most frequent domains are responsible for hosting 89% of all images. Images are most commonly hosted on blogs (e.g., Blogspot, WordPress), shopping sites (e.g., Amazon), cloud storage sites (e.g., AWS S3, Google storage), or general image hosting sites (e.g., Flickr, Imgur).

Image-text similarity. Figure 8.5 provides detail about the linear assignment process compared to a “max” assignment alternative, where each image is simply assigned to its maximally CLIP-similar sentence. The linear assignment process slightly decreases the average CLIP similarity between images/sentences (from 24.5 \rightarrow 24.0), but signif-

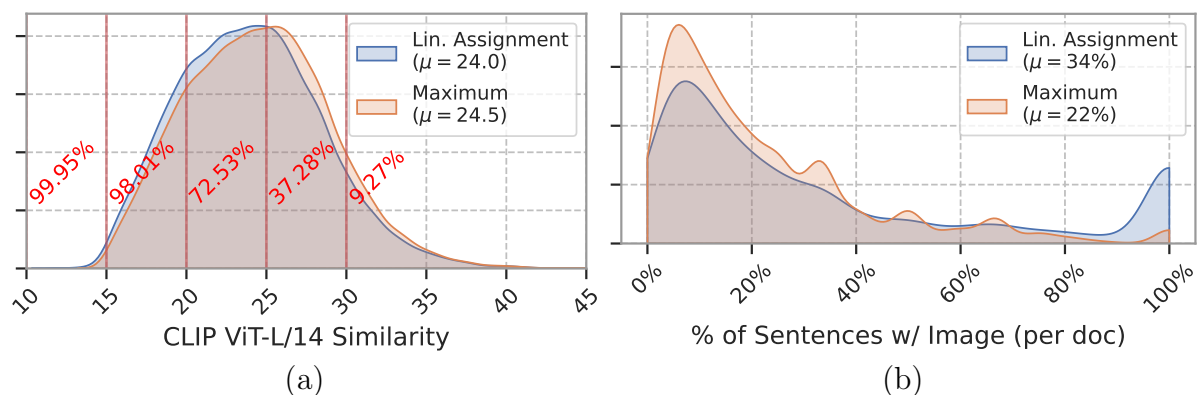


Figure 8.5: Using linear assignment results in comparable image-text similarities to max assignment, but the former spreads images much more evenly, e.g., the per-document mean percent of sentences with an associated image increases from 22% to 34%. (a) CLIP sim is similar between lin. assignment + max. In red: percent of images remaining at various CLIP thresholds. (b) Lin. assignment results in a higher percentage of sentences being associated with an image.

icantly more evenly “spreads” images throughout the documents: per-document, the mean percentage of sentences with an associated image rises from 22% \rightarrow 34%.

Topic-based assessment. We ran LDA [256] as implemented by Mallet [272] on a random sample of 22K documents from `mmc4` with $k = 30$ topics. The resulting clusters span a broad set of topics like cooking, communities, travel, music, art, etc. Figure 8.1 shows some example LDA topic clusters. In addition, we explore a sample of the images most associated with the corresponding topic,¹⁶ finding that, in general, image topic clusters align with qualitative expectations.

Manual verification of image relevance+properties. We randomly sample 200 documents from `mmc4` with the goal of assessing how relevant the images contained in the document are to the assigned sentences and to the document as a whole. Table 8.3

¹⁶We compute the mean CLIP ViT-L/14 image vector for each topic by associating each image in a document the document’s most common topic; then, we compute the mean image vector per topic. Finally, cosine similarity to this mean vector is used to identify the “most topically central” images per-topic.

% of 836 images	
Topically-related	87.7%
Sentence-aligned	80.4%
Has face?	28.3%
Has watermark?	1.6%
Logo-related	3.9%
Ads-related	3.2%
Duplicated	0.7%

Table 8.3: Results of manual verification of 200 randomly sampled documents containing 836 images. A majority of images are topically relevant and well sentence-aligned. The rate of watermarks, ads, duplicates, etc. is low.

shows the results on the 836 images contained in the 200 documents. 87.7% of all examined images are topically related to the corresponding document, and 80.4% images are well-aligned to the assigned sentences within each document.¹⁷ We also assessed several other factors, finding that: 1) 28.3% contain recognizable human faces; 2) 1.6% contain recognizable watermarks; 3) 3.9% are related to logos;¹⁸ 4) 3.2% are related to advertisements; and 5) 0.7% are duplicated with other images in the same document.

8.5 OpenFlamingo: An Early Application of mmc4

The first publicly available model to be trained on mmc4 is OpenFlamingo [20]. We run ablations on a small version of OpenFlamingo (3B: backbone = OPT-1.3B [238] language model and CLIP ViT-L/14 [23] vision model) to compare direct training on image captions (LAION-2B [96]) to the interleaved sequences of mmc4-core.¹⁹ To flatten

¹⁷The alignment between an image and its assigned sentence is a qualitative criterion. We consider an image-sentence pair to be “well-aligned” when the visual elements of the image have a direct and relevant relationship with the text. This can include instances where the image depicts the context or content of the sentence, or where there is a plausible literal overlap between the text and the image, etc.

¹⁸The logos can be website logos, commercial logos used by businesses or companies to represent their brand or product, or logos for organizations or events. In all cases, the label is assigned if the logo is the primary focus of the image.

¹⁹These experiments were conducted using a preliminary v1 of the mmc4-core corpus, see this pull request for discussion of small bugfixes in the current v1.1 version.

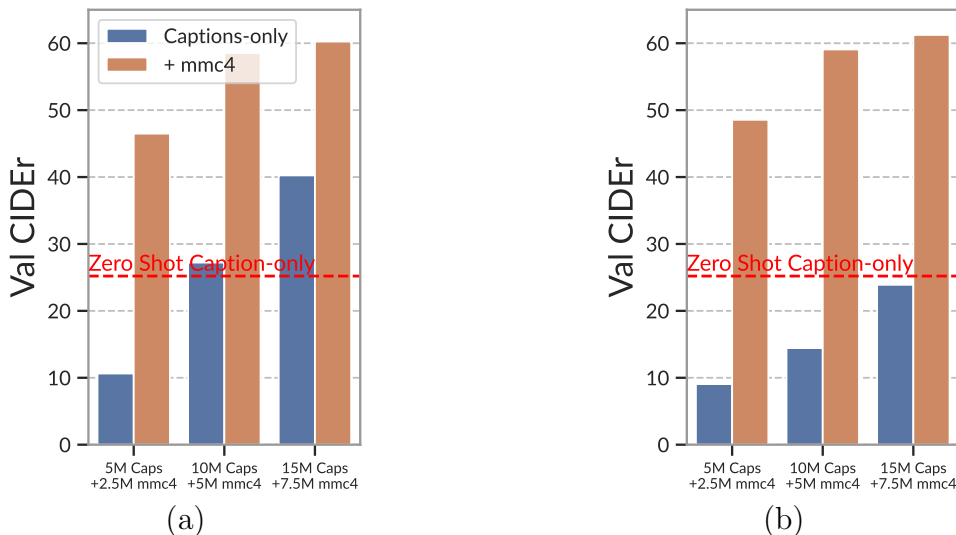


Figure 8.6: Few shot, in-context MSCOCO captioning performance of OpenFlamingo-3B when training on **just captions** from LAION-2B vs. **mixing in mmc4-core** sequences. **(a)** 4-shot. **(b)** 8-shot. **The model trained on mmc4 sequences is able to generalize to MSCOCO-style captions more effectively vs. the model trained just on LAION-2B image/caption pairs.** (**Zero shot caption-only**=15M caption LAION-2B model)

mmc4 documents to training sequences,²⁰ we: 1) sample a 256 token sub-sequence from each training document; 2) discard images with CLIP image-text similarity less than 20; 3) discard sequences that contain no images after filtering; 4) discard images if there are more than 5 in the resulting sequence.²¹ As in Huang et al. [254] we randomly drop sequences with a single image to increase multi-image sequences in the sample.

Validation CIDEr [115] results for COCO image captioning are in Figure 8.6. For 4/8-shot in-context learning settings, the model trained on mmc4-core shows 20-30 CIDEr point improvements. The performance of OpenFlamingo-3B trained on just 5M captions/2.5M mmc4 sequences also exceeds a zero-shot application of OpenFlamingo-3B trained on much more data (15M LAION-2B captions); this provides additional evidence

²⁰Future work would be well-suited to investigate the impact of various flattening schemes on downstream performance; the method described here is just one possible method.

²¹Similar to Flamingo [22], we find that training on a maximum of five image sequences can be sufficient for OpenFlamingo models to generalize to 32 shots during inference.

that the interleaving in-context setup enables adaptation to MSCOCO-style captions. The performance of the captions-only OpenFlamingo-3B model degrades from 4-shot to 8-shot learning presumably because these longer sequences are significantly different from the single image/captions it's seen at training time.

Chapter 9

OpenFlamingo: A Framework for Training Autoregressive Vision-Language Models

9.1 Introduction

A popular format for vision and language models is (image, text) \rightarrow text, i.e., models take as input an image and some text, and produce text as output, e.g., BLIP-2 [273]. The flexible format directly supports tasks like image classification and visual question answering (VQA).

However, assuming a single image as input is limiting: autoregressive vision-language models enable new capabilities by instead mapping an arbitrarily interleaved *sequence* of images and text to textual outputs. This interface provides important flexibility: the input sequence can include demonstrations for a new task, enabling few-shot, in-context learning [22] or multi-round multi-modal chatbot interactions. Evaluations suggest that autoregressive vision-language models can be performant foundation models [274]: mod-

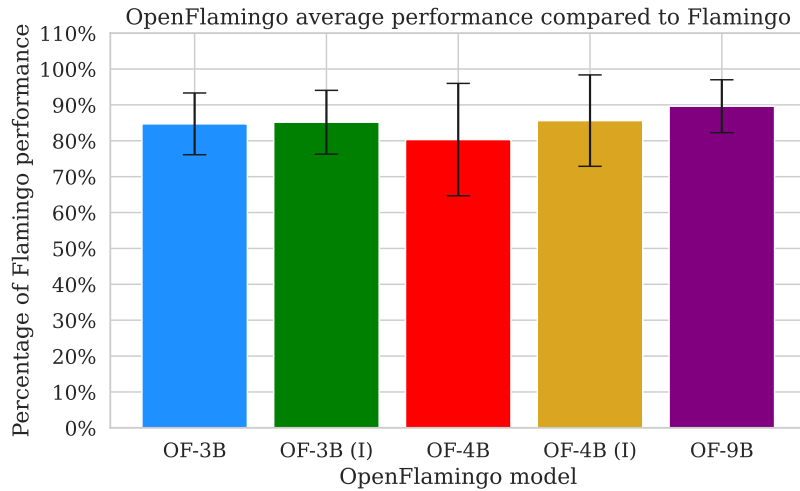


Figure 9.1: OpenFlamingo performance as a fraction of corresponding Flamingo performance, averaged across evaluation settings (7 datasets \times 5 options for number of in-context examples). Demonstrations are chosen using RICES (Retrieval-based In-Context Example Selection). More details regarding selecting demonstrations can be found in Section 9.3.4. We compare OpenFlamingo-3B and -4B models to Flamingo-3B, and OpenFlamingo-9B to Flamingo-9B. Error bars are standard deviations over settings. “OF-3B (I)” refers to OpenFlamingo-3B (Instruct), the 3B model trained with a language-instruction-tuned backbone.

els like Flamingo [22], CM3 [253], Kosmos-1 [254], PALM-E [275], and multimodal GPT-4 [205] generalize well across diverse vision-language tasks.

Unfortunately, these autoregressive vision-language models are closed-source, and their weights, training data, code, and hyperparameters are proprietary. This limits the academic community’s ability to conduct research on autoregressive vision-language models, e.g., to understand how web-scraped image-text data affects models’ performance and safety. Open-source alternatives, such as LLaVA [276], LLaMA-Adapter [277], BLIP-2 [278], and mPLUG-Owl [279], only take in single images, and they often directly train on curated datasets like COCO [117] rather than web data.

In this technical report, we document our experiences building an open-source reproduction of the Flamingo models [22]. Following Flamingo, we augment the layers of pretrained, frozen language models so that they cross attend to the outputs of a frozen

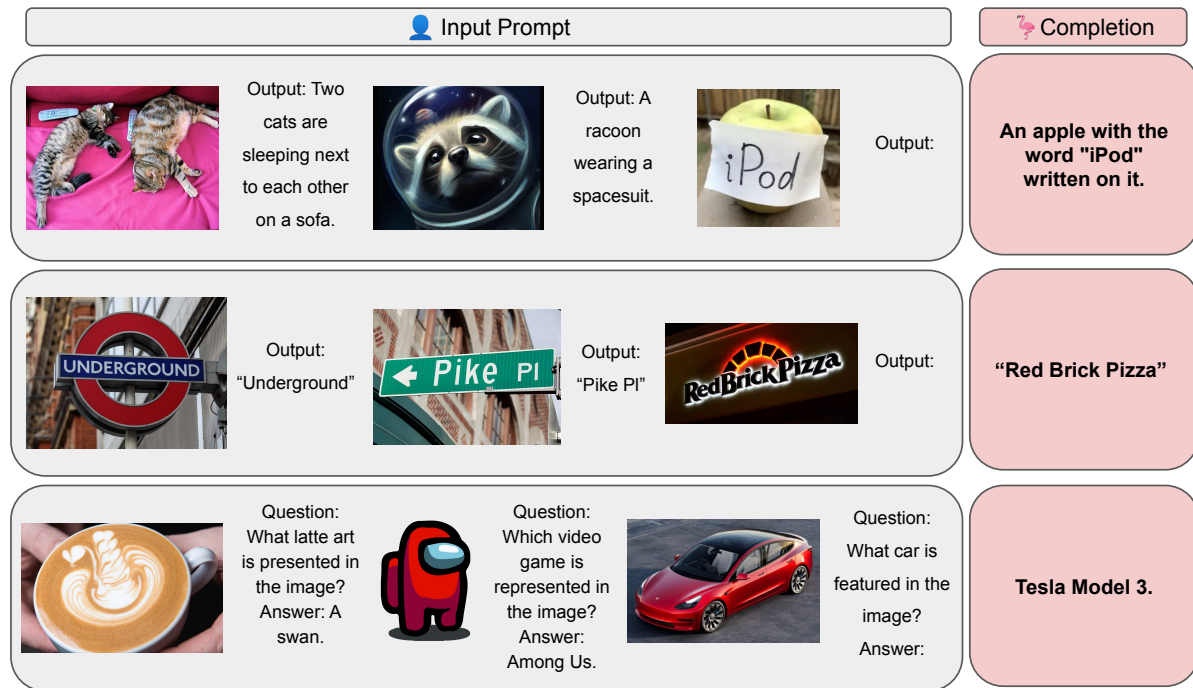


Figure 9.2: OpenFlamingo-9B (pictured) can process interleaved image-and-text sequences. This interface allows OpenFlamingo to learn many vision-language tasks through in-context demonstrations.

vision encoder while predicting the next token. The cross-modal module is trained on web-scraped image-text sequences, in our case, two open source datasets: LAION-2B [96] and Multimodal C4 [19]. Our stack is built using publicly available components, including CLIP as a vision encoder [23] and open-source language models as decoders [280, 281].

We call the resulting family of five models OpenFlamingo. These models range from 3B to 9B parameters, with both standard and instruction-tuned [200] language model backbones. When averaging performance across 7 evaluation datasets, OpenFlamingo-3B and -9B models attain 85% and 89% of their corresponding Flamingo models respectively (Figure 9.1). Models and code are open-sourced at https://github.com/mlfoundations/open_flamingo.

9.2 Related work

Generative vision-language models output text conditioned on an image-text sequence. While many such architectures, such as BLIP-2 and LLaVa, can incorporate only one image in their context [195, 273, 276, 277, 279, 282], autoregressive vision-language models accept interleaved image-text sequences, enabling in-context learning.

We chose to replicate Flamingo because of its strong in-context learning abilities. Aggregated across evaluation sets, Flamingo models see steady performance improvements up to 32 in-context examples [22]. This is in contrast with other autoregressive vision-language models, for example Kosmos-1 [254]; on captioning tasks COCO [117] and Flickr-30K [149], Kosmos-1 shows performance improvements up to 4 in-context examples, but performance degrades when using 8 in-context examples.

Open-source image-text datasets. Proprietary autoregressive vision-language models are typically trained on closed-source datasets [22, 253, 254, 275]. For example, Flamingo relies on image-text pairs from the ALIGN dataset [283] and interleaved image-text sequences from the M3W dataset [22]; both are unavailable to the public. Recent efforts to replicate these web-scraped datasets include LAION-2B, a dataset of image-text pairs, and Multimodal C4 [19] and OBELISC [284], datasets of image-text sequences. We use LAION-2B and Multimodal C4 for training OpenFlamingo models. Laurençon et al. [284] also train 9B and 80B Flamingo-style models; their models differ in the choice of pretraining dataset (OBELISC instead of Multimodal C4) and language model (LLaMA-9B [277] instead of the MPT and RedPajama-3B models [280, 281]).

Model	Language model	Cross-attention interval	<image> and < endofchunk >
OpenFlamingo-3B	MPT-1B [280]	1	Trainable
OpenFlamingo-3B (Instruct)	MPT-1B (Instruct) [280]	1	Trainable
OpenFlamingo-4B	RedPajama-3B [281]	2	Frozen
OpenFlamingo-4B (Instruct)	RedPajama-3B (Instruct) [281]	2	Frozen
OpenFlamingo-9B	MPT-7B [280]	4	Trainable

Table 9.1: Architecture details of the OpenFlamingo models. All five models use a CLIP ViT-L/14 vision encoder [23]. A cross-attention interval of 4 means that a cross-attention module is inserted every 4th language model layer. Note that OpenFlamingo models labeled (Instruct) use language models that were finetuned on language-only tasks; we have not instruction-tuned OpenFlamingo models on vision-language tasks.

9.3 Approach

9.3.1 Architecture

We match the Flamingo architecture [22]. Given an interleaved sequence of images with text tokens, OpenFlamingo models predict the next text token conditioned on all previous text tokens and the last preceding image. Text tokens attend to their corresponding images via *dense cross-attention modules*, which we attach to the layers of a frozen, autoregressive language model. To embed images, we extract patch features from a frozen vision encoder and pass these through a trainable Perceiver resampler [285].

As a preprocessing step, we first mark the locations of images in the text sequence with <image> tokens. We also insert <|endofchunk|> tokens after the text tokens following an image; *e.g.* the sequence x Hello world, where x is an image, would be preprocessed into <image> Hello world <|endofchunk|> .

Unlike Flamingo, we do not support video inputs at this time. We leave this for future work.

Table 9.1 describes the five OpenFlamingo models based on their language model and density of cross-attention layers; all models use CLIP ViT-L/14 [23] as a vision encoder.

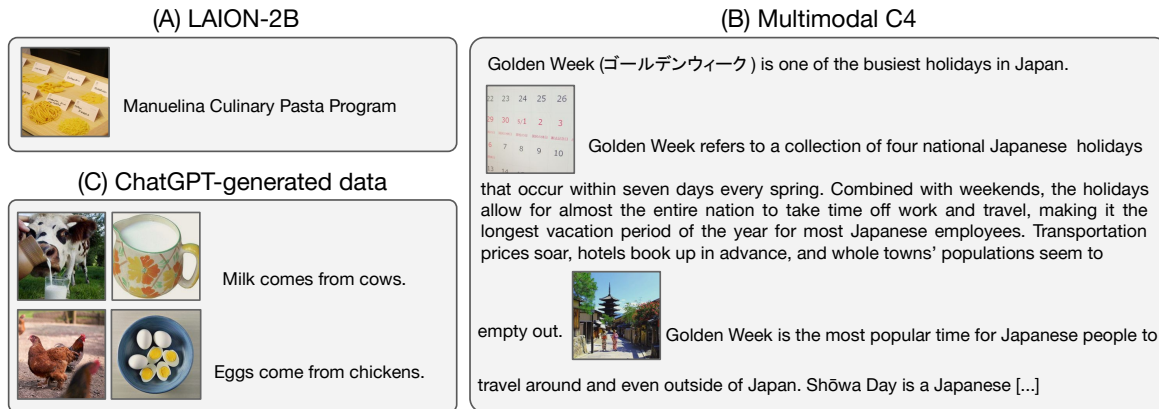


Figure 9.3: Samples from (A) LAION-2B [96], (B) Multimodal C4 [19], and (C) ChatGPT-generated data.

In most cases, the `<image>` and `<|endofchunk|>` embeddings are trainable, while other text embeddings are frozen. For the OpenFlamingo-4B models, all embeddings are frozen, including the randomly initialized `<image>` and `<|endofchunk|>` embeddings. This was due to complications with gradient masking when using Fully Sharded Data Parallel (Section 9.3.3).

9.3.2 Training data

We train our models on a mixture of image-text pairs and interleaved image-text sequences. During training, we sample dataset shards with replacement using the Web-Dataset format [286].

LAION-2B [96]. When training Flamingo, Alayrac et al. [22] use ALIGN [283], a closed-source dataset of over 1B single images paired with short alt-text captions. To train OpenFlamingo, we replace ALIGN with LAION-2B, an open-source web-scraped dataset consisting of 2B image-text pairs (Figure 9.3A). We use part of the English subset and truncate captions to 32 tokens. All image-text pairs in LAION-2B have a

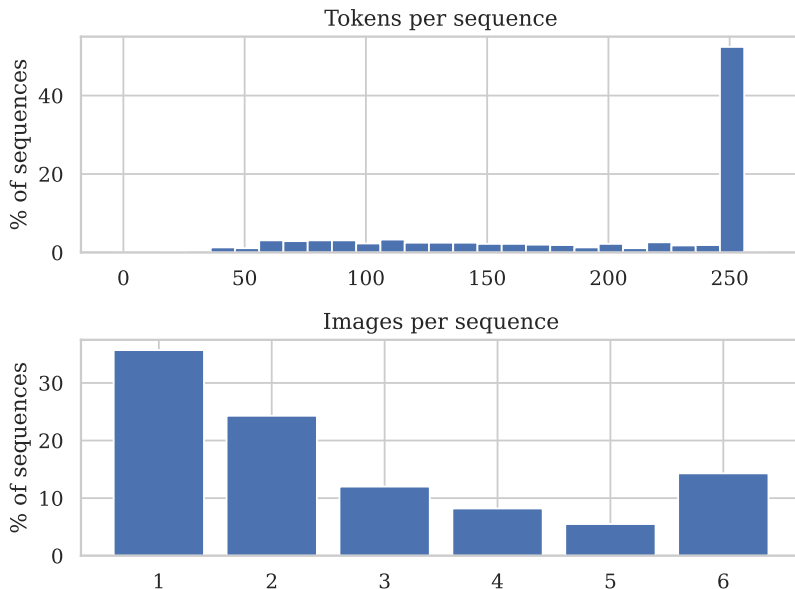


Figure 9.4: Histograms of the number of text tokens and images per MMC4 sequence, based on a sample of 1,000 sequences. Sequences are long with few images.

cosine similarity of at least 0.28 according to CLIP ViT-B/32.

Multimodal C4 [19]. In addition to image-text pairs, Alayrac et al. [22] train Flamingo using M3W, an internal web-scraped dataset of 43M interleaved image-text sequences. We replace M3W with Multimodal C4 (MMC4), an open-source dataset of 101M interleaved samples (Figure 9.3B). Unlike M3W or OBELISC [284], which directly parse HTML documents to extract multimodal sequences, MMC4 uses CLIP to soft align images with sentences in a document. To ensure data quality, we exclude images if their cosine similarity with the subsequent text falls below 0.24, according to CLIP ViT-L/14. Sequences contain between 1 and 6 images (median 2). To encourage learning from sequences with multiple images, we reject single-image sequences with probability 0.5. The resulting distribution is shown in Figure 9.4.

Dataset	Median images per sequence	Median tokens per sequence
LAION-2B	1	17
MMC4	2	256
ChatGPT	3	56

Table 9.2: Statistics for training datasets. “ChatGPT” stands for the ChatGPT-generated sequences. The median numbers of images and tokens per sequence were calculated using a random sample of 1,000 sequences.

Synthetic data. For the OpenFlamingo-4B models, we also experimented with training on ChatGPT-generated synthetic data (Figure 9.3C) These 417K image-text sequences were generated by prompting ChatGPT to generate a sequence of interleaved text and image alt-texts (in place of images). The alt-texts are used to retrieve a corresponding images from LAION-5B. The median number of images per sequence is higher than in MMC4, while the median number of text tokens is lower (Table 9.2). We release these sequences through the OpenFlamingo repository.

9.3.3 Training details

OpenFlamingo models were trained for 60M interleaved (MMC4) examples¹ and 120M LAION-2B examples. All models are trained using the next-token prediction objective and optimized with AdamW. The learning rate is linearly increased at the beginning of training, and then held constant at 1e-4 throughout training. We apply weight decay of 0.1 on the dense cross attention layers. The batch size for LAION-2B is twice the batch size of the interleaved dataset (MMC4, optionally with ChatGPT-generated sequences), and the loss weights are set to Flamingo defaults of 1 and 0.2 for MMC4 and LAION-2B respectively. We accumulate gradients over both datasets between optimizer steps.

¹OpenFlamingo-4B models use both MMC4 and ChatGPT-generated data as interleaved sequences; 60M interleaved examples translates to approximately 240K ChatGPT-generated sequences and 59.8M MMC4 sequences. Other models train on 60M MMC4 examples.

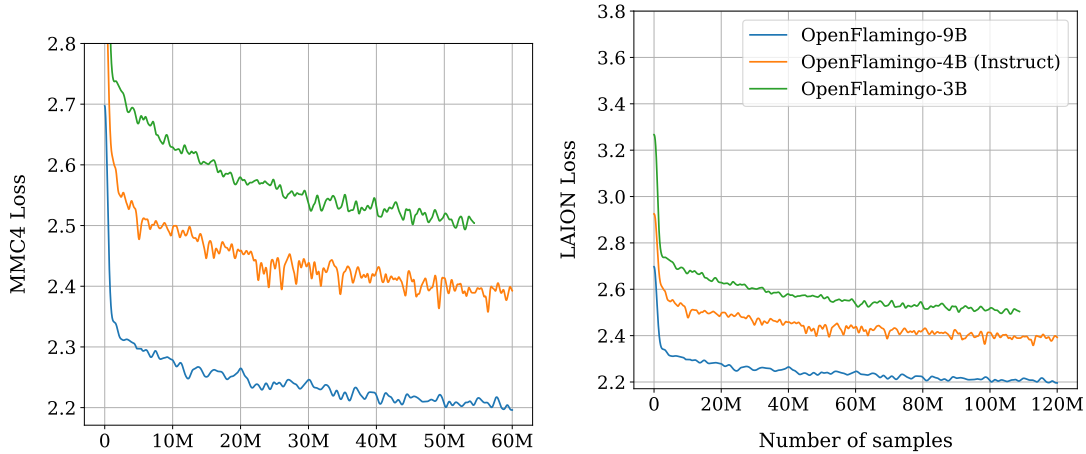


Figure 9.5: MMC4 and LAION-2B language modeling loss throughout training. Curves shown with Gaussian smoothing with window size 100.

Model	GPU type	Sharding strategy	Precision
OF-3B	A100-80GB	DDP	fp32
OF-3B (I)	A100-40GB	DDP	fp32
OF-4B	A100-40GB	FSDP	fp32
OF-4B (I)	A100-40GB	FSDP	fp32
OF-9B	A100-80GB	DDP	amp_bf16

Table 9.3: Training used either DistributedDataParallel (DDP) or FullyShardedDataParallel (FSDP) [287].

Distributed training. We train all models using 64 GPUs distributed across 8 nodes on Stability AI’s cluster (Table 9.3). OpenFlamingo-4B models were trained using model sharding with Fully Sharded Data Parallel [287]; other models were trained using only data parallel.

Loss curves. Figure 9.5 tracks LAION-2B and MMC4 loss over the course of training. After an initial improvement, MMC4 loss decreases very slowly. We speculate that, since MMC4 sequences tend to include long paragraphs between images (Figure 9.2), most text tokens can be generated without referencing the image. Thus, the loss may be dominated by whether the frozen language model can fit unrelated paragraphs of text.

9.3.4 Evaluation method

We evaluate OpenFlamingo on seven vision-language datasets including captioning (COCO [288], Flickr-30K [289]), visual question answering (VQAv2 [290], OK-VQA [291], TextVQA [292], VizWiz [293]), and rank classification (HatefulMemes [294]). For each dataset, we measure performance at 0, 4, 8, 16, and 32 in-context examples. Evaluation was done in automatic mixed precision, with linear layers computed in bfloat16.

Selecting in-context examples. For each evaluation example, we sample in-context examples from the training split uniformly at random.

Evaluation subsets. We evaluate on the dataset splits used by Alayrac et al. [22]. We run each evaluation across three seeds, where the randomness is over selected in-context demonstrations, and average the results to obtain our final scores.

Prompts. For captioning tasks, we format demonstrations as `<image> Output: [caption]`, replacing `[caption]` with the ground-truth caption. For VQA, we format examples as `<image> Question: [question] Short answer: [answer]`. For HatefulMemes, we prompt the model with `<image> is an image with: '[text]' written on it. Is it hateful? Answer: [answer]`.

Following Alayrac et al. [22], we prompt the model with two in-context examples during zero-shot evaluations, removing their images, and for classification tasks, we implement prompt ensembling by averaging logits across 6 permutations of the in-context examples.

Decoding parameters. We evaluate captioning and VQA using beam search with 3 beams, stopping generation at 20 tokens for captioning, 5 tokens for VQA, or whenever

Benchmark	Shots	F1-3B	F1-9B	OF-3B	OF-3B (I)	OF-4B	OF-4B (I)	OF-9B
COCO [288]	0	73.0	79.4	74.9 (0.2)	74.4 (0.6)	76.7 (0.2)	81.2 (0.3)	79.5 (0.2)
	4	85.0	93.1	77.3 (0.3)	82.7 (0.7)	81.8 (0.4)	85.8 (0.5)	89.0 (0.3)
	32	99.0	106.3	93.0 (0.6)	94.8 (0.3)	95.1 (0.3)	99.2 (0.3)	99.5 (0.1)
Flickr-30K [289]	0	60.6	61.5	52.3 (1.0)	51.2 (0.2)	53.6 (0.9)	55.6 (1.3)	59.5 (1.0)
	4	72.0	72.6	57.2 (0.4)	59.1 (0.3)	60.7 (1.2)	61.2 (0.5)	65.8 (0.6)
	32	71.2	72.8	61.1 (1.3)	64.5 (1.3)	56.9 (0.7)	53.0 (0.5)	61.3 (0.7)
VQA _{v2} [290]	0	49.2	51.8	44.6 (0.0)	44.1 (0.1)	45.1 (0.1)	46.9 (0.0)	52.7 (0.2)
	4	53.2	56.3	45.8 (0.0)	45.7 (0.1)	49.0 (0.0)	49.0 (0.0)	54.8 (0.0)
	32	57.1	60.4	47.0 (0.1)	44.8 (0.1)	43.0 (0.2)	47.3 (0.0)	53.3 (0.1)
OK-VQA [291]	0	41.2	44.7	28.2 (0.2)	28.7 (0.1)	30.7 (0.1)	31.7 (0.1)	37.8 (0.2)
	4	43.3	49.3	30.3 (0.5)	30.6 (0.2)	35.1 (0.0)	34.6 (0.0)	40.1 (0.1)
	32	45.9	51.0	31.0 (0.1)	30.6 (0.1)	26.4 (0.2)	34.7 (0.3)	42.4 (0.0)
TextVQA [292]	0	30.1	31.8	24.2 (0.2)	23.1 (0.2)	21.0 (0.3)	21.1 (0.4)	24.2 (0.5)
	4	32.7	33.6	27.0 (0.3)	28.1 (0.4)	25.9 (0.0)	27.2 (0.3)	28.2 (0.4)
	32	30.6	32.6	28.3 (0.2)	28.5 (0.1)	14.1 (0.2)	23.2 (0.2)	23.8 (0.2)
VizWiz [293]	0	28.9	28.8	23.7 (0.5)	23.4 (0.3)	18.8 (0.1)	21.5 (0.2)	27.5 (0.2)
	4	34.0	34.9	27.0 (0.3)	27.7 (0.1)	26.6 (0.5)	26.5 (0.4)	34.1 (0.7)
	32	45.5	44.0	39.8 (0.1)	39.3 (0.4)	23.1 (1.1)	31.3 (0.2)	44.0 (0.5)
HatefulMemes [294]	0	53.7	57.0	51.2 (2.5)	50.1 (2.2)	52.3 (2.3)	53.1 (2.2)	51.6 (1.8)
	4	53.6	62.7	50.6 (0.8)	49.5 (0.6)	51.5 (1.4)	54.9 (1.1)	54.0 (2.0)
	32	56.3	63.5	50.2 (1.8)	47.8 (2.2)	52.2 (1.2)	54.9 (1.1)	53.8 (2.1)

Table 9.4: Evaluation results across seven vision-language datasets using 0, 4, and 32 in-context examples. “OF-3B (I)” refers to OpenFlamingo-3B (Instruct), the 3B model trained with a language-instruction-tuned backbone, while “F1-3B” refers to Flamingo-3B. Flamingo results taken from Alayrac et al. [22]. The highest number in each row is bolded.

the model produces an `<|endofchunk|>` token. For HatefulMemes, we compute the log-likelihood of completions “yes” and “no” and answer with the most likely completion.

Metrics. For captioning, we use CIDEr score [115]. For VQA, we report VQA accuracy, *i.e.*, exact match accuracy over a set of ground truth answers [290]. For HatefulMemes, we compute AUC ROC.

9.4 Results

In Table 9.4, we compare OpenFlamingo and Flamingo models across 0, 4, and 32 in-context examples. On average, OpenFlamingo-3B, -3B (Instruct), -4B (Instruct), and -9B attain more than 86% of the performance of their corresponding Flamingo models

(Figure 9.1).

In the 0- and 4-shot regimes, OpenFlamingo models approach or match Flamingo performances on several datasets. For example, OpenFlamingo-9B improves upon Flamingo-9B’s 0-shot performance on VQAv2 (51.8% \rightarrow 52.7% VQA accuracy) and COCO (79.4 \rightarrow 79.5 CIDEr), and OpenFlamingo-9B approaches Flamingo-9B’s 0-shot performance on Flickr-30K and VizWiz. Moreover, OpenFlamingo-9B approaches the 4-shot performance of Flamingo-9B on COCO, VQAv2, and VizWiz.

However, on OK-VQA and TextVQA, OpenFlamingo models are notably weaker than their Flamingo counterparts: OpenFlamingo-9B underperforms Flamingo-9B in 0-shot evaluations by 6.9 percentage points on OK-VQA and 7.8 percentage points on TextVQA. OpenFlamingo-3B also underperforms Flamingo-3B by 4.6 percentage points in 0-shot VQAv2 accuracy. The reason for generally low VQA performance is unclear, although discussions in Section 9.5.2 may be related.

Extrapolating to more in-context examples. In Figure 9.6, we plot performance as a function of the number of in-context examples. We observe that the OpenFlamingo-3B and -9B models generally improve with the number of in-context examples. However, the rate of improvement is lower than the Flamingo models: in the bottom right corner of Figure 9.6, we observe that gaps between OpenFlamingo-9B and Flamingo-9B widen with the number of in-context examples. We speculate that this behavior may stem from the quality of our pre-training data, which mostly consists of sequences with few images (Table 9.2). In contrast with the -3B and -9B models, which generally improve with more in-context examples, the OpenFlamingo-4B models unexpectedly degrade in performance after 4 or 8 shots. The 4B models use RedPajama language models [281] instead of MPT backbones [280]; they also use frozen `<image>` and `<|endofchunk|>` embeddings. We investigate the effect of the latter in Section 9.5.1.

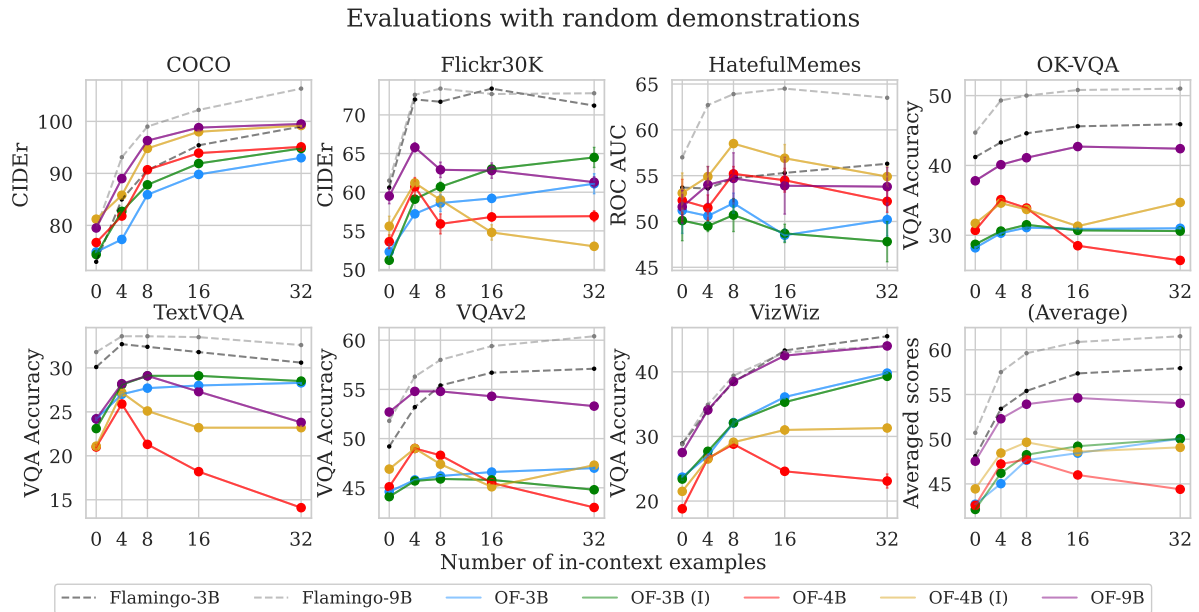


Figure 9.6: Evaluation results per dataset across 0, 4, 8, 16, and 32 in-context examples. Each point is the average across three evaluation runs, where the randomness is over choice of in-context demonstrations. Error bars are standard deviations over random seeds.

Trends by model size. OpenFlamingo-9B generally outperforms smaller models, except on HatefulMemes and for large numbers of in-context examples on Flickr-30K and TextVQA. However, OpenFlamingo-4B models often underperform the smaller 3B models, including on Flickr-30K, HatefulMemes, TextVQA, and VizWiz.

Effect of language instruction-tuning. We train two OpenFlamingo models at each of the 3B and 4B scales: one model using a base language model, and one with an instruction-tuned variant of the same language model. In the lower right corner of Figure 9.6, we observe that the instruction-tuned variants of MPT-1B and RedPajama-3B on average outperform the base models. The difference is starkest for RedPajama-3B. Transfer of language instruction tuning to vision-language tasks was previously reported in Huang et al. [254], Li et al. [278].

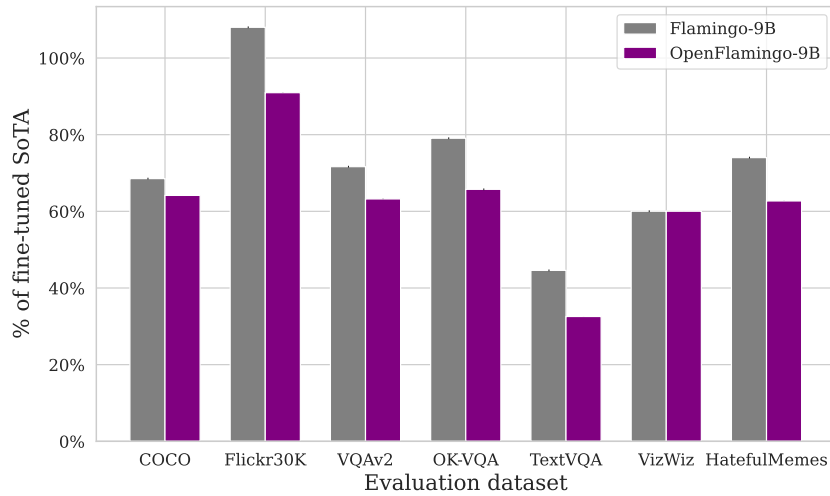


Figure 9.7: OpenFlamingo-9B and Flamingo-9B performance relative to fine-tuned SoTA performance.

Comparison to fine-tuned state-of-the-art. Figure 9.7 plots each model’s performance relative to fine-tuned state-of-the-art performance, as listed on Papers With Code on June 19, 2023. OpenFlamingo-9B averages more than 62% of fine-tuned state-of-the-art performance with 32 RICES-selected in-context examples, compared to 72% achieved by Flamingo-9B.

9.5 Discussion

9.5.1 Frozen embeddings

In Section 9.4, we observed that OpenFlamingo-4B models underperform their 3B counterparts on most datasets. One notable way the OpenFlamingo-4B models differ from the 3B and 9B models is that their `<image>` and `<|endofchunk|>` embeddings are randomly initialized and frozen, rather than trained.

In Table 9.5, we investigate the effect of this difference. We train small models using OPT-125M as a language model [238] to 20M interleaved samples (one-third of

		0-shot	4-shot	8-shot
COCO	trainable	46.5	58.6	61.2
	frozen	41.9 (-4.6)	54.5 (-4.1)	57.4 (-3.8)
VQAv2	trainable	17.6	23.2	28.7
	frozen	5.5 (-12.1)	8.4 (-14.8)	18.8 (-9.9)

Table 9.5: COCO and VQAv2 validation performance when using trainable `<image>` and `<|endofchunk|>` embeddings compared to frozen, randomly initialized embeddings. The model used in this experiment is based on CLIP ViT-L/14 and OPT 125M, with cross-attention every layer, and trained on 20M interleaved samples, including ChatGPT-sequences.

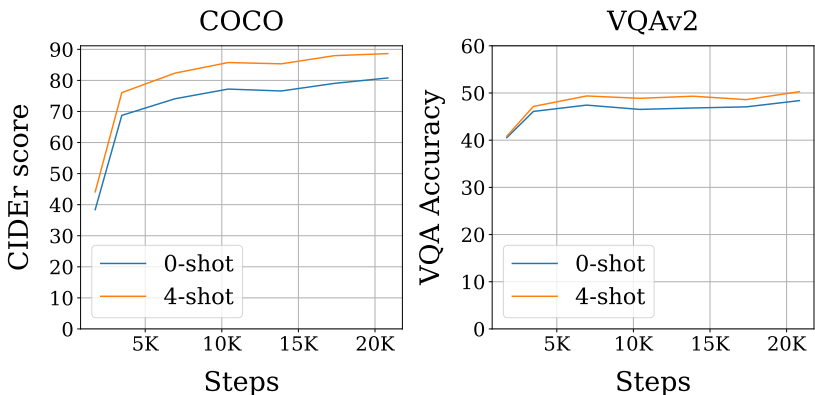


Figure 9.8: Validation split performance for OpenFlamingo-9B across training: while COCO CIDEr improves throughout training, VQAv2 performance is more stagnant.

full training). Freezing the `<image>` and `<|endofchunk|>` embeddings results in a drop of 4.6 CIDEr for 0-shot COCO, and 12.1% accuracy for 0-shot VQAv2. This suggests that frozen `<image>` and `<|endofchunk|>` embeddings may impact downstream trends.

9.5.2 VQAv2 validation trends

During development, we used the VQAv2 validation set as a temperature check for visual question answering capabilities. In this section, we discuss trends observed during development.

Training dynamics. To understand how evaluation performance evolves over the course of training, Figure 9.8 plots validation performance of OpenFlamingo-9B on

Language model	VQAv2 validation	
	<i>Shots</i>	
	0	4
OPT-125M	17.6	23.2
OPT-1.3B	32.8	27.2
MPT-1B (Instruct)	41.9	43.7
MPT-7B	47.4	49.4

Table 9.6: VQAv2 validation performance at 20M interleaved samples across different language models. Performance largely differs between language models.




Counting	Verbosity	Non-central object
 <p>Q: How many people are on the sidewalk?</p> <p>OF-9B: “one”</p> <p>GROUND TRUTH: { “4”, “5” }</p>	 <p>Q: What is this sheep trying to do?</p> <p>OF-9B: “it is trying to get”</p> <p>GROUND TRUTH: { “get out”, “escape” }</p>	 <p>Q: What color are the curtains?</p> <p>OF-9B: “green”</p> <p>GROUND TRUTH: { “yellow”, “gold” }</p>

Table 9.7: OpenFlamingo-9B errors from the VQAv2 validation split. Common failure modes for OpenFlamingo including counting, giving answers that are too verbose (and thus truncated), and answering about the central object in the image rather than the non-central object in the question.

COCO and VQAv2 throughout training. While COCO performance steadily improves, VQAv2 progress is flatter. This matches trends reported by Li et al.[278].

Effect of language model. Although additional training did not dramatically affect VQAv2 performance, changing language model backbones did. Table 9.6 illustrates this effect on the VQAv2 validation split; notably, switching from OPT-1.3B to MPT-1B (Instruct) added nearly 10 percentage points in 0-shot performance. We hypothesize that the language model has similarly large effects for other VQA tasks.

Common VQA failure modes (Table 9.7). OpenFlamingo models struggle with counting; on the VQAv2 validation split, OpenFlamingo-9B scores 30.5% on questions

with numerical answers, compared to 70.6% on yes / no questions. Additionally, because VQA accuracy uses an exact match criterion for generations, models must answer concisely to score well; OpenFlamingo models are often too verbose. Finally, VQA questions can ask about objects other than the central object in the image; models sometimes answer about the central item instead.

9.5.3 Applications of OpenFlamingo

Multiple models have already developed on top of OpenFlamingo. Li et al. [295] fine-tuned OpenFlamingo on MIMIC-IT [296], a multi-image/video instruction following dataset, creating Otter, a multimodal assistant. Gong et al.[297] released Multimodal-GPT, an OpenFlamingo model instruction fine-tuned on both vision and language instruction datasets. We hope the community continues to use OpenFlamingo models.

9.5.4 Limitations

OpenFlamingo models carry the same risks as their foundational language models. In particular, these models train on web-scraped data, and they have not undergone safety-focused fine-tuning. Models thus may produce unexpected, inappropriate, or inaccurate outputs. We hope to further investigate the safety properties of autoregressive vision-language models like OpenFlamingo.

Chapter 10

VisIT-Bench: A Benchmark for Vision-Language Instruction Following

10.1 Introduction

A long-standing challenge for artificial intelligence is to build general-purpose assistants that can, in collaboration with humans, solve diverse and never-before-seen tasks [298]. For textual tasks, several recent works [10, 205, 200, 299, 300, 301] have shown that fine-tuning language models such as GPT-3 and LLaMA with supervised instruction+response examples [8, 239, 258] enables them to respond to imperative requests and questions without task-specific training. Zero-shot generalization is promising not only for standard academic benchmarks, but – perhaps more-so – for creative, useful, and real-world queries that downstream users of language technologies are likely to make.

On the multimodal side, recent instruction-following vision-language models also provide a zero-shot interface. Given an image (or multiple images) and a query (e.g., “how


	Instruction	Instruction-Conditioned Caption	Human-Verified GPT4 Response
	Create a catchy title for a country song based upon the advice printed on the wheel cover.	There is a jeep parked on the street with the top down. There is a wheel cover on the spare tire in the back with an inscription that says, "Quit your Job, Buy a Ticket, Get a Tan, Fall in Love, Never Return."	"Tan Lines & Ticket Stubs: A Love Story Unbound" - A country song about leaving it all behind, finding love, and embracing the freedom of the open road.

Figure 10.1: An example from VisIT-Bench, featuring an image, a challenging instruction, an instruction-conditioned caption, and a human-verified GPT4 response. These elements are used for evaluating multimodal chatbots and updating a dynamic leaderboard.

many apples are in this image?" or "What is this?" or "Write a poem in the style of Robert Frost about this scene.") a textual response is provided. Recent works like OpenFlamingo [20, 22], LLaVA [276] and others [302, 303, 304, 305, 279], have implemented this interface with promising initial results. Although standard benchmarks like VQAv2 [306] and COCO captioning [117] are commonly used to assess performance, less is known about how models perform on broader, open-ended queries that resemble real-world user behavior. Evaluations of such queries typically rely on informal and qualitative approaches.

To support quantitative evaluation for this setting, we present VisIT-Bench (**Visual Instruction Benchmark**), a dynamic benchmark consisting of 592 challenging vision-language instructions. Each instance contains an instruction, input image(s), an instruction-conditioned caption (a human-crafted caption for the image(s)/instruction), and a human-verified reference (Figure 10.1). Instructions are image-contextual imperative requests or questions, e.g., for an image of pancakes, a user asks *"how can I cook this in a healthy way?"*. Different from existing zero-shot evaluations, many of the instructions focus on open-ended generation requests (e.g., *"write a poem..."* or *"what should I bring if I were to visit here?"*).

We created VisIT-Bench to cover a wide array of "instruction families". Our starting

point was a set of 70 “wish-list” tasks such as “home renovation” and “gardening tips” collected by the authors:¹ each requiring varied high-level skills from recognition to complex reasoning (Figure 10.2). We derived 25/70 instruction families from benchmark tasks such as Visual Question Answering (VQA) [307] and robust change captioning [308] into a chatbot-style format (this reformatting differs from prior work [276, 279, 302], as we focus on open-ended chatbot style responses.). Notably, 10 of these repurposed tasks involve multiple images.

We started with 10 images for each instruction family. Our annotators, guided by an example, create a new instruction, and provide a (permissively licensed) image. For each instruction, we next collect instruction-conditioned captions – unlike prior work [309, 310] these descriptions are designed not only to describe the image in general, but also, surface information targeted to the instruction. Finally, we use instruction-conditioned captions to generate a reference candidate output from GPT-4; an additional human verification step discards GPT-4 references deemed to be incorrect.

We conduct a large-scale empirical comparison of multimodal instruction-following models using VisIT-Bench (Section 10.4). We first gather predictions for each instance from 7 candidate models. Then, we collect 5K human judgements of output quality by pitting model outputs head-to-head, and (in a forced-choice setup) crowd-sourcing pairwise preference judgements. This analysis not only reveals significant differences between models (e.g., that LLaVA-13b [276] is generally preferred to Panda [305]), but also, that the human verified references in our corpus are preferred significantly more than the ones generated using multimodal models. We summarize head-to-head comparisons with two metrics: 1) Elo ratings [311, 312], which provide *relative* “skill” rating estimates encoding the probability that model A will be preferred to model B; and 2) win rate versus

¹We recognize that promising applications may not be covered by our set; and we don’t necessarily advocate for deploying models in all cases we cover – we hope VisIT-Bench can help to quantify shortcomings and risks.

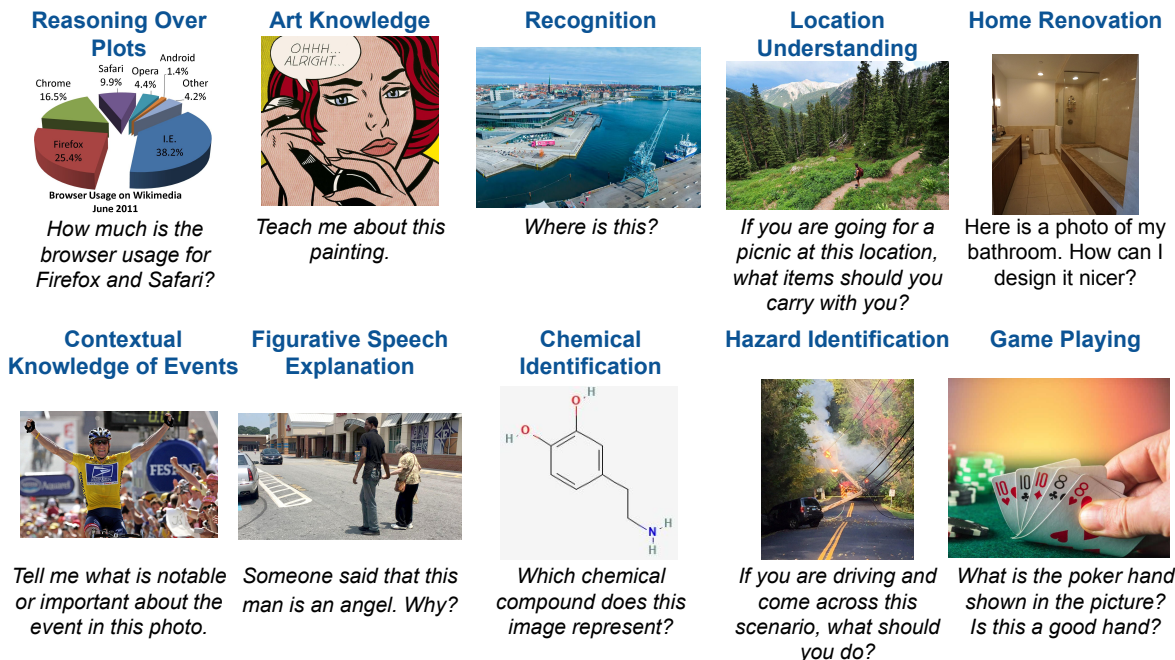


Figure 10.2: A sample from the 70 instruction families in VisIT-Bench representing tasks we envision instruction-following vision-language models *should* be able to follow.

our references, which provides an *absolute* metric. The best model according to human judgement is LLaMA-Adapter-v2 [303], yet it only wins in a pairwise setting against the reference in 27.4% of cases.

Finally, we design an automated evaluation for VisIT-Bench, utilizing GPT-4 to rank pairs of model responses based on factors like correctness, relevance, and fluency. Using the instruction-conditioned caption and the instruction, GPT-4 determines the better response between two options, expediting iteration compared to human preferences. We explore *reference-free* and *reference-backed* versions of this metric. Compared to various metrics (BLEU-4 [113], ROUGE-L [124], METEOR [114], CIDEr [115], and BERTScore [112]), our evaluation aligns best with human preferences. For example, it achieves a 94% agreement rate in the cases where all five annotators agree. See Figure 10.7 for a schematic of the process.

While it is difficult to *a priori* envision all possible scenarios under which more per-

	MultiInstruct [316]	Owl [279]	InstructBLIP [302]	M ³ IT [313]	LVLm [315]	GAVIE [314]	VisIT-Bench
Number of Models	1	5	3	4	8	5	10
Number of Skills Tested	9	6	13	13	47	16	70
Multiple-Images	✗	✓	✗	✗	✗	✗	✓
Video	✗	✗	✓	✓	✗	✗	✗
Multi-Turn Conversations	✓	✓	✓	✓	✓	✗	✗
Multilingual Conversations	✗	✓	✗	✓	✗	✗	✗
Instruction-conditioned Captions	✗	✗	✗	✗	✗	✗	✓
Chatbot-style Responses	✗	✗	✗	✗	✗	✗	✓
Dataset-specific Evaluation	✓	✓	✓	✓	✓	✗	✗
Human Evaluation	✗	✓	✗	✗	✓	✗	✓
Auto/GPT-4 Evaluation	✗	✓	✗	✓	✗	✓	✓
Win-rates*	✗	✓	✗	✓	✗	✓	✓
Elo Rating	✗	✗	✗	✗	✓	✗	✓

Table 10.1: Comparison with related works for evaluating instruction-following vision-language models. Win-rates* refers to the model win-rates against a reference output/model.

formant multimodal chatbots might be used, we hope VisIT-Bench can provide a path to improving vision-language models “in the wild.” Table 10.1 presents a summary of our contributions in comparison to the recent works [279, 302, 313, 314, 315, 316] in the evaluation of multimodal chatbots. We publicly release VisIT-Bench data, code, and automatic metrics to facilitate future model evaluations, available in <https://visit-bench.github.io/>.

10.2 VisIT-Bench: A Real-World Inspired VL Instruction Following Benchmark

VisIT-Bench was built to emulate real-world applications of multimodal models through image-text tasks, creating an extensive and practical benchmark. These tasks, or ‘instruction families’, are seen as key capabilities of a high-performing vision-and-language model. Although our selections are not exhaustive, they provide a broad basis for evaluating beyond academic benchmarks. We prioritize family coverage vs. number of instances-per-task. The final corpus, comprising 592 instances and 1,159 public images, can be found at VisIT-Bench Sheet and VisIT-Bench Sheet Multi-Images. VisIT-Bench in-

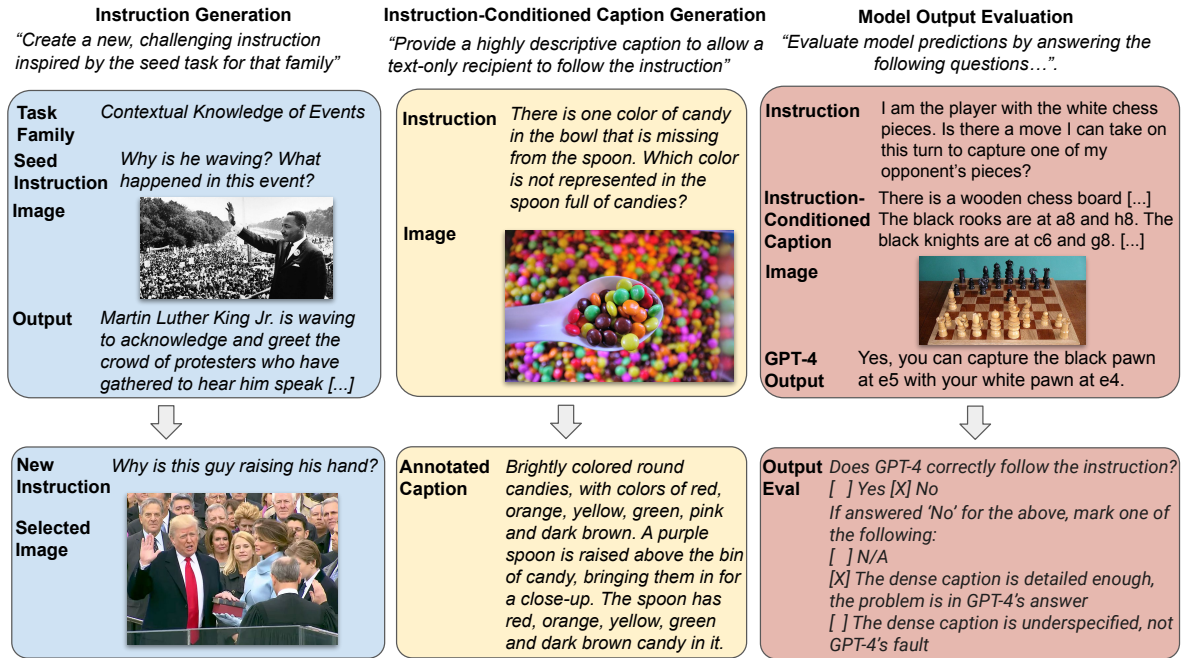


Figure 10.3: Data collection steps: (1) **Instruction Generation** - Annotators derive instances from a seed task, see Figure 10.3 (left). (2) **Caption Generation** - Rich *instruction-conditioned captions* are produced for GPT-4 references, shown in Figure 10.3 (middle). (3) **Model Evaluation** - GPT-4 responses are human-validated, illustrated in Figure 10.3 (right). Top blocks show rater instructions; bottom blocks depict outputs.

stances are either from 45 newly assembled instruction families or reformatted from 25 existing datasets. Notably, 10 instruction families cater to *multi-image* query scenarios (e.g., Figure 10.4).

10.2.1 Data Collection

The authors of this work perform an initial annotation step of curating instruction families. For each instruction family not derived from an existing task (45 out of 70), we designate a name for the family (e.g., “Contextual Knowledge of Events”) and identify an image-instruction pair that exemplifies the category, along with a sample response (“Martin Luther King Jr. is waving to acknowledge and greet the crowd of protesters

NLVR2 In this task you will be provided with two individual images i.e., BEFORE and AFTER. Please study them carefully and determine if the following sentence is correct (Answer "Yes"/"No"):

None of the dogs in these images are wearing clothing.

1  Two small chihuahuas, one with all white fur, and the other with a light brown coat of fur on its back, but white fur on its head and neck, are near a small pink pillow that has large white words proclaiming, "BE MINE" on it. The all white chihuahua is resting its head on the pillow, while the other one is laying behind it.

2  A small all white chihuahua is standing and looking straight at the camera. It's wearing a pink and white polka-dotted dress, over which is a pink shirt with some white text on it. Under the dog is a cement walkway, and there is grass to the dog's right and to the right of the walkway.

 **GPT-4** No. The second image has a dog wearing a pink and white polka-dotted dress and a pink shirt with white text on it.

Figure 10.4: An example of a multi-image instruction task from VisIT-Bench. Originally sourced from NLVR2 [317], this task aims to evaluate visual reasoning skills. The NLVR2 format includes a sentence to analyze, two images, and a binary response. Our adaptation adds a zero-shot instruction prompt, a instruction-conditioned caption for each image and a human-verified GPT-4 response. These enhancements, designed in a chatbot-style interaction, facilitate automatic evaluation of future chatbot responses to this instance.

[...]”). 10 sample families are in Figure 10.2.

The following steps are carried out in collaboration with crowdworkers, who receive an hourly wage of \$18. These steps are outlined in Figure 10.3: (1) taking the image/instruction example as a guiding seed task crowdworkers formulate a new instruction that examines the same instruction family (“instruction generation”); (2) crowdworkers create detailed image captions that describe the image and allow an entity, relying solely on this text, to interpret and execute the given instruction successfully (“instruction-conditioned caption generation”); (3) crowdworkers assess the correctness of GPT-4’s response to the instruction (“model output evaluation”). We further elaborate on these steps using human annotators below.

Re-formatting existing datasets. 25/70 instruction families (corresponding to $25 \times 10 = 250$ instances) are re-formatted versions of existing vision-language tasks.² This process involves re-formatting tasks into chatbot-style instruction/response versions. In re-formatting, we re-write instructions to retain the original task’s goal while maintaining the original images, see Figure 10.4. These repurposed tasks are integrated into our data collection process, ensuring uniformity between the chatbot-style answers in the full VisIT-Bench instances and the reinterpreted tasks.

Instruction Generation. Here, annotators create a new instance from the same instruction family as a given example, along with an instruction and corresponding image. For instance, in Figure 10.3 (left), the instruction family is “Contextual Knowledge of Events”, and the example instruction is “*Why is he waving? What happened in this event?*” alongside an image of Martin Luther King, Jr. To collect images, annotators were instructed to use Openverse (<https://openverse.org/>) for Creative Commons licenced images.

Instruction-Conditioned Caption Generation. Annotators are provided with the image and instruction, and are tasked to construct a caption that is rich enough to allow an entity, solely receiving the text they author, to follow the instruction. This caption will later facilitate GPT-4 reference candidate generation, and will be used for text-only auto-evaluation. We call these instructions *instruction-conditioned captions*. See Figure 10.3 (middle) for an example: an annotator doesn’t just mention the skittles and a spoon, but, given the query regarding specific colors, they indicate the exact colors in detail.

²Users of VisIT-Bench should also cite the original datasets.

Model Output Evaluation. The goal of this stage is to gather human-validated reference chatbot responses for each multimodal instruction query. We initially obtain response candidates from GPT-4 given the instruction and the instruction-conditioned caption. GPT4’s prompt is: *“Consider an image depicted by: $\langle caption \rangle$ ’. Now, briefly follow this instruction, and you can add a short explanation: $\langle instruction \rangle$ ’.* Response: This prompt is employed for both single and multiple image instances, with appropriate modifications for the latter. Then we verify each response with human annotators.³ If a response is marked incorrect, the annotator identifies whether the issue lies with the detail level of the instruction-conditioned captions or with GPT-4’s response itself. For VisIT-Bench, we discard any case marked as incorrect for either reason.⁴ An example is given in Figure 10.3 (right), where GPT-4’s candidate reference response aims to answer a question about a chess position (which it does so incorrectly, and thus, the instance is discarded).

10.2.2 Data Collection Annotation and Results

We conduct the data collection steps in Figure 10.3 using Amazon’s Mechanical Turk (MTurk) platform. Prior to annotating, each MTurk worker passed a qualification test, which involved five to ten sample tasks designed to assess their ability to generate high-quality annotations.

Our annotation results are summarized in Table 10.2. We measure the throughput of the collection and filtration pipeline. For single-image instances, our pipeline’s yield was 91.5% from the original candidate set. However, the success rate dropped to 63.0% in the more complex multi-image tasks, accompanied by an uptick in issues either in

³An alternate annotation scheme would have been to task annotators to write target responses from scratch. The rationale for using GPT-4 verification instead is derived from prior results that show promising human-machine collaboration of this form [318].

⁴The annotators are also tasked to screen for any offensive, unsound, or harmful advice present in the responses. We did not find or discard any instances.

Metrics	Overall	Single	Multi
GPT-4 Correct (%)	87.3	91.5	63.0
Problem in Caption (%)	4.0	3.6	6.0
Problem in GPT-4 (%)	7.7	3.8	30.0

Table 10.2: Human rating metrics for the VisIT-Bench dataset: overall, single-, and multi-image tasks.

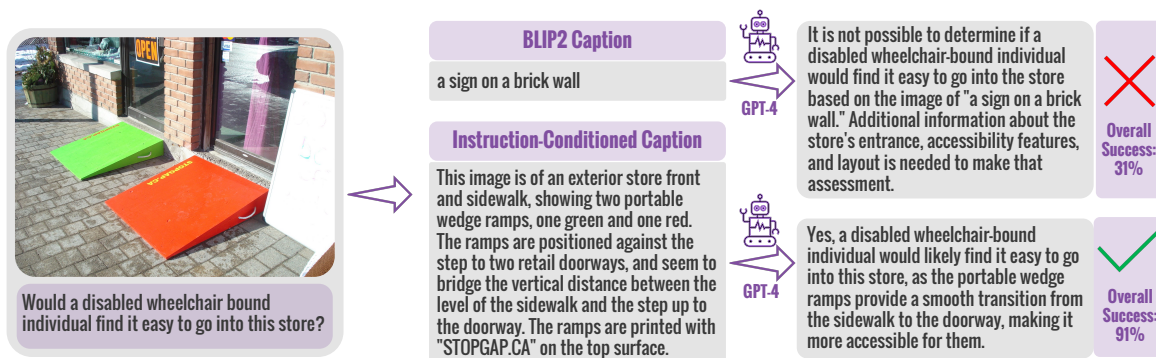


Figure 10.5: This experiment evaluates the value of instruction-conditioned captions in accurate instruction-following tasks. Given an image and instruction, GPT-4 generates responses using both a instruction-conditioned caption and a less detailed BLIP-2 [304] caption. The latter’s imprecision leads to an error, emphasizing the need for detailed, task-specific captions.

the captions (6.0%) or GPT-4’s responses (30.0%). This drop suggests that multi-image queries may pose a more difficult data collection challenge.

10.3 VisIT-Bench Analysis

We analyze the tasks, images, and instruction-conditioned captions of VisIT-Bench.

10.3.1 Are instruction-conditioned captions necessary?

To clarify the role of the instruction-conditioned captions we collect, we conducted an experiment covering 150 single-image instances. Instead of using our instruction-conditioned captions, we use BLIP2 [304] image captions, which is a state-of-the-art image captioning model. We extract image captions, and feed them to GPT-4 as detailed

earlier, to provide a text-based chatbot response. This process is depicted in Figure 10.5.

We manually evaluated whether the resulting output accurately followed the instructions. We find that while instruction-conditioned captions led to correct outputs in 91% of the cases, the success rate fell to 31% when using BLIP2 captions (Table 10.2). These results highlight the importance of instruction-conditioned captions in the construction of VisIT-Bench, and show that the instances in our dataset are sophisticated enough such that most are not solvable by using a simple Socratic model [319] baseline of caption \rightarrow LLM.

10.3.2 What skills are required for VisIT-Bench?

Following Wang et al. [320], for the VisIT-Bench instructions, we extract the most frequent root verbs and their direct nouns (a full plot is in Figure 10.6). The most common include: ‘*answer question*’, ‘*write story/poem*’, ‘*create title*’, etc. There’s also a long-tail of diverse requests that demand comprehension, commonsense, and cross-modal understanding, e.g., ‘*identifying objects*’ to ‘*need ingredient*’ to ‘*connect device*’. Additional qualitative examination reveals a range of underlying skills required ranging from ‘*emotion identification*’ to complex reasoning tasks such as ‘*paper folding*’.

10.3.3 What is contained in VisIT-Bench images?

We detect all the COCO [117] objects present in the images from our dataset using Yolov5-L [321]; The most common detected objects in VisIT-Bench are “person” (~ 900 detections), chair, and car (~ 100). But, a long tail of rarer objects exists as well. Overall, to perform well at VisIT-Bench, a model must account for a broad range of scenes and objects.

automatic evaluation on VisIT-Bench in Section 10.4.3, that can be scaled and improved given new and improved models. Finally, we establish the trustworthiness of our automatic evaluation method by performing agreement analysis with the human judgments in Section 10.4.3.

10.4.1 Models

We evaluate LLaVA-13B [276], InstructBLIP-13B [302], MiniGPT4-7B [323], mPLUG-Owl-7B [279], LlamaAdapter-v2-7B [303], PandaGPT-13B [305], VisualChatGPT [173], Multimodal GPT [297], OpenFlamingo v1 [20], Otter v1 [295], Lynx [324] and idefics [284]. For the execution-based VisualChatGPT [173], we implement a chat window for each sample, hold inputs and intermediate chains of thoughts and actions in memory, and feed the images and the instruction sequentially. For OpenFlamingo [20] and Otter [295], we feed the image(s) and the instruction in an interleaved format. For the others, we feed the image to the vision feature extractor and feed the instruction as a prompt to the text encoder.⁵

10.4.2 Human Evaluation

We collect 5K pairwise human preference judgements across an initial set of 6 models and the human-verified references. For 1K uniformly randomly sampled tuples of (query, model A, model B), we collect 5 crowdworker judgements each. Preferences are collected in a “forced choice” setting, annotators are instructed to decide based on accuracy, helpfulness, and detail. We summarize the results with two metrics:

Relative metric: Elo We follow Zheng et al. [312] and compute Elo ratings, treating

⁵Following the authors’ instructions, we run all models using default settings to obtain the best possible responses. We include specific samples for reproducibility. We acknowledge hyperparameter impact and are willing to reassess submissions to VisIT-Bench if conditions were sub-optimal.

	Model	Elo	matches	Win-rate vs. reference (w/ # ratings)
Single Image	Human Verified GPT-4 Reference	1223	1439	–
	LLaVA (13B)	1085	1462	26.23% (n=244)
	LlamaAdapter-v2 (7B)	1061	1507	27.41% (n=259)
	mPLUG-Owl (7B)	995	1345	14.95% (n=214)
	InstructBLIP (13B)	957	1315	12.37% (n=194)
	MiniGPT-4 (7B)	893	1513	14.72% (n=299)
	PandaGPT (13B)	786	1441	10.48% (n=229)
Multiple Images	Human Verified GPT-4 Reference	1193	210	–
	mPLUG-Owl	997	190	15.38% (n=78)
	Otter v1	917	147	3.17% (n=63)
	OpenFlamingo v1	893	171	4.35% (n=69)

Table 10.3: Human scoring results for the models, shown as both an ELO rating and win-rate against the reference. In total, this summarizes 5.0K pairwise human judgments. matches column indicates the number of total matches in which a particular model participates. Win-rate vs. reference indicates the win-rate of a model against the reference outputs.

each pairwise human judgement as a “match.”⁶ The difference between the Elo ratings of two different models provides an estimate for the win probability when pitting model A vs. model B.

Absolute metric: Win rate vs. reference. We provide a win-rate vs. the human-verified reference. We use the 1.4K pairwise human judgments where one of A or B is the reference. We report the percent of cases where the human judge prefers the output from that model vs. the human-verified GPT-4 reference output. Because we do not allow for ties in our forced-choice setup, if the annotator believes the responses are of equal quality, they choose one arbitrarily.

Results Table 10.3 contains the Elo and win-rate vs. reference. In terms of Elo, the Human Verified GPT-4 reference achieves a higher rating than all alternatives, validating the quality of our reference set: concretely, for our Elo settings, the reference (Elo =1223) has an estimated win-rate over one of the best performing models, LLaVA, (Elo =1085) of 69%, and an estimated win rate of 93% against the lowest performing model

⁶We use the following code/hyperparameters for Elo ratings: https://github.com/lm-sys/FastChat/blob/main/fastchat/serve/monitor/elo_analysis.py

Category	Model	Elo	# Matches	Win vs. Reference (w/ # ratings)
Single Image	Human Verified GPT-4 Reference	1,382	5,880	—
	LLaVA-Plus (13B)	1,203	678	35.07% (n=134)
	LLaVA (13B)	1,095	5,420	18.53% (n=475)
	mPLUG-Owl (7B)	1,087	5,440	15.83% (n=480)
	LlamaAdapter-v2 (7B)	1,066	5,469	14.14% (n=488)
	Lynx(8B)	1,037	787	11.43% (n=140)
	idefics (9B)	1,020	794	9.72% (n=144)
	InstructBLIP (13B)	1,000	5,469	14.12% (n=503)
	Otter v1 (9B)	962	5,443	7.01% (n=499)
	VisualGPT (Da Vinci 003)	941	5,437	1.57% (n=510)
	MiniGPT-4 (7B)	926	5,448	3.36% (n=506)
	Octopus V2 (9B)	925	790	8.90% (n=146)
	OpenFlamingo V1 (9B)	851	5,479	2.95% (n=509)
	PandaGPT (13B)	775	5,465	2.70% (n=519)
	Multimodal GPT	731	5,471	0.19% (n=527)
Multiple Images	Human Verified GPT-4 Reference	1,192	180	-
	mPLUG-Owl	995	180	6.67% (n=60)
	Otter v1	911	180	1.69% (n=59)
	OpenFlamingo v1	902	180	1.67% (n=60)

Table 10.4: Current reference-free Elo rankings as of September 25th, 2023. In total, these rankings summarize 31,735 “matches” between models; each match consists of 2 queries to GPT-4. Because VisIT-Bench is dynamic, these rankings are updated as more models are added to the leaderboard, and more pairs of models are evaluated head-to-head for more instances.

in this setup, PandaGPT (Elo =786). This result can partly be explained by the training process of the underlying models: The improved performance of LLaVA (13B) might be attributed to its fine-tuning process, which utilized 150K instruction-tuning data that is rich in both diversity and quality. Interestingly, despite achieving a slightly lower Elo (the computation of which is based on *all* head-to-head “matches”, rather than just ones against the human reference), LlamaAdapter-v2 (7B) wins with the highest rate against the reference. However, the complexity and variety of models and tasks in VisIT-Bench makes it challenging to definitively pinpoint the factors influencing performance. While we make a preliminary attempt to unravel these intricacies in Section 10.4.3, a comprehensive understanding will necessitate more nuanced and extensive future research.

10.4.3 Automatic Evaluation and Leaderboard

Because it is costly to gather human pairwise preference judgements for new model submissions, to support faster model development, we seek an automatic evaluation procedure that produces high correlation with our human evaluation setup.

Automatic evaluation metric candidates. We consider several existing reference-backed evaluation metrics: BLEU-4 [113], ROUGE-L [124], METEOR [114], CIDEr [115], and BERTScore [112], we use the RoBERTa-Large english version [325], treating the human-verified GPT-4 reference as the evaluation reference. We additionally report two baseline metrics: random, which assigns a random score without accounting for the candidate, and length, which assigns a score equal to the number of non-whitespace tokens in the candidate. Beyond existing metrics and baselines, following the recent line of work utilizing API-accessed LLMs with a prompt for automatic evaluation [299, 326], we consider two GPT-4 [205] backed evaluation metrics.

Specifically, we provide the LLM with: 1) a system prompt describing the desired evaluation behavior; 2) the instruction-conditioned caption for the image; 3) the instruction to be followed; and 4) two candidate generations dubbed “Response A” and “Response B”. We also consider a reference-backed version where the human-verified reference is provided as well. To mitigate potential biases in “A” and “B” positioning, for all pairs of candidates, we run two queries covering both possible orderings. Our prompt encourages the model to think step-by-step so that its chain-of-thought process is made explicit [158, 327]. Despite strongly encouraging the model to select between the two references in a forced-choice setup, it sometimes refuses and outputs “tie” which we account for later. We call the reference-free version of this metric “GPT4-no-ref”, and the reference-backed version of this metric “GPT4-ref”.

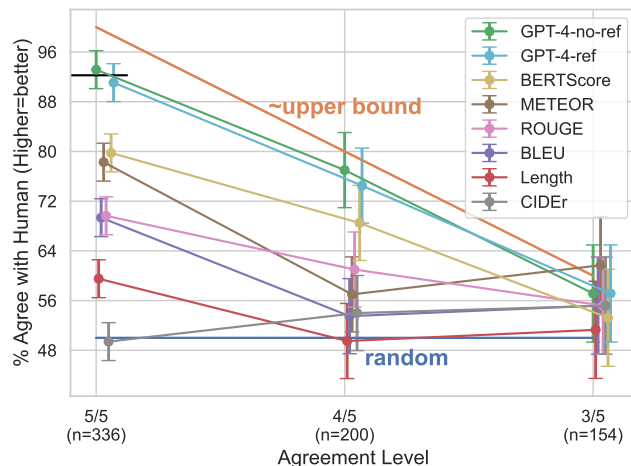


Figure 10.8: Correlations between evaluation metrics and human preferences are ranked in performance order, with our reference free evaluation (GPT-4-no-ref) showing the strongest alignment. Bottom line: random chance (50%), top line: upper performance bound.

Evaluating evaluation metrics. We measure the correlation between the candidate metrics and human judgements using a pairwise framework. Specifically, we use a subset of the 5K pairwise human judgements in Section 10.4.2. For 690 pairwise instances where both candidate instances are model-generated (rather than human-verified references), we have 5 pairwise judgements from crowd-workers. For 336 pairs, there is 5/5 agreement, for 200 pairs, there is 4/5 agreement, and for 154 pairs, there is 3/5 agreement. For each metric, we measure the percent of time the metric is able to accurately reconstruct a majority vote judgement from the 5 crowdworkers. The newly proposed GPT-4 based metrics sometimes outputs “tie” (this happens in 10-15% of cases overall) – for fair comparison with the other metrics in forced choice setting, we randomly choose one of the two options when GPT-4 reports a tie.

The results are in Figure 10.8, with GPT-4-no-ref best aligns with human correlation. The best performing metric is our newly proposed GPT-4 based metric, which accurately reconstructs majority-vote pairwise human judgments better than alternatives ($p < .05$;

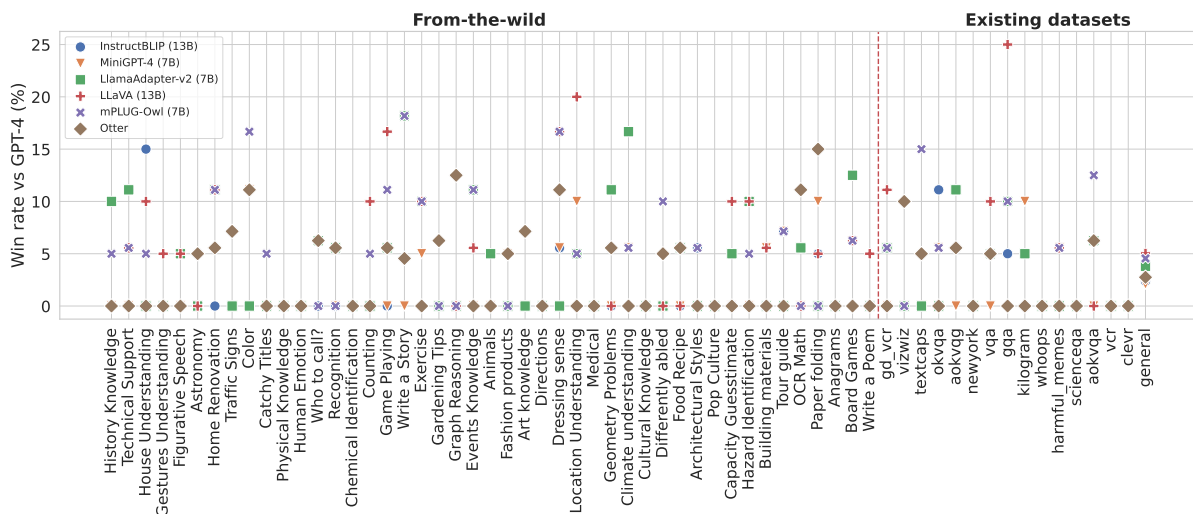


Figure 10.9: Reference-free assesment win rate vs. human-verified GPT4 response for each instruction category. Axes: win rate (Y), instruction categories (X). Categories are from-the-wild or existing datasets. VisIT-Bench facilitates analysis of diverse instruction tuning tasks.

binomial proportion CI nonoverlapping). For example, for instances where 5/5 annotators agree, GPT4-no-ref, with no reference, accurately reconstructs human judgment 93% of the time, whereas the next best metrics BERTScore/METEOR/ROUGE-L reconstruct accurately 80%/78%/70% of the time; among the metrics we consider, these are reasonable options for static/offline evaluation without relying on OpenAI API access, especially when compared to our length baseline metric, which achieves only 60%. Notably, the reference-backed version of the newly proposed GPT-4 based metric achieves comparable (but slightly worse) performance compared to the reference-free version. Thus, we adopt the reference-free version, which additionally enables us to place the references themselves into the Elo setup, because they are not used in the prompts.

Per-category results. In Figure 10.9, we plot the win-rate vs reference for the models across all the single-image instruction families. We find that there is no model that performs the best and worst across all the instruction families. Thus, VisIT-Bench aids in highlighting the strengths and weaknesses of the instruction-following models

along various real-world use-cases.

10.5 Related Work

Multimodal Models for Image-Text Understanding Recently, the field of machine learning has experienced a rapid proliferation of new models which can perform various image-text tasks [22, 276, 302, 304, 305, 328]. This growth has been driven by several factors, including the emergence of large-scale multimodal datasets (e.g. LAION-5B [96], Multimodal C4 [19]), improved software and hardware frameworks, and advances in modality-specific models such as language models (e.g., [239]). Our work specifically evaluates models which can generate textual outputs, given one or more images, and text. Recent examples of such models include LLaVA [276], mPLUG-Owl [279], InstructBLIP, LLaMA-Adapter, Flamingo [22] and OpenFlamingo [20], PandaGPT [305], and GPT-4 [205] (which reports multimodal capabilities but has not yet seen a release of the multimodal variant).

Instruction Following “Instruction-following” is an emerging paradigm for training models via language, where instead of being trained to complete only a single, fixed task (such as image classification or captioning), models are trained to follow textual instructions that describe an arbitrary task, with the aim of generalizing to novel instructions. Examples of instruction-following models include Alpaca [300], LLaMA-Adapter [303], Koala [329], InstructBLIP [302], LLaVA [276], and mPLUG-owl [279]. As the downstream capabilities of these models are influenced by the quality of the training dataset, there has also been extensive work on developing instruction-following datasets [276, 320, 330, 331, 332].

To build these models, two broad approaches have been shown to be effective. One

approach focuses on leveraging existing pretrained task-specific tools such as image captioners [304], object detectors [204] and text-to-image generators [4] by either creating multimodal prompt interfaces [173, 193] or by executing LLM-generated programs [192, 196, 197]. The other approach [20, 254, 276, 279, 295, 297, 303] focuses on building a single pretrained model that can follow instructions by supervised finetuning on multimodal vision-language data.

Despite the success of both these approaches on the existing vision-language datasets e.g., VQA, GQA, Image Captioning [117, 307, 333], there is a lack of a high-quality benchmarking dataset for multimodal instruction-following tasks that reliably replicates the way in which humans would interact with multimodal chatbots in the wild. Similar to the image-text models discussed above, many instruction-following models have been released directly as open-source without undergoing peer review or thorough evaluation. As a result, the effectiveness of these models for many tasks is not well-understood.

Benchmarks for Machine Learning High-quality evaluation datasets have served both to (re)assess, and to accelerate, progress on many machine learning tasks [334]. For example, our work draws particularly from the fields of computer vision and natural language processing, where benchmarking datasets have been critical drivers of progress. On the vision side, datasets such as ImageNet [335] and CIFAR [336] have proven to be critical yardsticks of progress. On the language side, benchmarks such as SQuAD [337], SST [62], GLUE/SuperGLUE [14, 338] and more [339, 340] seen wide use. Recent work has indicated that improvements on these high-quality benchmark datasets is *not* the result of overfitting, and is a reliable indicator of genuine progress beyond the benchmark data [341, 342, 343, 344].

However, high-quality benchmarking datasets and evaluation methods do not yet exist for multimodal instruction-following. As a result, it is difficult to assess progress in

this direction, which both reduces the field’s ability to identify true breakthroughs and increases vulnerability to potential pitfalls of evaluation that have hampered progress in other areas of machine learning [334, 345].

Bibliography

- [1] M. Just, S. Newman, T. Keller, A. McEleney, and P. Carpenter, *Imagery in sentence comprehension: an fmri study*, in *NeuroImage*, vol. 21, pp. 112–124, 2004.
- [2] S. F. Popham, A. G. Huth, N. Y. Bilenko, F. Deniz, J. S. Gao, A. O. Nunez-Elizalde, and J. L. Gallant, *Visual and linguistic semantic representations are aligned at the border of human visual cortex*, *Nature neuroscience* **24** (November, 2021) 1628—1636.
- [3] M. Sadoski and A. Paivio, *A dual coding view of imagery and verbal processes in reading comprehension.*, in *Theoretical Models and Processes of Reading*, pp. 582–601, International Reading Association, 1994.
- [4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-resolution image synthesis with latent diffusion models*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.
- [5] L. Midjourney, 2022.
- [6] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. Fleet, and M. Norouzi, *Photorealistic text-to-image diffusion models with deep language understanding*, *ArXiv* **abs/2205.11487** (2022).
- [7] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, *Hierarchical text-conditional image generation with clip latents*, *ArXiv* **abs/2204.06125** (2022).
- [8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et. al.*, *Language models are few-shot learners*, *Advances in neural information processing systems* **33** (2020) 1877–1901.
- [9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Language models are unsupervised multitask learners*, 2019.

- [10] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et. al.*, *Training language models to follow instructions with human feedback*, *Advances in Neural Information Processing Systems* **35** (2022) 27730–27744.
- [11] Y. Lu, W. Zhu, X. Wang, M. Eckstein, and W. Y. Wang, *Imagination-augmented natural language understanding*, in *NAACL*, (Seattle, United States), pp. 4392–4402, Association for Computational Linguistics, July, 2022.
- [12] W. Zhu, A. Yan, Y. Lu, W. Xu, X. Wang, M. Eckstein, and W. Y. Wang, *Visualize before you write: Imagination-guided open-ended text generation*, in *Findings of the Association for Computational Linguistics: EACL 2023* (A. Vlachos and I. Augenstein, eds.), (Dubrovnik, Croatia), pp. 78–92, Association for Computational Linguistics, May, 2023.
- [13] W. Zhu, X. Wang, A. Yan, M. Eckstein, and W. Y. Wang, *ImaginE: An imagination-based automatic evaluation metric for natural language generation*, in *Findings of the Association for Computational Linguistics: EACL 2023*, (Dubrovnik, Croatia), pp. 93–105, Association for Computational Linguistics, May, 2023.
- [14] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, *GLUE: A multi-task benchmark and analysis platform for natural language understanding*, in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, (Brussels, Belgium), pp. 353–355, Association for Computational Linguistics, Nov., 2018.
- [15] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, *Swag: A large-scale adversarial dataset for grounded commonsense inference*, in *EMNLP*, 2018.
- [16] W. Zhu, X. Wang, Y. Lu, T.-J. Fu, X. E. Wang, M. P. Eckstein, and W. Y. Wang, *Collaborative generative AI: Integrating GPT-k for efficient editing in text-to-image generation*, in *The 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- [17] W. Feng, W. Zhu, T.-J. Fu, V. Jampani, A. R. Akula, X. He, S. Basu, X. E. Wang, and W. Y. Wang, *LayoutGPT: Compositional visual planning and generation with large language models*, in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [18] R. Schumann, W. Zhu, W. Feng, T.-J. Fu, S. Riezler, and W. Y. Wang, *Velma: Verbalization embodiment of llm agents for vision and language navigation in street view*, in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

- [19] W. Zhu, J. Hessel, A. Awadalla, S. Y. Gadre, J. Dodge, A. Fang, Y. Yu, L. Schmidt, W. Y. Wang, and Y. Choi, *Multimodal c4: An open, billion-scale corpus of images interleaved with text*, in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.
- [20] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa, *et. al.*, *Openflamingo: An open-source framework for training large autoregressive vision-language models*, *arXiv preprint arXiv:2308.01390* (2023).
- [21] Y. Bitton, H. Bansal, J. Hessel, R. Shao, W. Zhu, A. Awadalla, J. Gardner, R. Taori, and L. Schmidt, *Visit-bench: A benchmark for vision-language instruction following inspired by real-world use*, in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.
- [22] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, *Flamingo: a visual language model for few-shot learning*, vol. 35, pp. 23716–23736, 2022.
- [23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et. al.*, *Learning transferable visual models from natural language supervision*, in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.
- [24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of deep bidirectional transformers for language understanding*, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June, 2019.
- [25] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, *Albert: A lite bert for self-supervised learning of language representations*, .
- [26] L. Zhuang, L. Wayne, S. Ya, and Z. Jun, *A robustly optimized BERT pre-training approach with post-training*, in *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, (Huhhot, China), pp. 1218–1227, Chinese Information Processing Society of China, Aug., 2021.
- [27] H. Tan and M. Bansal, *Vokenization: Improving language understanding with contextualized, visual-grounded supervision*, in *Proceedings of the 2020 Conference*

- on *Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 2066–2080, Association for Computational Linguistics, Nov., 2020.
- [28] D. Elliott, S. Frank, K. Sima'an, and L. Specia, *Multi30k: Multilingual english-german image descriptions*, in *Proceedings of the 5th Workshop on Vision and Language, hosted by the 54th Annual Meeting of the Association for Computational Linguistics, VL@ACL 2016, August 12, Berlin, Germany*, pp. 70–74, Association for Computational Linguistics (ACL), Aug., 2016. 5th Workshop on Vision and Language, VL 2016 ; Conference date: 12-08-2016 Through 12-08-2016.
- [29] M. Grubinger, P. D. Clough, H. Müller, and T. Deselaers, *The iapr tc-12 benchmark: A new evaluation resource for visual information systems*, 2006.
- [30] R. Funaki and H. Nakayama, *Image-mediated learning for zero-shot cross-lingual document retrieval*, in *EMNLP*, 2015.
- [31] J. Gu, J. Cai, S. R. Joty, L. Niu, and G. Wang, *Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models*, 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)* 7181–7189.
- [32] G. Christie, A. Laddha, A. Agrawal, S. Antol, Y. Goyal, K. Kochersberger, and D. Batra, *Resolving language and vision ambiguities together: Joint segmentation & prepositional attachment resolution in captioned scenes*, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 1493–1503, Association for Computational Linguistics, Nov., 2016.
- [33] H. Shi, J. Mao, K. Gimpel, and K. Livescu, *Visually grounded neural syntax acquisition*, in *ACL*, 2019.
- [34] N. Xie, F. Lai, D. Doran, and A. Kadav, *Visual entailment: A novel task for fine-grained image understanding*, *ArXiv abs/1901.06706* (2019).
- [35] D. Kiela, I. Vulic, and S. Clark, *Visual bilingual lexicon induction with transferred convnet features*, in *EMNLP*, 2015.
- [36] I. Vulic, D. Kiela, S. Clark, and M.-F. Moens, *Multi-modal representations for improved bilingual lexicon learning*, in *ACL*, 2016.
- [37] X. Liu, D. Yin, Y. Feng, and D. Zhao, *Things not written in text: Exploring spatial commonsense from visual signals*, 2022.
- [38] P. Bordes, E. Zablocki, L. Soulier, B. Piwowarski, and P. Gallinari, *Incorporating visual semantics into sentence representations within a grounded space*, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), (Hong Kong, China), pp. 696–707, Association for Computational Linguistics, Nov., 2019.

- [39] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, *Uniter: Universal image-text representation learning*, in *ECCV*, 2020.
- [40] G. Collell, T. Zhang, and M.-F. Moens, *Imagined visual representations as multimodal embeddings*, in *AAAI*, 2017.
- [41] D. Kiela, A. Conneau, A. Jabri, and M. Nickel, *Learning visually grounded sentence representations*, in *NAACL*, 2018.
- [42] A. Lazaridou, N. T. Pham, and M. Baroni, *Combining language and vision with a multimodal skip-gram model*, in *NAACL*, 2015.
- [43] G. Li, N. Duan, Y. Fang, M. Gong, and D. Jiang, *Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training*, in *AAAI*, pp. 11336–11344, AAAI Press, 2020.
- [44] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, *Visualbert: A simple and performant baseline for vision and language*, *ArXiv abs/1908.03557* (2019).
- [45] J. Lu, D. Batra, D. Parikh, and S. Lee, *Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks*, in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 13–23, 2019.
- [46] H. Luo, L. Ji, B. Shi, H. Huang, N. Duan, T. Li, X. Chen, and M. Zhou, *Univilm: A unified video and language pre-training model for multimodal understanding and generation*, *ArXiv abs/2002.06353* (2020).
- [47] C. Sun, A. Myers, C. Vondrick, K. P. Murphy, and C. Schmid, *Videobert: A joint model for video and language representation learning*, *ICCV* (2019) 7463–7472.
- [48] Q. Long, M. Wang, and L. Li, *Generative imagination elevates machine translation*, in *NAACL*, (Online), pp. 5738–5748, Association for Computational Linguistics, June, 2021.
- [49] L. Xu, H. Hu, X. Zhang, L. Li, C. Cao, Y. Li, Y. Xu, K. Sun, D. Yu, C. Yu, Y. Tian, Q. Dong, W. Liu, B. Shi, Y. Cui, J. Li, J. Zeng, R. Wang, W. Xie, Y. Li, Y. Patterson, Z. Tian, Y. Zhang, H. Zhou, S. Liu, Z. Zhao, Q. Zhao, C. Yue, X. Zhang, Z. Yang, K. Richardson, and Z. Lan, *CLUE: A Chinese language understanding evaluation benchmark*, in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)),

- pp. 4762–4772, International Committee on Computational Linguistics, Dec., 2020.
- [50] L. Fei-Fei, R. Fergus, and P. Perona, *One-shot learning of object categories*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28** (2006) 594–611.
- [51] M. Fink, *Object classification from a single example utilizing class relevance metrics*, in *NIPS*, 2004.
- [52] D. Sui, Y. Chen, B. Mao, D. Qiu, K. Liu, and J. Zhao, *Knowledge guided metric learning for few-shot text classification*, in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Online), pp. 3266–3271, Association for Computational Linguistics, June, 2021.
- [53] T. Bansal, R. Jha, and A. McCallum, *Learning to few-shot learn across diverse natural language classification tasks*, in *COLING*, 2020.
- [54] R. Geng, B. Li, Y. Li, X. Zhu, P. Jian, and J. Sun, *Induction networks for few-shot text classification*, in *EMNLP*, 2019.
- [55] C. Han, Z. Fan, D. Zhang, M. Qiu, M. Gao, and A. Zhou, *Meta-learning adversarial domain adaptation network for few-shot text classification*, in *FINDINGS*, 2021.
- [56] S. Murty, T. B. Hashimoto, and C. D. Manning, *Dreca: A general task augmentation strategy for few-shot natural language inference*, in *NAACL*, 2021.
- [57] J. Wei, C. Huang, S. Vosoughi, Y. Cheng, and S. Xu, *Few-shot text classification with triplet networks, data augmentation, and curriculum learning*, in *NAACL*, 2021.
- [58] M. Tsimpoukelli, J. L. Menick, S. Cabi, S. M. A. Eslami, O. Vinyals, and F. Hill, *Multimodal few-shot learning with frozen language models*, in *Advances in Neural Information Processing Systems*, vol. 34, pp. 200–212, Curran Associates, Inc., 2021.
- [59] P. Esser, R. Rombach, and B. Ommer, *Taming transformers for high-resolution image synthesis*, in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 12873–12883, Computer Vision Foundation / IEEE, 2021.
- [60] K. Crowson, S. R. Biderman, D. Kornis, D. Stander, E. Hallahan, L. Castricato, and E. Raff, *Vqgan-clip: Open domain image generation and editing with natural language guidance*, in *ECCV*, 2022.

- [61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.
- [62] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, *Recursive deep models for semantic compositionality over a sentiment treebank*, in *EMNLP*, 2013.
- [63] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, *SQuAD: 100,000+ questions for machine comprehension of text*, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 2383–2392, Association for Computational Linguistics, Nov., 2016.
- [64] S. Iyer, N. Dandekar, and K. Csernai, *First quora dataset release: Question pairs*, January, 2017.
- [65] A. Williams, N. Nangia, and S. Bowman, *A broad-coverage challenge corpus for sentence understanding through inference*, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 1112–1122, Association for Computational Linguistics, June, 2018.
- [66] W. B. Dolan and C. Brockett, *Automatically constructing a corpus of sentential paraphrases*, in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.
- [67] E. Agirre, L. M´arquez, and R. Wicentowski, *Semantic textual similarity benchmark*, in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, (Prague, Czech Republic), Computational Linguistics, June, 2007.
- [68] L. B. Gambrell and R. J. Bales, *Mental imagery and the comprehension-monitoring performance of fourth-and fifth-grade poor readers*, *Reading Research Quarterly* (1986) 454–464.
- [69] V. L. Joffe, K. Cain, and N. Marić, *Comprehension problems in children with specific language impairment: does mental imagery training help?*, *International Journal of Language & Communication Disorders* **42** (2007), no. 6 648–664.
- [70] L. B. Gambrell and P. S. Koskinen, *Imagery: A strategy for enhancing comprehension*, *Comprehension instruction: Research-based best practices* (2002) 305–318.

- [71] J. Guan, Y. Wang, and M. Huang, *Story ending generation with incremental encoding and commonsense knowledge*, in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 6473–6480, AAAI Press, 2019.
- [72] A. Fan, M. Lewis, and Y. Dauphin, *Hierarchical neural story generation*, in *ACL*, (Melbourne, Australia), pp. 889–898, Association for Computational Linguistics, July, 2018.
- [73] S. Goldfarb-Tarrant, T. Chakrabarty, R. Weischedel, and N. Peng, *Content planning for neural story generation with aristotelian rescoring*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 4319–4338, Association for Computational Linguistics, Nov., 2020.
- [74] Y. Su, T. Lan, Y. Wang, D. Yogatama, L. Kong, and N. Collier, *A contrastive framework for neural text generation*, in *NeurIPS*, 2022.
- [75] B. Swanson, K. Mathewson, B. Pietrzak, S. Chen, and M. Dinulescu, *Story centaur: Large language model few shot learning as a creative writing tool*, in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, (Online), pp. 244–256, Association for Computational Linguistics, Apr., 2021.
- [76] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young, *Agenda-based user simulation for bootstrapping a POMDP dialogue system*, in *NAACL (Companion Volume, Short Papers)*, (Rochester, New York), pp. 149–152, Association for Computational Linguistics, Apr., 2007.
- [77] Z. Wei, Q. Liu, B. Peng, H. Tou, T. Chen, X. Huang, K.-f. Wong, and X. Dai, *Task-oriented dialogue system for automatic diagnosis*, in *ACL (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 201–207, Association for Computational Linguistics, July, 2018.
- [78] T.-H. Wen, M. Gašić, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young, *Semantically conditioned LSTM-based natural language generation for spoken dialogue systems*, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 1711–1721, Association for Computational Linguistics, Sept., 2015.
- [79] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gašić, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, *A network-based end-to-end trainable task-oriented*

- dialogue system*, in *ACL*, (Valencia, Spain), pp. 438–449, Association for Computational Linguistics, Apr., 2017.
- [80] J. Wu, I. Harris, and H. Zhao, *Spoken language understanding for task-oriented dialogue systems with augmented memory networks*, in *NAACL*, (Online), pp. 797–806, Association for Computational Linguistics, June, 2021.
- [81] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, *Zero-shot text-to-image generation*, in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831, PMLR, 18–24 Jul, 2021.
- [82] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, *Opa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework*, in *International Conference on Machine Learning*, pp. 23318–23340, PMLR, 2022.
- [83] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, *BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, in *ACL*, (Online), pp. 7871–7880, Association for Computational Linguistics, July, 2020.
- [84] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, *Exploring the limits of transfer learning with a unified text-to-text transformer*, *Journal of Machine Learning Research* **21** (2020), no. 140 1–67.
- [85] D. Ippolito, D. Grangier, D. Eck, and C. Callison-Burch, *Toward better storylines with sentence-level language models*, in *ACL*, (Online), pp. 7472–7478, Association for Computational Linguistics, July, 2020.
- [86] A. See, A. Pappu, R. Saxena, A. Yerukola, and C. D. Manning, *Do massively pretrained language models make better storytellers?*, in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, (Hong Kong, China), pp. 843–861, Association for Computational Linguistics, Nov., 2019.
- [87] J. Guan, F. Huang, Z. Zhao, X. Zhu, and M. Huang, *A knowledge-enhanced pretraining model for commonsense story generation*, *ArXiv abs/2001.05139* (2020).
- [88] L. Yao, N. Peng, R. M. Weischedel, K. Knight, D. Zhao, and R. Yan, *Plan-and-write: Towards better automatic storytelling*, in *AAAI*, 2019.
- [89] Y. Liu, Y. Wan, L. He, H. Peng, and P. S. Yu, *Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning*, in *AAAI*, 2021.

- [90] W. Zhou, D.-H. Lee, R. K. Selvam, S. Lee, and X. R. 0001, *Pre-training text-to-text transformers for concept-centric common sense*, in *ICLR 2021*, OpenReview.net, 2021.
- [91] P. Wang, J. Zamora, J. Liu, F. Ilievski, M. Chen, and X. Ren, *Contextualized scene imagination for generative commonsense reasoning*, in *ICLR*, 2022.
- [92] J. Cho, J. Lei, H. Tan, and M. Bansal, *Unifying vision-and-language tasks via text generation*, in *ICML*, vol. 139 of *Proceedings of Machine Learning Research*, pp. 1931–1942, PMLR, 18–24 Jul, 2021.
- [93] Y. Su, T. Lan, Y. Liu, F. Liu, D. Yogatama, Y. Wang, L. Kong, and N. Collier, *Language models can see: Plugging visual controls in text generation*, *ArXiv abs/2205.02655* (2022).
- [94] P. Yang, B. Chen, P. Zhang, and X. Sun, *Visual agreement regularized training for multi-modal machine translation*, in *AAAI*, vol. 34, pp. 9418–9425, 2020.
- [95] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, (Cham), pp. 234–241, Springer International Publishing, 2015.
- [96] C. Schuhmann, R. Beaumont, C. W. Gordon, R. Wightman, T. Coombes, A. Katta, C. Mullis, P. Schramowski, S. R. Kundurthy, K. Crowson, *et. al.*, *Laion-5b: An open large-scale dataset for training next generation image-text models*, in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.
- [97] R. Mokady, A. Hertz, and A. H. Bermano, *Clipcap: Clip prefix for image captioning*, *arXiv preprint arXiv:2111.09734* (2021).
- [98] A. van den Oord, Y. Li, and O. Vinyals, *Representation learning with contrastive predictive coding*, in *ArXiv*, vol. abs/1807.03748, 2018.
- [99] A. Yan, Z. He, X. Lu, J. Du, E. Chang, A. Gentili, J. McAuley, and C.-N. Hsu, *Weakly supervised contrastive learning for chest X-ray report generation*, in *Findings of the Association for Computational Linguistics: EMNLP 2021*, (Punta Cana, Dominican Republic), pp. 4009–4015, Association for Computational Linguistics, Nov., 2021.
- [100] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, *Activitynet: A large-scale video benchmark for human activity understanding*, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–970, 2015.

- [101] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, *HellaSwag: Can a machine really finish your sentence?*, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 4791–4800, Association for Computational Linguistics, July, 2019.
- [102] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, *A corpus and cloze evaluation for deeper understanding of commonsense stories*, in *NAACL*, (San Diego, California), pp. 839–849, Association for Computational Linguistics, June, 2016.
- [103] Y. Dauphin, A. Fan, M. Auli, and D. Grangier, *Language modeling with gated convolutional networks*, in *ICML*, 2017.
- [104] A. Sriram, H. Jun, S. Satheesh, and A. Coates, *Cold fusion: Training seq2seq models together with language models*, in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pp. 387–391, ISCA, 2018.
- [105] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. Dauphin, *Convolutional sequence to sequence learning*, in *ICML*, 2017.
- [106] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, *On the properties of neural machine translation: Encoder–decoder approaches*, in *Proceedings of SSSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, (Doha, Qatar), pp. 103–111, Association for Computational Linguistics, Oct., 2014.
- [107] B. Y. Lin, W. Zhou, M. Shen, P. Zhou, C. Bhagavatula, Y. Choi, and X. Ren, *CommonGen: A constrained text generation challenge for generative commonsense reasoning*, in *Findings of the Association for Computational Linguistics: EMNLP 2020*, (Online), pp. 1823–1840, Association for Computational Linguistics, Nov., 2020.
- [108] K. Ma, F. Ilievski, J. Francis, S. Ozaki, E. Nyberg, and A. Oltramari, *Exploring strategies for generalizable commonsense reasoning with pre-trained models*, in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, (Online and Punta Cana, Dominican Republic), pp. 5474–5483, Association for Computational Linguistics, Nov., 2021.
- [109] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, *Neural text generation with unlikelihood training*, in *ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.
- [110] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, *A diversity-promoting objective function for neural conversation models*, in *NAACL*, (San Diego, California), pp. 110–119, Association for Computational Linguistics, June, 2016.

- [111] K. Pillutla, S. Swayamdipta, R. Zellers, J. Thickstun, S. Welleck, Y. Choi, and Z. Harchaoui, *MAUVE: measuring the gap between neural text and human text using divergence frontiers*, in *NeurIPS*, pp. 4816–4828, 2021.
- [112] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, *BERTScore: Evaluating text generation with BERT*, in *ICLR*, 2020.
- [113] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, *Bleu: a method for automatic evaluation of machine translation*, in *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318, Association for Computational Linguistics, 2002.
- [114] S. Banerjee and A. Lavie, *METEOR: An automatic metric for MT evaluation with improved correlation with human judgments*, in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, (Ann Arbor, Michigan), pp. 65–72, Association for Computational Linguistics, June, 2005.
- [115] R. Vedantam, C. L. Zitnick, and D. Parikh, *Cider: Consensus-based image description evaluation*, *CVPR* (2015) 4566–4575.
- [116] P. Anderson, B. Fernando, M. Johnson, and S. Gould, *Spice: Semantic propositional image caption evaluation*, in *ECCV*, 2016.
- [117] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context*, in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.
- [118] T.-H. K. Huang, F. Ferraro, N. Mostafazadeh, I. Misra, A. Agrawal, J. Devlin, R. Girshick, X. He, P. Kohli, D. Batra, C. L. Zitnick, D. Parikh, L. Vanderwende, M. Galley, and M. Mitchell, *Visual storytelling*, in *NAACL*, (San Diego, California), pp. 1233–1239, Association for Computational Linguistics, June, 2016.
- [119] W. Wang, J. Peter, H. Rosendahl, and H. Ney, *Character: Translation edit rate on character level*, in *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11–12, Berlin, Germany*, pp. 505–510, The Association for Computer Linguistics, 2016.
- [120] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, *From word embeddings to document distances*, in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015*, vol. 37 of *JMLR Workshop and Conference Proceedings*, pp. 957–966, JMLR.org, 2015.

- [121] E. Clark, A. Celikyilmaz, and N. A. Smith, *Sentence mover’s similarity: Automatic evaluation for multi-sentence texts*, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 2748–2760, Association for Computational Linguistics, July, 2019.
- [122] T. Sellam, D. Das, and A. Parikh, *BLEURT: Learning robust metrics for text generation*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 7881–7892, Association for Computational Linguistics, July, 2020.
- [123] Z. Eviatar and M. A. Just, *Brain correlates of discourse processing: An fmri investigation of irony and conventional metaphor comprehension*, in *Neuropsychologia*, vol. 44, pp. 2348–2359, Elsevier, 2006.
- [124] C.-Y. Lin, *ROUGE: A package for automatic evaluation of summaries*, in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July, 2004.
- [125] J. Panja and S. K. Naskar, *ITER: improving translation edit rate through optimizable edit costs*, in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pp. 746–750, Association for Computational Linguistics, 2018.
- [126] M. G. Snover, B. J. Dorr, R. M. Schwartz, L. Micciulla, and J. Makhoul, *A study of translation edit rate with targeted human annotation*, in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, AMTA 2006, Cambridge, Massachusetts, USA, August 8-12, 2006*, pp. 223–231, Association for Machine Translation in the Americas, 2006.
- [127] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf, *Accelerated DP based search for statistical translation*, in *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997*, ISCA, 1997.
- [128] J. Tomás, J. À. Mas, and F. Casacuberta, *A quantitative method for machine translation evaluation*, in *Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing: are evaluation methods, metrics and resources reusable?*, pp. 27–34, 2003.
- [129] C. Lo, *MEANT 2.0: Accurate semantic MT evaluation for any output language*, in *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pp. 589–597, Association for Computational Linguistics, 2017.

- [130] C. Lo, *Yisi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources*, in *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pp. 507–513, Association for Computational Linguistics, 2019.
- [131] Y. Rubner, C. Tomasi, and L. J. Guibas, *A metric for distributions with applications to image databases*, in *Proceedings of the Sixth International Conference on Computer Vision (ICCV-98), Bombay, India, January 4-7, 1998*, pp. 59–66, IEEE Computer Society, 1998.
- [132] M. Jiang, Q. Huang, L. Zhang, X. Wang, P. Zhang, Z. Gan, J. Diesner, and J. Gao, *TIGEr: Text-to-image grounding for image caption evaluation*, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, (Hong Kong, China), pp. 2141–2152, Association for Computational Linguistics, Nov., 2019.
- [133] K. Lee, X. Chen, G. Hua, H. Hu, and X. He, *Stacked cross attention for image-text matching*, in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, vol. 11208 of *Lecture Notes in Computer Science*, pp. 212–228, Springer, 2018.
- [134] H. Lee, S. Yoon, F. Dernoncourt, D. S. Kim, T. Bui, and K. Jung, *ViLBERTScore: Evaluating image caption using vision-and-language BERT*, in *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, (Online), pp. 34–39, Association for Computational Linguistics, Nov., 2020.
- [135] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, *CLIPScore: A reference-free evaluation metric for image captioning*, in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, (Online and Punta Cana, Dominican Republic), pp. 7514–7528, Nov., 2021.
- [136] E. T. Troscianko, *Reading imaginatively: the imagination in cognitive science and cognitive literary studies*, in *Journal of Literary Semantics*, vol. 42, pp. 181–198, De Gruyter Mouton, 2013.
- [137] S. M. Kosslyn, G. Ganis, and W. L. Thompson, *Neural foundations of imagery*, in *Nature reviews neuroscience*, vol. 2, pp. 635–642, Nature Publishing Group, 2001.
- [138] J. Pearson and S. M. Kosslyn, *The heterogeneity of mental representation: Ending the imagery debate*, in *Proceedings of the National Academy of Sciences*, vol. 112, pp. 10089–10092, National Acad Sciences, 2015.

- [139] M. A. Nippold and J. K. Duthie, *Mental imagery and idiom comprehension: a comparison of school-age children and adults*, in *Journal of Speech, Language, and Hearing Research*, ASHA, 2003.
- [140] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An image is worth 16x16 words: Transformers for image recognition at scale*, in *ICLR*, 2021.
- [141] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*, in *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, Curran Associates, Inc., 2020.
- [142] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, *fairseq: A fast, extensible toolkit for sequence modeling*, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, (Minneapolis, Minnesota), pp. 48–53, Association for Computational Linguistics, June, 2019.
- [143] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, *Report on the 11th IWSLT evaluation campaign*, in *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, Dec. 4-5, 2014.
- [144] L. Barrault, O. Bojar, M. R. Costa-jussà, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, S. Malmasi, C. Monz, M. Müller, S. Pal, M. Post, and M. Zampieri, *Findings of the 2019 conference on machine translation (WMT19)*, in *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pp. 1–61, Association for Computational Linguistics, 2019.
- [145] P. Li, W. Lam, L. Bing, and Z. Wang, *Deep recurrent generative decoder for abstractive text summarization*, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 2091–2100, Association for Computational Linguistics, 2017.
- [146] W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, and M. Zhou, *Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training*, in *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, vol. EMNLP 2020 of *Findings of ACL*, pp. 2401–2410, Association for Computational Linguistics, 2020.
- [147] D. Freedman, R. Pisani, and R. Purves, *Statistics (international student edition)*. 2007.

- [148] A. Brock, J. Donahue, and K. Simonyan, *Large scale GAN training for high fidelity natural image synthesis*, in *ICLR*, 2019.
- [149] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, *Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models*, in *International Journal of Computer Vision*, vol. 123, pp. 74–93, 2015.
- [150] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, *Stanza: A python natural language processing toolkit for many human languages*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, (Online), pp. 101–108, Association for Computational Linguistics, July, 2020.
- [151] Z. J. Wang, E. Montoya, D. Munechika, H. Yang, B. Hoover, and D. H. Chau, *Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models*, 2022.
- [152] T. Chakrabarty, A. Saakyan, O. Winn, A. Panagopoulou, Y. Yang, M. Apidianaki, and S. Muresan, *I spy a metaphor: Large language models and diffusion models co-create visual metaphors*, in *Findings of the Association for Computational Linguistics: ACL 2023*, (Toronto, Canada), pp. 7370–7388, Association for Computational Linguistics, July, 2023.
- [153] D. M. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, *Universal sentence encoder for english*, in *Conference on Empirical Methods in Natural Language Processing*, 2018.
- [154] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*, in *Knowledge Discovery and Data Mining*, 1996.
- [155] Z. Yang, Z. Gan, J. Wang, X. Hu, Y. Lu, Z. Liu, and L. Wang, *An empirical study of gpt-3 for few-shot knowledge-based vqa*, in *AAAI Conference on Artificial Intelligence*, 2021.
- [156] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. ” O’Reilly Media, Inc.”, 2009.
- [157] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, *Emergent abilities of large language models*, *Transactions on Machine Learning Research* (2022). Survey Certification.

- [158] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et. al.*, *Chain-of-thought prompting elicits reasoning in large language models*, *Advances in Neural Information Processing Systems* **35** (2022) 24824–24837.
- [159] W. Feng, X. He, T.-J. Fu, V. Jampani, A. Akula, P. Narayana, S. Basu, X. E. Wang, and W. Y. Wang, *Training-free structured diffusion guidance for compositional text-to-image synthesis*, *ICLR* (2023).
- [160] K. Lee, H. Liu, M. Ryu, O. Watkins, Y. Du, C. Boutilier, P. Abbeel, M. Ghavamzadeh, and S. S. Gu, *Aligning text-to-image models using human feedback*, *arXiv preprint arXiv:2302.12192* (2023).
- [161] J. Liu, W. Xiong, I. Jones, Y. Nie, A. Gupta, and B. Oğuz, *Clip-layout: Style-consistent indoor scene synthesis with semantic furniture embedding*, *arXiv preprint arXiv:2303.03565* (2023).
- [162] A. Luo, Z. Zhang, J. Wu, and J. B. Tenenbaum, *End-to-end optimization of scene layout*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3754–3763, 2020.
- [163] W.-D. K. Ma, J. Lewis, W. B. Kleijn, and T. Leung, *Directed diffusion: Direct control of object placement through attention guidance*, *arXiv preprint arXiv:2302.13153* (2023).
- [164] B. Wang, T. Wu, M. Zhu, and P. Du, *Interactive image synthesis with panoptic layout generation*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7783–7792, 2022.
- [165] C.-F. Yang, W.-C. Fan, F.-E. Yang, and Y.-C. F. Wang, *Layouttransformer: Scene layout generation with conceptual and spatial diversity*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3732–3741, 2021.
- [166] K. Gupta, J. Lazarow, A. Achille, L. S. Davis, V. Mahadevan, and A. Shrivastava, *Layouttransformer: Layout generation and completion with self-attention*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1004–1014, 2021.
- [167] A. A. Jyothi, T. Durand, J. He, L. Sigal, and G. Mori, *Layoutvae: Stochastic scene layout generation from a label set*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9895–9904, 2019.
- [168] X. Kong, L. Jiang, H. Chang, H. Zhang, Y. Hao, H. Gong, and I. Essa, *Blt: bidirectional layout transformer for controllable layout generation*, in *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pp. 474–490, Springer, 2022.

- [169] J. Li, T. Xu, J. Zhang, A. Hertzmann, and J. Yang, *LayoutGAN: Generating graphic layouts with wireframe discriminator*, in *International Conference on Learning Representations*, 2019.
- [170] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee, *Gligen: Open-set grounded text-to-image generation*, *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023).
- [171] Z. Yang, J. Wang, Z. Gan, L. Li, K. Lin, C. Wu, N. Duan, Z. Liu, C. Liu, M. Zeng, *et. al.*, *Reco: Region-controlled text-to-image generation*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14246–14255, 2023.
- [172] Q. Wu, Y. Liu, H. Zhao, T. Bui, Z. Lin, Y. Zhang, and S. Chang, *Harnessing the spatial-temporal attention of diffusion models for high-fidelity text-to-image synthesis*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7766–7776, 2023.
- [173] C. Wu, S. Yin, W. Qi, X. Wang, Z. Tang, and N. Duan, *Visual chatgpt: Talking, drawing and editing with visual foundation models*, *arXiv preprint arXiv:2303.04671* (2023).
- [174] T. Brooks, A. Holynski, and A. A. Efros, *Instructpix2pix: Learning to follow image editing instructions*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, June, 2023.
- [175] L. Zhang, A. Rao, and M. Agrawala, *Adding conditional control to text-to-image diffusion models*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.
- [176] L. Huang, D. Chen, Y. Liu, Y. Shen, D. Zhao, and J. Zhou, *Composer: Creative and controllable image synthesis with composable conditions*, in *International Conference on Machine Learning*, 2023.
- [177] D. Ritchie, K. Wang, and Y.-a. Lin, *Fast and flexible indoor scene synthesis via deep convolutional generative models*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6182–6190, 2019.
- [178] K. Wang, Y.-A. Lin, B. Weissmann, M. Savva, A. X. Chang, and D. Ritchie, *Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks*, *ACM Transactions on Graphics (TOG)* **38** (2019), no. 4 1–15.
- [179] T.-J. Fu, W. Y. Wang, D. McDuff, and Y. Song, *DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents*, in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2022.

- [180] X. Zheng, X. Qiao, Y. Cao, and R. W. Lau, *Content-aware generative modeling of graphic design layouts*, *ACM Transactions on Graphics (TOG)* **38** (2019), no. 4 1–15.
- [181] X. Zhong, J. Tang, and A. J. Yepes, *Publaynet: largest dataset ever for document layout analysis*, in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1015–1022, IEEE, 2019.
- [182] B. Deka, Z. Huang, C. Franzen, J. Hibsman, D. Afergan, Y. Li, J. Nichols, and R. Kumar, *Rico: A mobile app dataset for building data-driven design applications*, in *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pp. 845–854, 2017.
- [183] E. M. Bakr, P. Sun, X. Shen, F. F. Khan, L. E. Li, and M. Elhoseiny, *HRS-Bench: Holistic, Reliable and Scalable Benchmark for Text-to-Image Models*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20041–20053, October, 2023.
- [184] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or, *Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models*, *ACM Transactions on Graphics (TOG)* **42** (2023), no. 4 1–10.
- [185] M. Chen, I. Laina, and A. Vedaldi, *Training-free layout control with cross-attention guidance*, *arXiv preprint arXiv:2304.03373* (2023).
- [186] X. Wang, C. Yeshwanth, and M. Nießner, *Sceneformer: Indoor scene generation with transformers*, in *2021 International Conference on 3D Vision (3DV)*, pp. 106–115, IEEE, 2021.
- [187] D. Paschalidou, A. Kar, M. Shugrina, K. Kreis, A. Geiger, and S. Fidler, *Atiss: Autoregressive transformers for indoor scene synthesis*, *Advances in Neural Information Processing Systems* **34** (2021) 12013–12026.
- [188] S. Frank, E. Bugliarello, and D. Elliott, *Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers*, in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9847–9857, 2021.
- [189] T.-J. Fu*, L. Li*, Z. Gan, K. Lin, W. Y. Wang, L. Wang, and Z. Liu, *An Empirical Study of End-to-End Video-Language Transformers with Masked Visual Modeling*, in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [190] Z. Li, C. Xie, B. Van Durme, and A. Yuille, *Localization vs. semantics: How can language benefit visual representation learning?*, *arXiv preprint arXiv:2212.00281* (2022).

- [191] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning, *Generating semantically precise scene graphs from textual descriptions for improved image retrieval*, in *Proceedings of the fourth workshop on vision and language*, pp. 70–80, 2015.
- [192] P. Lu, B. Peng, H. Cheng, M. Galley, K.-W. Chang, Y. N. Wu, S.-C. Zhu, and J. Gao, *Chameleon: Plug-and-play compositional reasoning with large language models*, *arXiv preprint arXiv:2304.09842* (2023).
- [193] Z. Yang, L. Li, J. Wang, K. Lin, E. Azarnasab, F. Ahmed, Z. Liu, C. Liu, M. Zeng, and L. Wang, *Mm-react: Prompting chatgpt for multimodal reasoning and action*, *arXiv preprint arXiv:2303.11381* (2023).
- [194] Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, and A. Smola, *Multimodal chain-of-thought reasoning in language models*, *arXiv preprint arXiv:2302.00923* (2023).
- [195] J. Y. Koh, R. Salakhutdinov, and D. Fried, *Grounding language models to images for multimodal generation*, in *International Conference on Machine Learning*, 2023.
- [196] T. Gupta and A. Kembhavi, *Visual programming: Compositional visual reasoning without training*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14953–14962, 2023.
- [197] D. Surís, S. Menon, and C. Vondrick, *Vipergpt: Visual inference via python execution for reasoning*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
- [198] Y. Lu, X. Yang, X. Li, X. E. Wang, and W. Y. Wang, *LLMScore: Unveiling the power of large language models in text-to-image synthesis evaluation*, in *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [199] V. Sanh, A. Webson, C. Raffel, S. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, T. L. Scao, S. Biderman, L. Gao, T. Wolf, and A. M. Rush, *Multitask prompted training enables zero-shot task generalization*, in *International Conference on Learning Representations*, 2022.
- [200] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, *Finetuned language models are zero-shot learners*, in *International Conference on Learning Representations*, 2022.

- [201] J. Cho, A. Zala, and M. Bansal, *Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3043–3054, 2023.
- [202] A. El-Nouby, S. Sharma, H. Schulz, D. Hjelm, L. E. Asri, S. E. Kahou, Y. Bengio, and G. W. Taylor, *Tell, Draw, and Repeat: Generating and Modifying Images Based on Continual Linguistic Instruction*, in *International Conference on Computer Vision (ICCV)*, 2019.
- [203] T.-J. Fu, X. E. Wang, S. Grafton, M. Eckstein, and W. Y. Wang, *SSCR: Iterative Language-Based Image Editing via Self-Supervised Counterfactual Reasoning*, in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.
- [204] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, *et. al.*, *Grounded language-image pre-training*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10965–10975, 2022.
- [205] OpenAI, *GPT-4 technical report*, *ArXiv* **abs/2303.08774** (2023).
- [206] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, *et. al.*, *3d-front: 3d furnished rooms with layouts and semantics*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.
- [207] H. Fu, R. Jia, L. Gao, M. Gong, B. Zhao, S. Maybank, and D. Tao, *3d-future: 3d furniture shape with texture*, *International Journal of Computer Vision* **129** (2021) 3313–3337.
- [208] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *Gans trained by a two time-scale update rule converge to a local nash equilibrium*, *Advances in neural information processing systems* **30** (2017).
- [209] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, *ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks*, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [210] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, *Voyager: An open-ended embodied agent with large language models*, *arXiv preprint arXiv: Arxiv-2305.16291* (2023).
- [211] G. Zhou, Y. Hong, and Q. Wu, *Navigpt: Explicit reasoning in vision-and-language navigation with large language models*, 2023.

- [212] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, *Touchdown: Natural language navigation and spatial reasoning in visual street environments*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Long Beach, California), 2019.
- [213] H. Mehta, Y. Artzi, J. Baldridge, E. Ie, and P. Mirowski, *Retouchdown: Releasing touchdown on StreetLearn as a public resource for language grounding tasks in street view*, in *Proceedings of the Third International Workshop on Spatial Language Understanding (SpLU)*, (Online), 2020.
- [214] R. Schumann and S. Riezler, *Analyzing generalization of vision and language navigation to unseen outdoor areas*, in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Dublin, Ireland), pp. 7519–7532, Association for Computational Linguistics, May, 2022.
- [215] Y. Sun, Y. Qiu, Y. Aoki, and H. Kataoka, *Outdoor vision-and-language navigation needs object-level alignment*, *Sensors* **23** (2023), no. 13.
- [216] J. Xiang, X. Wang, and W. Y. Wang, *Learning to stop: A simple yet effective approach to urban vision-language navigation*, in *Findings of the Association for Computational Linguistics (ACL Findings)*, (Online), 2020.
- [217] J. Armitage, L. Impett, and R. Sennrich, *A priority map for vision-and-language navigation with trajectory plans and feature-location cues*, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1094–1103, 2023.
- [218] W. Zhu, X. Wang, T.-J. Fu, A. Yan, P. Narayana, K. Sone, S. Basu, and W. Y. Wang, *Multimodal text style transfer for outdoor vision-and-language navigation*, in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, (Online), pp. 1207–1221, Association for Computational Linguistics, Apr., 2021.
- [219] V. Zhong, A. W. Hanjie, S. Wang, K. Narasimhan, and L. Zettlemoyer, *Silg: The multi-domain symbolic interactive language grounding benchmark*, in *Advances in Neural Information Processing Systems*, vol. 34, pp. 21505–21519, Curran Associates, Inc., 2021.
- [220] D. Shah, B. Osinski, B. Ichter, and S. Levine, *Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action*, 2022.
- [221] S. Chen, P.-L. Guhur, C. Schmid, and I. Laptev, *History aware multimodal transformer for vision-and-language navigation*, *Advances in neural information processing systems* **34** (2021) 5834–5847.

- [222] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, *Speaker-follower models for vision-and-language navigation*, in *Neural Information Processing Systems (NeurIPS)*, 2018.
- [223] T.-J. Fu, X. E. Wang, M. F. Peterson, S. T. Grafton, M. P. Eckstein, and W. Y. Wang, *Counterfactual vision-and-language navigation via adversarial path sampler*, in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pp. 71–86, Springer, 2020.
- [224] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould, *Vln bert: A recurrent vision-and-language bert for navigation*, in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 1643–1653, 2021.
- [225] J. Li, H. Tan, and M. Bansal, *Envedit: Environment editing for vision-and-language navigation*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15407–15417, 2022.
- [226] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. v. d. Hengel, *Reverie: Remote embodied visual referring expression in real indoor environments*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9982–9991, 2020.
- [227] H. Tan, L. Yu, and M. Bansal, *Learning to navigate unseen environments: Back translation with environmental dropout*, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 2610–2621, Association for Computational Linguistics, June, 2019.
- [228] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, *Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2019.
- [229] W. Zhu, H. Hu, J. Chen, Z. Deng, V. Jain, E. Ie, and F. Sha, *Babywalk: Going farther in vision-and-language navigation by taking baby steps*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2539–2556, 2020.
- [230] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, *Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments*, in *CVPR*, pp. 3674–3683, 2018.

- [231] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldrige, *Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4392–4412, 2020.
- [232] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, *Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI*, in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- [233] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, *Esc: Exploration with soft commonsense constraints for zero-shot object navigation*, *arXiv preprint arXiv:2301.13166* (2023).
- [234] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, *Simple but effective: Clip embeddings for embodied ai*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14829–14838, June, 2022.
- [235] V. S. Dorbala, G. Sigurdsson, R. Piramuthu, J. Thomason, and G. S. Sukhatme, *Clip-nav: Using clip for zero-shot vision-and-language navigation*, in *CoRL 2022 Workshop on Language and Robot Learning*, 2022.
- [236] R. Schumann and S. Riezler, *Generating landmark navigation instructions from maps as a graph-to-text problem*, in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Online), pp. 489–502, Association for Computational Linguistics, Aug., 2021.
- [237] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, *Language models are few-shot learners*, in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020.
- [238] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, *Opt: Open pre-trained transformer language models*, *ArXiv abs/2205.01068* (2022).
- [239] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, *Llama: Open and efficient foundation language models*, 2023.

- [240] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, *Llama 2: Open foundation and fine-tuned chat models*, 2023.
- [241] Mistral AI Team, *Mixtral of experts: A high quality sparse mixture-of-experts*, *Mistral AI Blog* (Dec., 2023). Accessed: December 18, 2023.
- [242] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, *LoRA: Low-rank adaptation of large language models*, in *International Conference on Learning Representations*, 2022.
- [243] J. Clarke, D. Goldwasser, M.-W. Chang, and D. Roth, *Driving semantic parsing from the world’s response*, in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, (Uppsala, Sweden), pp. 18–27, Association for Computational Linguistics, July, 2010.
- [244] S. Ross, G. J. Gordon, and J. A. Bagnell, *A reduction of imitation learning and structured prediction to no-regret online learning.*, in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, (Fort Lauderdale, FL, USA), 2011.
- [245] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. Iii, and K. Crawford, *Datasheets for datasets*, *Communications of the ACM* **64** (2021), no. 12 86–92.
- [246] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, *Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts*, in *CVPR*, 2021.
- [247] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, *YFCC100M: the new data in multimedia research*, *Communications of the ACM* **59** (2016), no. 2 64–73.
- [248] E. E. Marsh and M. Domas White, *A taxonomy of relationships between images and text*, *Journal of documentation* **59** (2003), no. 6 647–672.

- [249] A. Yan, Z. He, J. Li, T. Zhang, and J. McAuley, *Personalized showcases: Generating multi-modal explanations for recommendations*, in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2251–2255, 2023.
- [250] K. Srinivasan, K. Raman, J. Chen, M. Bendersky, and M. Najork, *WIT: Wikipedia-based image text dataset for multimodal multilingual machine learning*, in *SIGIR*, 2021.
- [251] A. Birhane, V. U. Prabhu, and E. Kahembwe, *Multimodal datasets: misogyny, pornography, and malignant stereotypes*, *arXiv preprint arXiv:2110.01963* (2021).
- [252] A. Mei, M. Saxon, S. Chang, Z. C. Lipton, and W. Y. Wang, *Users are the north star for ai transparency*, *arXiv preprint arXiv:2303.05500* (2023).
- [253] A. Aghajanyan, B. Huang, C. Ross, V. Karpukhin, H. Xu, N. Goyal, D. Okhonko, M. Joshi, G. Ghosh, M. Lewis, and L. Zettlemoyer, *Cm3: A causal masked multimodal model of the internet*, *ArXiv abs/2201.07520* (2022).
- [254] S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, Q. Liu, K. Aggarwal, Z. Chi, J. Bjorck, V. Chaudhary, S. Som, X. Song, and F. Wei, *Language is not all you need: Aligning perception with language models*, *ArXiv abs/2302.14045* (2023).
- [255] L. Van der Maaten and G. Hinton, *Visualizing data using t-sne.*, *Journal of machine learning research* **9** (2008), no. 11.
- [256] D. M. Blei, A. Y. Ng, and M. I. Jordan, *Latent dirichlet allocation*, *Journal of machine Learning research* **3** (2003), no. Jan 993–1022.
- [257] J. Dodge, M. Sap, A. Marasović, W. Agnew, G. Ilharco, D. Groeneveld, M. Mitchell, and M. Gardner, *Documenting large webtext corpora: A case study on the colossal clean crawled corpus*, in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, (Online and Punta Cana, Dominican Republic), pp. 1286–1305, Association for Computational Linguistics, Nov., 2021.
- [258] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. M. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. C. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. García, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz,

- E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Díaz, O. Firat, M. Catasta, J. Wei, K. S. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, *Palm: Scaling language modeling with pathways*, *ArXiv abs/2204.02311* (2022).
- [259] D. So, W. Mañke, H. Liu, Z. Dai, N. Shazeer, and Q. V. Le, *Searching for efficient transformers for language modeling*, in *Advances in Neural Information Processing Systems*, 2021.
- [260] J. Suzuki, H. Zen, and H. Kazawa, *Extracting representative subset from extensive text data for training pre-trained language models*, *Information Processing & Management* **60** (2023), no. 3 103249.
- [261] R. Thoppilan, D. D. Freitas, J. Hall, N. M. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, Y. Li, H. Lee, H. Zheng, A. Ghafouri, M. Menegali, Y. Huang, M. Krikun, D. Lepikhin, J. Qin, D. Chen, Y. Xu, Z. Chen, A. Roberts, M. Bosma, Y. Zhou, C.-C. Chang, I. A. Krivokon, W. J. Rusch, M. Pickett, K. S. Meier-Hellstern, M. R. Morris, T. Doshi, R. D. Santos, T. Duke, J. H. Søraaker, B. Zevenbergen, V. Prabhakaran, M. Díaz, B. Hutchinson, K. Olson, A. Molina, E. Hoffman-John, J. Lee, L. Aroyo, R. Rajakumar, A. Butryna, M. Lamm, V. O. Kuzmina, J. Fenton, A. Cohen, R. Bernstein, R. Kurzweil, B. Aguera-Arcas, C. Cui, M. Croak, E. H. hsin Chi, and Q. Le, *Lamda: Language models for dialog applications*, *ArXiv abs/2201.08239* (2022).
- [262] S. Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortsman, D. Ghosh, J. Zhang, E. Orgad, R. Entezari, G. Daras, S. Pratt, V. Ramanujan, Y. Bitton, K. Marathe, S. Mussmann, R. Vencu, M. Cherti, R. Krishna, P. W. Koh, O. Saukh, A. J. Ratner, S. Song, H. Hajishirzi, A. Farhadi, R. Beaumont, S. Oh, A. G. Dimakis, J. Jitsev, Y. Carmon, V. Shankar, and L. Schmidt, *Datacomp: In search of the next generation of multimodal datasets*, *arXiv preprint arXiv:2304.14108* (2023).
- [263] J. Hessel, L. Lee, and D. Mimno, *Unsupervised discovery of multimodal links in multi-image, multi-sentence documents*, in *EMNLP*, 2019.
- [264] Z. Li, Z. Wei, Z. Fan, H. Shan, and X. Huang, *An unsupervised sampling approach for image-sentence matching using document-level structural information*, in *AAAI*, 2021.
- [265] H. W. Kuhn, *The hungarian method for the assignment problem*, *Naval research logistics quarterly* **2** (1955), no. 1-2 83–97.
- [266] R. Jonker and T. Volgenant, *A shortest augmenting path algorithm for dense and sparse linear assignment problems*, in *DGOR/NSOR: Papers of the 16th Annual*

Meeting of DGOR in Cooperation with NSOR/Vorträge der 16. Jahrestagung der DGOR zusammen mit der NSOR, pp. 622–622, Springer, 1988.

- [267] K. Desai, G. Kaul, Z. Aysola, and J. Johnson, *RedCaps: Web-curated image-text data created by the people, for the people*, in *NeurIPS Datasets and Benchmarks*, 2021.
- [268] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, *Large-scale privacy protection in google street view*, in *2009 IEEE 12th international conference on computer vision*, pp. 2373–2380, IEEE, 2009.
- [269] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, *Retinaface: Single-shot multi-level face localisation in the wild*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.
- [270] S. I. Serengil and A. Ozpinar, *Lightface: A hybrid deep face recognition framework*, in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 23–27, IEEE, 2020.
- [271] S. I. Serengil and A. Ozpinar, *Hyperextended lightface: A facial attribute analysis framework*, in *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1–4, IEEE, 2021.
- [272] A. K. McCallum, *Mallet: A machine learning for language toolkit*, 2002.
- [273] J. Li, D. Li, C. Xiong, and S. Hoi, *Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation*, in *International Conference on Machine Learning*, pp. 12888–12900, PMLR, 2022.
- [274] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et. al.*, *On the opportunities and risks of foundation models*, *arXiv preprint arXiv:2108.07258* (2021).
- [275] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, *Palm-e: An embodied multimodal language model*, .
- [276] H. Liu, C. Li, Q. Wu, and Y. J. Lee, *Visual instruction tuning*, in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [277] R. Zhang, J. Han, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, P. Gao, and Y. Qiao, *Llama-adapter: Efficient fine-tuning of language models with zero-init attention*, *arXiv preprint arXiv:2303.16199* (2023).

- [278] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, *Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models*, in *International Conference on Machine Learning*, 2023.
- [279] Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi, C. Li, Y. Xu, H. Chen, J. Tian, Q. Qi, J. Zhang, and F. Huang, *mplug-owl: Modularization empowers large language models with multimodality*, *ArXiv abs/2304.14178* (2023).
- [280] MosaicML, *Introducing mpt-7b: A new standard for open-source, commercially usable llms*, 2023.
- [281] Together.xyz, *Releasing 3b and 7b redpajama-incite family of models including base, instruction-tuned & chat models*, 2023.
- [282] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, *et. al.*, *Pali: A jointly-scaled multilingual language-image model*, *arXiv preprint arXiv:2209.06794* (2022).
- [283] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, *Scaling up visual and vision-language representation learning with noisy text supervision*, in *International Conference on Machine Learning*, pp. 4904–4916, PMLR, 2021.
- [284] H. Laurençon, L. Saulnier, L. Tronchon, S. Bekman, A. Singh, A. Lozhkov, T. Wang, S. Karamcheti, A. M. Rush, D. Kiela, M. Cord, and V. Sanh, *OBELICS: An open web-scale filtered dataset of interleaved image-text documents*, in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.
- [285] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, *Perceiver: General perception with iterative attention*, in *International conference on machine learning*, pp. 4651–4664, PMLR, 2021.
- [286] Thomas Breuel, “WebDataset: A high-performance Python-based I/O system for large (and small) deep learning problems, with strong support for PyTorch..” Available at: <https://github.com/webdataset/webdataset>, 2020.
- [287] Y. Zhao, A. Gu, R. Varma, L. Luo, C.-C. Huang, M. Xu, L. Wright, H. Shojanazeri, M. Ott, S. Shleifer, *et. al.*, *Pytorch fsdp: experiences on scaling fully sharded data parallel*, *arXiv preprint arXiv:2304.11277* (2023).
- [288] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, *Microsoft coco captions: Data collection and evaluation server*, *arXiv preprint arXiv:1504.00325* (2015).

- [289] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, *From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions*, *Transactions of the Association for Computational Linguistics* **2** (2014) 67–78.
- [290] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Parikh, and D. Batra, *Vqa: Visual question answering*, *International Journal of Computer Vision* **123** (2015) 4–31.
- [291] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi, *Ok-vqa: A visual question answering benchmark requiring external knowledge*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3190–3199, 2019.
- [292] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, *Towards vqa models that can read*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8309–8318, 2019.
- [293] D. Gurari, Q. Li, A. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, *Vizwiz grand challenge: Answering visual questions from blind people*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617, 2018.
- [294] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, *The hateful memes challenge: Detecting hate speech in multimodal memes*, *arXiv preprint arXiv:2005.04790* (2020).
- [295] B. Li, Y. Zhang, L. Chen, J. Wang, J. Yang, and Z. Liu, *Otter: A multi-modal model with in-context instruction tuning*, *arXiv preprint arXiv:2305.03726* (2023).
- [296] B. Li, Y. Zhang, L. Chen, J. Wang, F. Pu, J. Yang, C. Li, and Z. Liu, *Mimic-it: Multi-modal in-context instruction tuning*, *arXiv preprint arXiv:2306.05425* (2023).
- [297] T. Gong, C. Lyu, S. Zhang, Y. Wang, M. Zheng, Q. Zhao, K. Liu, W. Zhang, P. Luo, and K. Chen, *Multimodal-gpt: A vision and language model for dialogue with humans*, *arXiv preprint arXiv:2305.04790* (2023).
- [298] A. Aspell, Y. Bai, A. Chen, D. Drain, D. Ganguli, T. Henighan, A. Jones, N. Joseph, B. Mann, N. DasSarma, *et. al.*, *A general language assistant as a laboratory for alignment*, *arXiv preprint arXiv:2112.00861* (2021).
- [299] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, *Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality*, March, 2023.

- [300] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, “Stanford alpaca: An instruction-following llama model.” https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [301] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Naik, A. Ashok, A. S. Dhanasekaran, A. Arunkumar, D. Stap, *et. al.*, *Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks*, in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, 2022.
- [302] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi, *Instructblip: Towards general-purpose vision-language models with instruction tuning*, *arXiv preprint arXiv:2305.06500* (2023).
- [303] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue, *et. al.*, *Llama-adapter v2: Parameter-efficient visual instruction model*, *arXiv preprint arXiv:2304.15010* (2023).
- [304] J. Li, D. Li, S. Savarese, and S. Hoi, *Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models*, *arXiv preprint arXiv:2301.12597* (2023).
- [305] Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, and D. Cai, *Pandagpt: One model to instruction-follow them all*, *arXiv preprint arXiv:2305.16355* (2023).
- [306] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, *Making the v in vqa matter: Elevating the role of image understanding in visual question answering*, in *CVPR*, pp. 6904–6913, 2017.
- [307] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, *Vqa: Visual question answering*, in *ICCV*, 2015.
- [308] D. H. Park, T. Darrell, and A. Rohrbach, *Robust change captioning*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4624–4633, 2019.
- [309] Y. Hu, H. Hua, Z. Yang, W. Shi, N. A. Smith, and J. Luo, *Promptcap: Prompt-guided task-aware image captioning*, *arXiv preprint arXiv:2211.09699* (2022).
- [310] J. Pont-Tuset, J. Uijlings, S. Changpinyo, R. Soricut, and V. Ferrari, *Connecting vision and language with localized narratives*, in *European conference on computer vision*, pp. 647–664, Springer, 2020.
- [311] A. E. Elo, *The proposed uscf rating system. its development, theory, and applications*, *Chess Life* **22** (1967), no. 8 242–247.

- [312] L. Zheng, Y. Sheng, W.-L. Chiang, H. Zhang, J. E. Gonzalez, and I. Stoica, *Chatbot arena: Benchmarking llms in the wild with elo ratings*, .
- [313] L. Li, Y. Yin, S. Li, L. Chen, P. Wang, S. Ren, M. Li, Y. Yang, J. Xu, X. Sun, *et. al.*, *M3it: A large-scale dataset towards multi-modal multilingual instruction tuning*, *arXiv preprint arXiv:2306.04387* (2023).
- [314] F. Liu, K. Lin, L. Li, J. Wang, Y. Yacoob, and L. Wang, *Aligning large multi-modal model with robust instruction tuning*, *arXiv preprint arXiv:2306.14565* (2023).
- [315] P. Xu, W. Shao, K. Zhang, P. Gao, S. Liu, M. Lei, F. Meng, S. Huang, Y. Qiao, and P. Luo, *Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models*, *arXiv preprint arXiv:2306.09265* (2023).
- [316] Z. Xu, Y. Shen, and L. Huang, *Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning*, *arXiv preprint arXiv:2212.10773* (2022).
- [317] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, *A corpus for reasoning about natural language grounded in photographs*, *arXiv preprint arXiv:1811.00491* (2018).
- [318] S. Wiegrefe, J. Hessel, S. Swayamdipta, M. Riedl, and Y. Choi, *Reframing human-ai collaboration for generating free-text explanations*, *arXiv preprint arXiv:2112.08674* (2021).
- [319] A. Zeng, M. Attarian, B. Ichter, K. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, *et. al.*, *Socratic models: Composing zero-shot multimodal reasoning with language*, *arXiv preprint arXiv:2204.00598* (2022).
- [320] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, *Self-instruct: Aligning language model with self generated instructions*, *arXiv preprint arXiv:2212.10560* (2022).
- [321] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, in *CVPR*, 2016.
- [322] N. Bitton-Guetta, Y. Bitton, J. Hessel, L. Schmidt, Y. Elovici, G. Stanovsky, and R. Schwartz, *Breaking common sense: Whoops! a vision-and-language benchmark of synthetic and compositional images*, *arXiv preprint arXiv:2303.07274* (2023).
- [323] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, *Minigt-4: Enhancing vision-language understanding with advanced large language models*, *arXiv preprint arXiv:2304.10592* (2023).

- [324] Y. Huang, Z. Meng, F. Liu, Y. Su, N. Collier, and Y. Lu, *Sparkles: Unlocking chats across multiple images for multimodal instruction-following models*, *arXiv preprint arXiv:2308.16463* (2023).
- [325] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, *arXiv preprint arXiv:1907.11692* (2019).
- [326] Y. Dubois, X. Li, R. Taori, T. Zhang, I. Gulrajani, J. Ba, C. Guestrin, P. Liang, and T. B. Hashimoto, *Alpacafarm: A simulation framework for methods that learn from human feedback*, 2023.
- [327] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, *Large language models are zero-shot reasoners*, *Advances in neural information processing systems* **35** (2022) 22199–22213.
- [328] R. Pandey, R. Shao, P. P. Liang, R. Salakhutdinov, and L.-P. Morency, *Cross-modal attention congruence regularization for vision-language relation alignment*, *arXiv preprint arXiv:2212.10549* (2022).
- [329] X. Geng, A. Gudibande, H. Liu, E. Wallace, P. Abbeel, S. Levine, and D. Song, “Koala: A dialogue model for academic research.” Blog post, April, 2023.
- [330] B. Peng, C. Li, P. He, M. Galley, and J. Gao, *Instruction tuning with gpt-4*, *arXiv preprint arXiv:2304.03277* (2023).
- [331] D. Yin, X. Liu, F. Yin, M. Zhong, H. Bansal, J. Han, and K.-W. Chang, *Dynosaur: A dynamic growth paradigm for instruction-tuning data curation*, *arXiv preprint arXiv:2305.14327* (2023).
- [332] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, *et. al.*, *Lima: Less is more for alignment*, *arXiv preprint arXiv:2305.11206* (2023).
- [333] D. A. Hudson and C. D. Manning, *Gqa: A new dataset for real-world visual reasoning and compositional question answering*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.
- [334] T. Liao, R. Taori, I. D. Raji, and L. Schmidt, *Are we learning yet? a meta review of evaluation failures across machine learning*, in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- [335] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et. al.*, *Imagenet large scale visual recognition challenge*, *International journal of computer vision* **115** (2015) 211–252.

- [336] A. Krizhevsky, G. Hinton, *et. al.*, *Learning multiple layers of features from tiny images*, .
- [337] P. Rajpurkar, R. Jia, and P. Liang, *Know what you don't know: Unanswerable questions for squad*, *arXiv preprint arXiv:1806.03822* (2018).
- [338] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, *Superglue: A stickier benchmark for general-purpose language understanding systems*, *Advances in neural information processing systems* **32** (2019).
- [339] L. Gao, J. Tow, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, K. McDonell, N. Muennighoff, *et. al.*, *A framework for few-shot language model evaluation*, *Version v0. 0.1. Sept* (2021).
- [340] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, *et. al.*, *Holistic evaluation of language models*, *arXiv preprint arXiv:2211.09110* (2022).
- [341] J. Miller, K. Krauth, B. Recht, and L. Schmidt, *The effect of natural distribution shift on question answering models*, in *International Conference on Machine Learning*, pp. 6905–6916, PMLR, 2020.
- [342] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, *Do cifar-10 classifiers generalize to cifar-10?*, *arXiv preprint arXiv:1806.00451* (2018).
- [343] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, *Do imagenet classifiers generalize to imagenet?*, in *International conference on machine learning*, pp. 5389–5400, PMLR, 2019.
- [344] R. Roelofs, V. Shankar, B. Recht, S. Fridovich-Keil, M. Hardt, J. Miller, and L. Schmidt, *A meta-analysis of overfitting in machine learning*, *Advances in Neural Information Processing Systems* **32** (2019).
- [345] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, *A survey on multimodal large language models*, *arXiv preprint arXiv:2306.13549* (2023).