eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Master Thesis

Master in Language Analysis and Processing

Grounding Language Models for Compositional and Spatial Reasoning

Author
Julen Etxaniz

Advisors
Oier Lopez de Lacalle
Aitor Soroa

Departments
Computer Systems and Languages
Computational Science and Artificial Intelligence

October 2022

Abstract

Humans can learn to understand and process the distribution of space, and one of the initial tasks of Artificial Intelligence has been to show machines the relationships between space and the objects that appear in it. Humans naturally combine vision and textual information to acquire compositional and spatial relationships among objects, and when reading a text, we are able to mentally depict the spatial relationships that may appear in it. Thus, the visual differences between images depicting "a person sits and a dog stands" and "a person stands and a dog sits" are obvious for humans, but still not clear for automatic systems.

In this project, we propose to evaluate grounded Neural Language models that can perform compositional and spatial reasoning. Neural Language models (LM) have shown impressive capabilities on many NLP tasks but, despite their success, they have been criticized for their lack of meaning. Vision-and-Language models (VLM), trained jointly on text and image data, have been offered as a response to such criticisms, but recent work has shown that these models struggle to ground spatial concepts properly. In the project, we evaluate state-of-the-art pre-trained and fine-tuned VLMs to understand their grounding level on compositional and spatial reasoning. We also propose a variety of methods to create synthetic datasets specially focused on compositional reasoning.

We managed to accomplish all the objectives of this work. First, we improved the state-of-the-art in compositional reasoning. Next, we performed some zero-shot experiments on spatial reasoning. Finally, we explored three alternatives for synthetic dataset creation: text-to-image generation, image captioning and image retrieval. Code is released at `https://github.com/juletx/spatial-reasoning` and models are released at `https://huggingface.co/juletxara`.

Keywords: Artificial Intelligence, Deep Learning, Natural Language Processing, Computer Vision, Grounding, Visual Reasoning, Compositional Reasoning, Spatial Reasoning

Acknowledgements

I would like to thank everyone that has helped in the development of this project. Specially to my directors Oier Lopez de Lacalle and Aitor Soroa. I also want to thank Gorka Azkune, Ander Salaberria and Eneko Agirre, who have also attended the weekly meetings and have taken part in the annotation process. Their ideas and advice have helped me a lot in this work.

The infrastructure for this project was provided by the IXA research group. IXA is internationally acknowledged as an expert in Deep Learning and Natural Language Processing, as shown by the more than 3000 citations obtained in the last two years. The IXA research group owns several high-performance servers with the necessary storage and GPUs to accomplish the project objectives.

Images that correspond to datasets and models are taken from the original papers. Diffusion images are from the HuggingFace Diffusers examples. Image retrieval images are from LAION-5B dataset. Winoground images are a compilation of assets, including ©Getty Images/Natasha Breen, Maki Nakamura, Jessica Peterson, Kundanlall Sharma, lacaosa, Alberto Bogo, Vu Le, Toson Rueangsuksut, Nisian Hughes, Tanja Walter, Douglas Sacha, PBNJ Productions, Glow Images, 10'000 Hours, zoranm, Marlene Ford, Westend61.

Contents

List of Figures

List of Tables

1 Introduction

Neural **Language Models** (LM) have shown **impressive capabilities** on many Natural Language Processing (NLP) tasks [1, 2, 3]. LMs are pretrained on large corpora in order for them to learn universal language representations, which are beneficial for downstream NLP tasks and can avoid training a new model from scratch. The **pretrained models are fine-tuned in specific downstream tasks**, using annotated data that is orders of magnitude smaller than the text used in the pretraining phase. Following this transfer learning methodology, researchers have extended the state of the art on a wide array of tasks as measured by leaderboards on specific benchmarks for English [4, 1].

Despite the impressive results of LMs for different language-related tasks, many authors criticize them for their **lack of meaning** [5, 6]. In their opinion, language models trained exclusively on language are unable to learn meaning. Those authors suggest that **grounding is one of the key elements to bring human-like language understanding**. However, language grounding is a very broad area that covers a great diversity of techniques, modalities and concepts.

In this project, we will focus on grounding LMs for **compositional and spatial reasoning**. We choose compositional and spatial reasoning because they are crucial to how humans organize the mental space and make sense of the physical world, and therefore fundamental for a grounded theory of semantics [7]. However, **spatial reasoning** has been found to be **particularly challenging for current models** [8]. Another work has found that these models are not as skilled at visio-linguistic **compositional reasoning** as we might have hoped [9]. That is, compositional and spatial reasoning are much more challenging than capturing properties of individual entities.

Vision Language Models (VLMs), which are trained jointly on text and image, have been proposed as a general solution to the lack of grounding in language models [10, 11, 12, 13]. Vision-language pre-training aims to improve performance of downstream vision and language tasks by **pretraining the model on many image-text pairs**. These pre-trained models can then be fine-tuned on each downstream task. VLMs have been fine-tuned in tasks that require grounding spatial concepts, such as VQA [14] or NLVR2 [15].

With the objective of **evaluating spatial relations**, a recent work provides new unified datasets [16]. As the objective of such work is to evaluate whether VLMs learn more spatial commonsense than LMs, the datasets are purely textual, so they do not provide any means to ground spatial concepts. Interestingly, authors find that VLMs, and more concretely text-to-image systems, perform much better than text-only LMs. Still, they show that VLMs **struggle to ground spatial concepts properly**.

Large generative **text-to-image diffusion models**, like DALLE-2 [12], IMAGEN [13] and Stable Diffusion [17], are able to generate stunning images. They are known to possess some visual-reasoning skills [18]. However, a recent work [19] has shown that they **struggle to understand the composition of some concepts**, such as confusing the attributes and relations of different objects. They propose a new method, where an image is generated by composing a set of diffusion models, with each of them modelling a certain component of the image. Another work [20] proposes manipulating cross-attention representations to address three challenging phenomena in Stable Diffusion [17]: attribute leakage, interchanged attributes and missing objects.

There are several **works that try to ground language models to spatial relations**. For example, [21, 22] focus on the acquired commonsense knowledge of models about object scales (e.g. is a person bigger than an ant?). However, they ask about generic object scale relations, without providing any context. Some other authors [23, 24] work on implicit and explicit spatial relations of objects, given some descriptive texts. The proposed benchmark datasets are designed for object bounding box generation.

1.1 Objectives

Despite the impressive performance of pretrained vision and language models (VLMs) on a wide variety of multimodal tasks, they remain poorly understood. One important question is to what extent such models are able to conduct unimodal and multimodal **compositional reasoning** and **spatial reasoning**. For example, the visual differences between images depicting "a person sits and a dog stands" and "a person stands and a dog sits" are really obvious for humans, but still not clear for current state-of-the-art VLMs. To perform well on tasks where compositional and spatial reasoning is required, the models do not only need a proper encoding of text and images but also to be able to **ground meaning across the two modalities**.

Thus the main objective of the project is to **evaluate grounded language models for compositional and spatial reasoning**. The goal is to investigate ways to acquire grounded representation for compositional and spatial reasoning. In that sense, we will evaluate state-of-the-art pre-trained and fine-tuned VLMs to understand their grounding level. Towards this goal, this project will focus on using the latest advances in deep-learning techniques, and pre-trained LMs for effective zero-shot transfer learning. We have defined these objectives for this project:

Improve the state of the art in compositional reasoning. Winoground [9] is a dataset for evaluating the ability of VLMs to conduct visio-linguistic compositional reasoning. The original Winoground paper included zero-shot experiments with many pre-trained SOTA systems, and they concluded that none of them does much better than chance. In this work, we will extend the previous experiments with new pre-trained models and models that are fine-tuned for specific tasks such as image-text retrieval and visual reasoning. Our objective is to improve results and to find out which models and tasks help most for compositional reasoning.

Perform zero-shot experiments in spatial reasoning. Visual Spatial Reasoning [25] is a dataset for investigating VLMs capabilities in recognising 65 types of spatial relationships in natural text-image pairs. VSR authors [25] train and test three popular VLMs: VisualBERT [26], LXMERT [11], and ViLT [27]. They conclude that there is still a large gap between model and human performance. We will extend these experiments with zero-shot experiments with ViLT [27] and BLIP [28] models fine-tuned on NLVR2 dataset. Our aim is to measure how this affects performance on VSR.

Investigate the use of synthetic datasets to overcome the lack of annotated datasets. As to avoid the scarcity of multimodal datasets that explicitly describe compositional and spatial relations, we propose to automatically construct synthetic datasets. These datasets could be used in the future to train existing language models in a self-supervised way, with the final aim of obtaining spatially grounded language models. In particular, we will explore the viability of three alternatives with evaluations: text-to-image generation, image captioning and image retrieval. These techniques could be combined to produce synthetic datasets in the future.

Large generative text-to-image VLMs are known to generate great images from text descriptions, and they possess some visual reasoning skills. We will use Stable Diffusion [17] to obtain realistic images with entities that are arranged following certain spatial relations. We will do a manual evaluation of the generated images to measure their quality and decide if they can be used to create synthetic datasets.

Image captioning models could provide extra information about the images to the models, that is not included in the original captions. Most large datasets contain captions directly obtained from the image descriptions on the internet. Those descriptions are often not very good. Image captioning models can be used to get decent captions much faster than with human annotation. We will do a qualitative evaluation of a small sample of captions to know how good they are.

Image retrieval systems could be used to retrieve images of interest from large image datasets. Images can be retrieved using similarity scores between caption and image embeddings. This can be combined with captioning to improve original captions. We will evaluate a small sample of the retrieved images to decide their viability for synthetic dataset generation.

1.2 Structure

This section provides an overview of the next chapters in this work: 2 Background, 3 Winoground Zero-shot Experiments, 4 VSR Zero-shot Experiments, 5 Synthetic Dataset Creation, 6 Conclusions and 7 Future Work.

First, Chapter 2 contains the background knowledge used in this project's development. This chapter includes two main sections: Multimodal Models and Visual Reasoning Datasets. Section 2.1 explains the types of models that are related to this work. Section 2.2 includes synthetic and natural visual reasoning datasets and the datasets that we chose for this work.

Second, Chapter 3 explains how we achieve the first objective, improving SOTA in compositional reasoning. First, it describes the Winoground [9] dataset (Section 3.1) and explains the metrics used for evaluation. We also describe a series of previous and new experiments performed over the Winoground dataset using state-of-the-art vision and language models (Section 3.3). The Winoground dataset does not contain a training split, and therefore the experiments are conducted in a zero-shot fashion, where the models are trained on different datasets, and tested on Winoground.

Next, Chapter 4 is related to the second objective, performing zero-shot experiments on spatial reasoning. We focus on Visual Spatial Reasoning [25], a dataset for spatial reasoning (Section 4.1). Unlike Winoground, VSR contains training and validation splits and can be used to train models (Section 4.2). However, we mainly focus on zero-shot experiments with models that are trained on other datasets. We explain previous and new experiments we performed and the results we obtained in VSR (Section 4.3).

Then, Chapter 5 describes the third objective, synthetic dataset creation. We investigate three different options for synthetic dataset generation: Text-to-Image Generation (5.1), Image Captioning (5.2) and Image Retrieval (5.3). These experiments also allow us to gain more insight into the dataset and the tested models.

Finally, Chapters 6 and 7 provide an overview of the main contributions, conclusions and future work areas for further research. This chapter provides an overview of future work areas for further research. We propose four ideas for synthetic dataset generation (Section 7.1): explicit verbalization (7.1.1), text-to-image (7.1.2), image-to-image (7.1.3) and image captioning and retrieval (7.1.4). We also include some ideas for extending current datasets to be multilingual (Section 7.2).

2 Background

This chapter introduces the background knowledge used in this project's development. This chapter includes two main sections: Multimodal Models and Visual Reasoning Datasets. Section 2.1 explains the types of models that are related to this work. Section 2.2 includes synthetic and natural visual reasoning datasets and the datasets that we chose for this work.

2.1 Multimodal Models

Multimodal models are trained jointly on text and image pairs. This is different from language models, which are only trained with text and vision models which only use images. The aim of VLMs is to ground LMs with visual concepts.

This section explains the types of models that are related to this work, Multimodal Transformers, Multimodal RNNs and Diffusion Models. Section 2.1.1 includes descriptions of the following **multimodal transformers**: OFA [29], BLIP [28], CLIP [30], OpenCLIP [31], FLAVA [32], LXMERT [11], UniT [33], UNITER [34], VILLA [35], VinVL [36], ViLT [27], VisualBERT [26] and ViLBERT [10]. Section 2.1.1 explains two types of **multimodal RNN** models: VSE++ [37] and VSRN [38]. Section 2.1.3 introduces **diffusion models** and explains Stable Diffusion [17], the diffusion model that we use in this work.

Overview. Table 2.1 provides a high-level overview of the Transformer and RNN models that are described in the next sections. This overview includes pretraining datasets, architecture, and attention mechanisms between the modalities. We omit datasets that were only used to train backbones. We exclude the language embedding from this table as every model uses a pretrained BERT tokenizer, except CLIP, VSE++, and VSRN. The pretraining datasets include COCO [39], Visual Genome (VG) [40], Conceptual Captions (CC) [41], SBU Captions [42], Flickr30k [43], VQA 2.0 [44], VCR [45], NLVR2 [46], SNLI-VE [47], QNLI [48], MLNI-mm [49], QQP [50], Localized Narratives (LN) [51], Wikipedia Image Text (WIT) [52], Conceptual Captions 12M (CC 12M) [53], Red Caps (RC) [54], YFCC100M [55], SST-2 [56], LAION-400M [57] and LAION-2B [58]. CLIP uses their own dataset for pretraining.

Model	Datasets	# Images, Captions	Architecture	Attention
VinVL [36]	VQA, GQA, VG-QA, COCO, Flickr30k, CC, SBU	1.89, 4.87	single-stream	merged
UNITER [34]	COCO, VG, CC, SBU	4.20, 9.58	single-stream	merged
ViLLA [35]	COCO, VG, CC, SBU	4.20, 9.58	single-stream	merged
VisualBERT [26]	COCO, NVLR2	0.30, 0.52	single-stream	merged
ViLT [27]	COCO, VG, SBU, CC	4.10, 9.85	single-stream	merged
LXMERT [11]	COCO, VG	0.18, 9.18	dual-stream	modality-specific, co-attn, merged
ViLBERT [10]	CC	3.30, 3.30	dual-stream	modality-specific, co-attn, merged
UniT [33]	COCO, VG, VQAv2, SNLI-VE QNLI, MNLI-mm, QQP, SST-2	0.69, 1.91	dual-stream	modality-specific, merged
FLAVA $_{ITM}$ [32]	COCO, SBU, LN, CC, VG, WIT, CC 12M, RC, YFCC100M	70.00, 70.00	dual-stream	modality-specific, merged
FLAVA $_{ITC}$ [32]	COCO, SBU, LN, CC, VG, WIT, CC 12M, RC, YFCC100M	70.00, 70.00	dual-stream	modality-specific
CLIP [30]	–	400.00, 400.00	dual-stream	modality-specific
OpenCLIP [31]	LAION-2B	2320.00, 2320.00	dual-stream	modality-specific
OFA [29]	CC 12M, CC 3M, SBU, COCO, VG-Cap	20.00, 20.00	single-stream	modality-specific, merged
BLIP$_{ITM}$ 14M [28]	COCO, VG, SBU, CC, CC 12M	14.00, 15.00	dual-stream	modality-specific, merged
BLIP$_{ITC}$ 14M [28]	COCO, VG, SBU, CC, CC 12M	14.00, 15.00	dual-stream	modality-specific
BLIP$_{ITM}$ 129M [28]	COCO, VG, SBU, CC, CC 12M, LAION-400M	129.00, 130.00	dual-stream	modality-specific, merged
BLIP$_{ITC}$ 129M [28]	COCO, VG, SBU, CC, CC 12M, LAION-400M	129.00, 130.00	dual-stream	modality-specific
VSE++ $_{COCO}$ [37]	COCO	0.11, 0.57	dual-stream	–
VSE++ $_{Flickr30k}$ [37]	Flickr30k	0.03, 0.16	dual-stream	–
VSRN $_{COCO}$ [38]	COCO	0.11, 0.57	dual-stream	–
VSRN $_{Flickr30k}$ [38]	Flickr30k	0.03, 0.16	dual-stream	–

Table 2.1: A high-level overview of the differences between the models by the pretraining datasets, architecture, and attention mechanisms between the modalities.

2.1.1 Multimodal Transformers

Multimodal transformers are state-of-the-art in many vision-language tasks, and that includes spatial reasoning. Most of the models tested in Winoground [9] and VSR [25] are multimodal transformers. Those transformers differ in embedding, architecture, pretraining objectives and cross-modal attention. First, we provide some examples of different types of transformers. Then, we describe every model that was used in previous and current experiments.

Embedding. Most models use a pretrained BERT tokenizer for text encoding. For image embedding, there are more different options. Some models use Faster R-CNN [59] to extract region features from images: VisualBERT, ViLBERT, LXMERT, UNITER, ViLLA [26, 10, 11, 34, 35]. Another common approach is to use Vision Transformer (ViT) [60], which is used by CLIP, FLAVA, and ViLT [30, 32, 27].

Architecture. Depending on their architecture, they can mainly be classified into two types: single-stream and dual-stream transformers. On the one hand, in **single-stream** transformers the image and text embeddings are concatenated and then jointly encoded. For instance, the following transformers are single-stream: UNITER, VILLA, VinVL, ViLT and VisualBERT. [34, 35, 36, 27, 26]. On the other hand, **dual-stream** transformers have two separate modality-specific encoders with optional cross-modality fusion. Some examples include: CLIP, FLAVA, UniT, LXMERT and ViLBERT [30, 32, 33, 11, 10].

Cross-Modal Attention. There are different types of multimodal attention as presented in [61]. In **modality-specific attention**, the language and visual input attend to their modality. Every dual-stream transformer that we mentioned uses this type of attention. In **merged attention**, the language and visual input attend to both themselves and the other modality. All single-stream models use merged attention, and some dual-stream transformers use it too. In **co-attention**, the language and visual input only attend to the other modality input. For example, dual-stream models LXMERT and ViLBERT use co-attention.

Pretraining Objectives. Vision-language transformers use a different pretraining objectives including **masked language modeling** (MLM), image-conditioned **language modeling** (LM), **image-text contrastive** learning (ITC), **image-text matching** (ITM). For Winoground, we are mainly interested in models that are trained with ITC or ITM objectives. For example, BLIP [28] is jointly pre-trained with three vision-language objectives: ITC, ITM and LM.

LXMERT. LXMERT [11] consists of three transformer encoders: object relationship encoder, a language encoder, and a cross-modality encoder (see Figure 2.1). The images are represented as a sequence of objects, whereas each sentence is a sequence of words. It combines self-attention and cross-attention layers to generate language, image, and cross-modality representations. The model is pre-trained with five pre-training tasks: masked language modelling, masked object prediction, cross-modality matching, and image question answering.

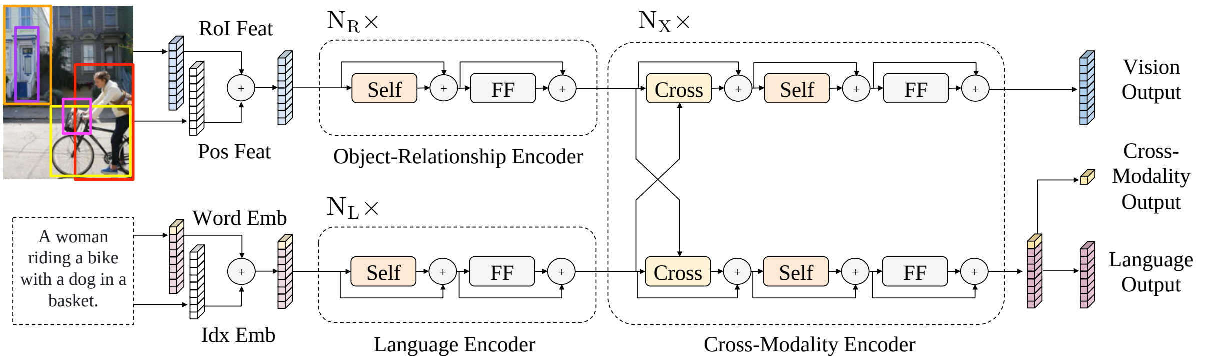


Figure 2.1: The LXMERT model for learning vision-and-language cross-modality representations.

VisualBERT. VisualBERT [26] aims to reuse transformer self-attention to align elements of the input text and regions in the input image (see Figure 2.2). Visual embeddings are constructed by summing visual feature representation, segment embedding and position embeddings. Visual feature representations are obtained from a bounding region object detector. VisualBERT is trained using COCO using two objectives: masked language modelling (MLM) and sentence-image prediction task.

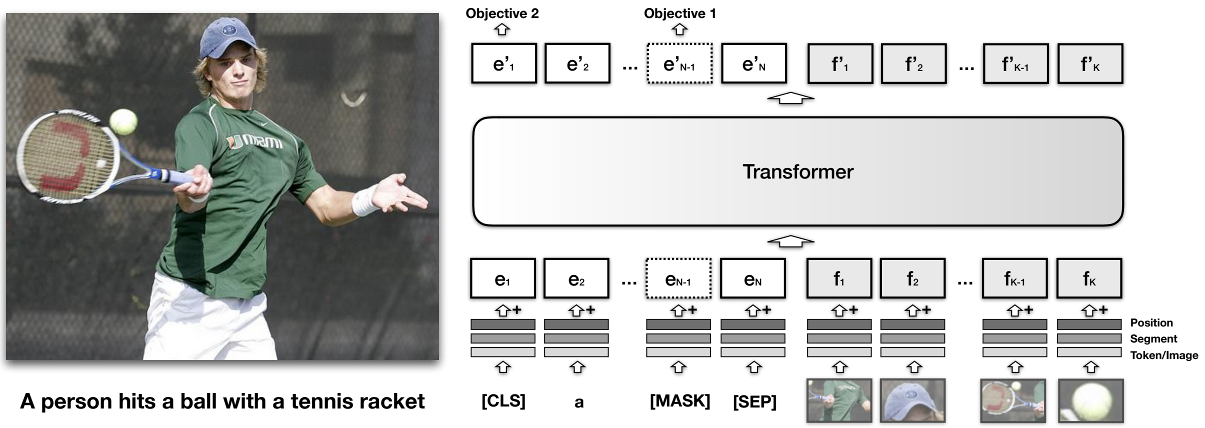


Figure 2.2: The architecture of VisualBERT combines image regions and language with a transformer.

UniT. UniT [33] is a Unified Transformer model to simultaneously learn multiple tasks, such as object detection, natural language understanding and multimodal reasoning (see Figure 2.3). UniT encodes each modality with an encoder and makes predictions on each task with a shared decoder and task-specific output heads. Model parameters are shared across all tasks instead of separately fine-tuning task-specific models.

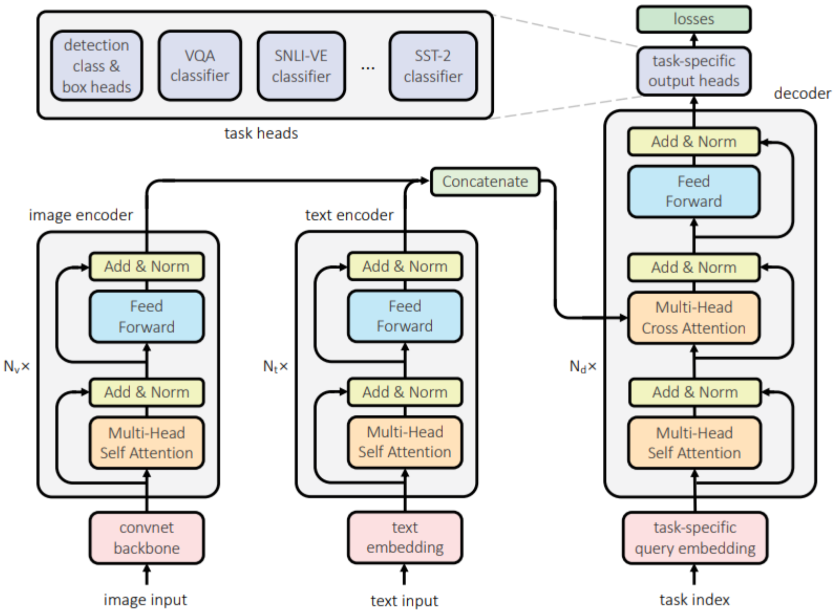


Figure 2.3: An overview UniT, which jointly handles a wide range of tasks in different domains with a unified transformer encoder-decoder architecture.

UNITER. UNITER [34] is a large-scale pre-trained model for joint multimodal embedding (see Figure 2.4). An Image Embedder is used to extract the visual features of each region and a Text Embedder to tokenize the input sentence. It is pre-trained using four image-text datasets: COCO, Visual Genome, Conceptual Captions, and SBU Captions. Four pretraining objectives were designed for this model:

Masked Language Modeling (MLM), Masked Region Modeling (MRM), Image-Text Matching (ITM), and Word-Region Alignment (WRA).

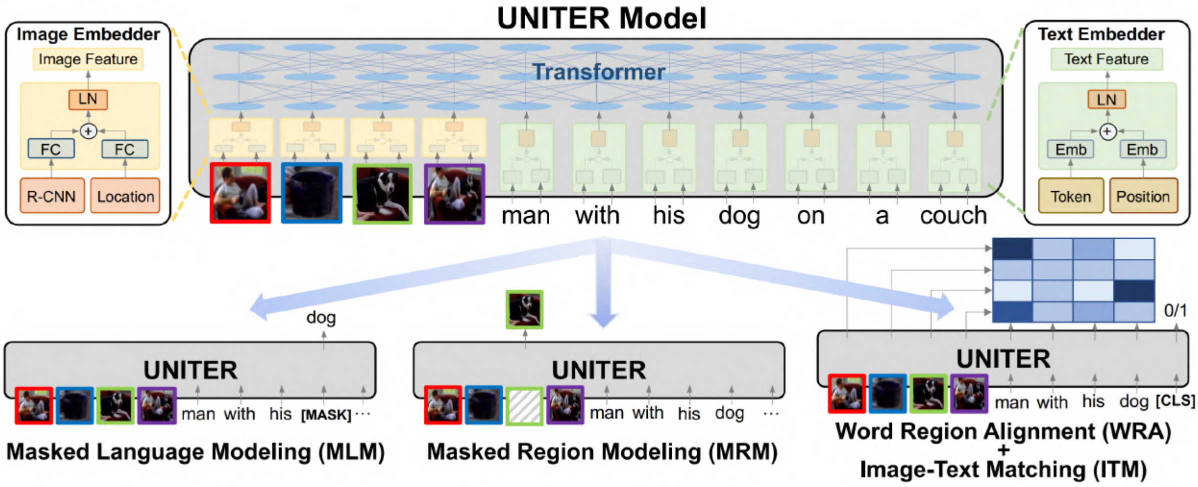


Figure 2.4: Overview of the UNITER model, consisting of an Image Embedder, a Text Embedder and a multi-layer Transformer

VILLA. VILLA [35] is the first known effort on large-scale adversarial training for vision-and-language representation learning (see Figure 2.5). VILLA consists of two training stages: task-agnostic adversarial pre-training and task-specific adversarial finetuning. Instead of adding adversarial perturbations on image pixels and textual tokens, it performs adversarial training in the embedding space of each modality. VILLA achieves SOTA on a wide range of tasks, including VQA, VCR, Image-Text Retrieval, Referring Expression Comprehension, Visual Entailment, and NLVR2.

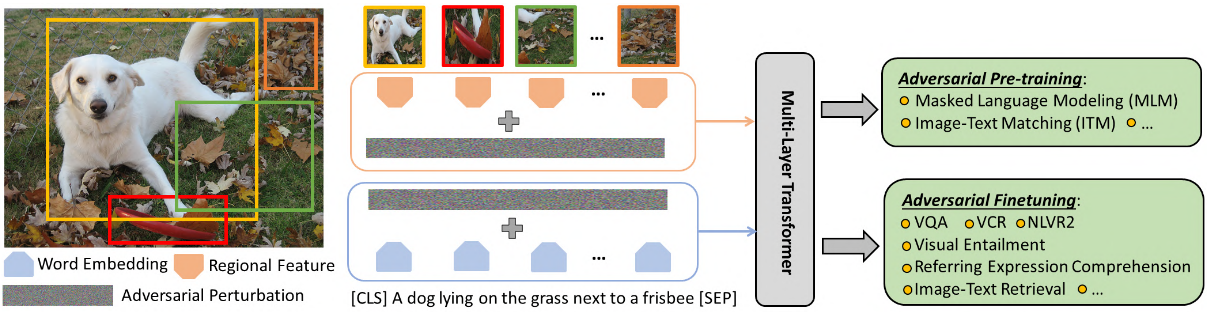

Word Embedding
Regional Feature
Adversarial Perturbation
[CLS] A dog lying on the grass next to a frisbee [SEP]
...
Multi-Layer Transformer
Adversarial Pre-training:
Masked Language Modeling (MLM)
Image-Text Matching (ITM) ...
Adversarial Finetuning:
VQA VCR NLVR2
Visual Entailment
Referring Expression Comprehension
Image-Text Retrieval ...

Figure 2.5: Overview of the VILLA framework for vision-and-language representation learning.

VinVL. VinVL [36] feeds the visual features generated by a new object detection model into a Transformer-based VL fusion model OSCAR [62] (see Figure 2.6). VinVL develops an improved object detection model to provide object-centric representations of images. The new visual features significantly improve the performance across all VL tasks, achieving state-of-the-art results.

ViLBERT. ViLBERT [10] is a BERT-based model for learning task-agnostic joint representations of images and language (see Figure 2.7). ViLBERT extends the BERT architecture to a multi-modal model of two streams, which interact through co-attention transformer layers. ViLBERT is trained on the Conceptual Captions dataset under two training tasks: multi-modal learning and multi-modal alignment prediction.

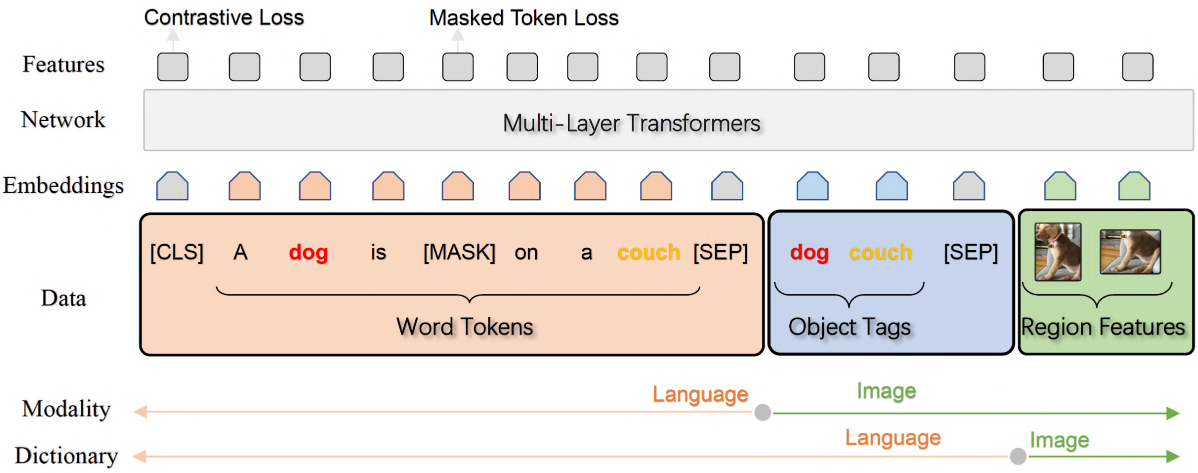


Figure 2.6: OSCAR model architecture that represents the image-text pair as a triple of word tokens, object tags and region features.

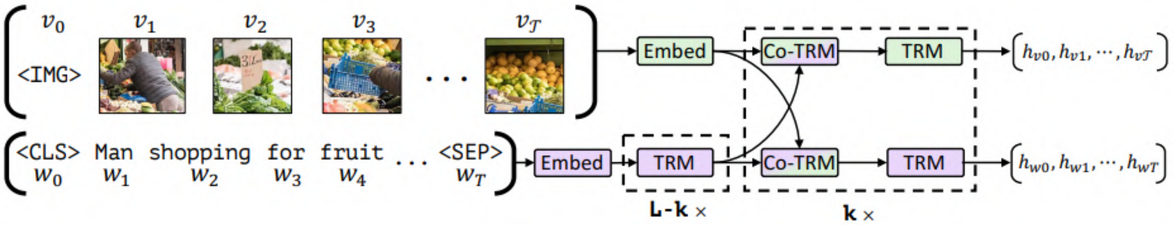

v_0 v_1 v_2 v_3 v_T
<IMG>
<CLS> Man shopping for fruit ... <SEP>
w_0 w_1 w_2 w_3 w_4 w_T
Embed
Co-TRM
TRM
Embed
TRM
Co-TRM
TRM
L-k ×
k ×
$h_{v0}, h_{v1}, \cdots, h_{vT}$
$h_{w0}, h_{w1}, \cdots, h_{wT}$

Figure 2.7: ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through co-attentional transformer layers.

ViLT. ViLT [27] is a minimal vision-and-language pre-training transformer model where the processing of visual inputs is simplified to the same way that text inputs are processed (see Figure 2.8). ViLT requires much less computation than previous VLMs and still gets good performance on downstream tasks. ViLT is pre-trained on the following objectives: image text matching (ITM), masked language modelling (MLM), and word patch alignment (WPA). It is fine-tuned on four downstream tasks: visual question answering (VQA2), visual reasoning (NLVR2) and image-text retrieval (COCO and Flickr30K).

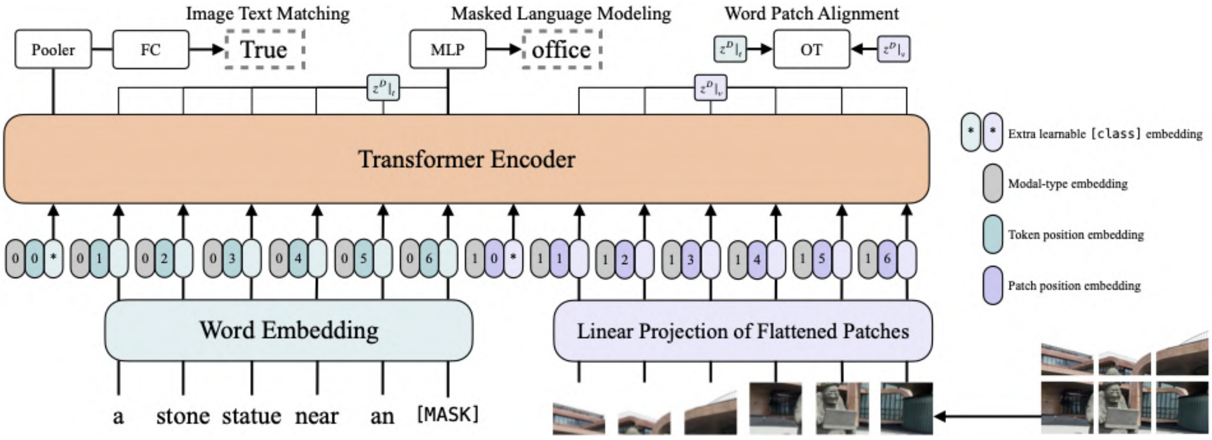


Figure 2.8: ViLT model overview.

FLAVA. FLAVA [32] is a language vision alignment model that learns representations from multimodal and unimodal data. The model consists of three transformers, an image encoder, a text encoder and a multimodal encoder (see Figure 2.9). During pretraining, masked image modelling (MIM) and mask language modelling (MLM), image-text contrastive (ITC), masked multimodal modelling (MMM), and image-text matching (ITM) objectives are used. Classification heads are applied to the outputs from the encoders for visual recognition, language understanding, and multimodal reasoning tasks.

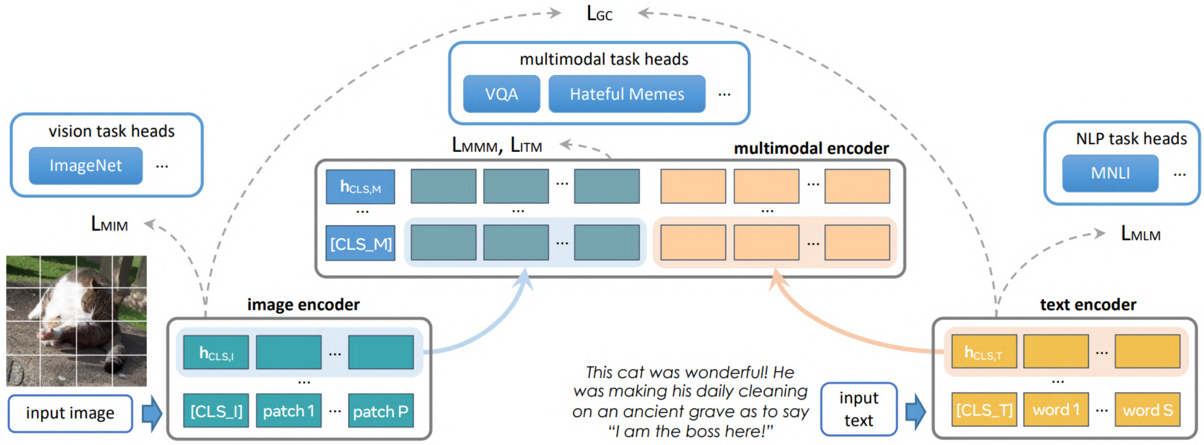
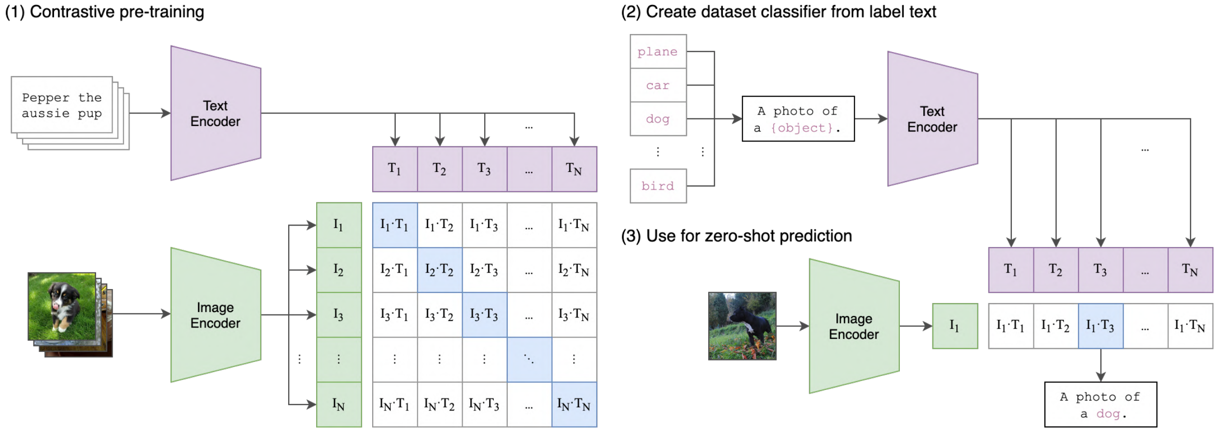


Figure 2.9: FLAVA model overview.

CLIP. CLIP [30] models adopt two unimodal encoders to get image and text representations (see Figure 2.10). CLIP maximizes the similarity between positive image-text pairs, rendering strong unimodal representations. CLIP was trained by OpenAI on a closed dataset of 400M image-text pairs. CLIP variants use different visual backbones, including ViT-B/16, ViT-B/32, ViT-L/14, and ViT-L/14-336.


(1) Contrastive pre-training

(2) Create dataset classifier from label text

(3) Use for zero-shot prediction

Figure 2.10: CLIP model architecture.

OpenCLIP. OpenCLIP [31] models follow the same architecture (see Figure 2.10), but are trained on LAION-2B, a subset of LAION-5B [58] with 2.32 billion English captions. There are different OpenCLIP variants depending on visual backbones: ViT-B/32, ViT-L/14, ViT-H/14, and ViT-g/14. The H/14 model achieves 78.0% zero-shot top-1 accuracy on ImageNet and 73.4% on zero-shot image retrieval at Recall@5 on MS COCO. This makes it the best open-source CLIP model.

BLIP. BLIP [28] achieves state-of-the-art performance on five vision-language tasks: image-text retrieval, image captioning, visual question answering, visual reasoning, and visual dialogue. It employs a Vision Transformer (ViT) [60] as the image encoder and a BERT as the text encoder. BLIP proposes a mixture of encoder-decoder (MED), which can operate either as a unimodal image or text encoder, an image-grounded text encoder, or an image-grounded text decoder (see Figure 2.11). This enables both multimodal understanding and generation. Moreover, BLIP proposes dataset bootstrapping to improve the quality of the pretraining captions by removing noisy ones and generating new ones. BLIP is jointly pretrained with three objectives: language modeling (LM), image-text contrastive learning (ITC) and image-text matching (ITM). There are BLIP variants that use different vision transformers: ViT-B/16 and ViT-L/16. Fine-tuned checkpoints are also available for many downstream tasks.

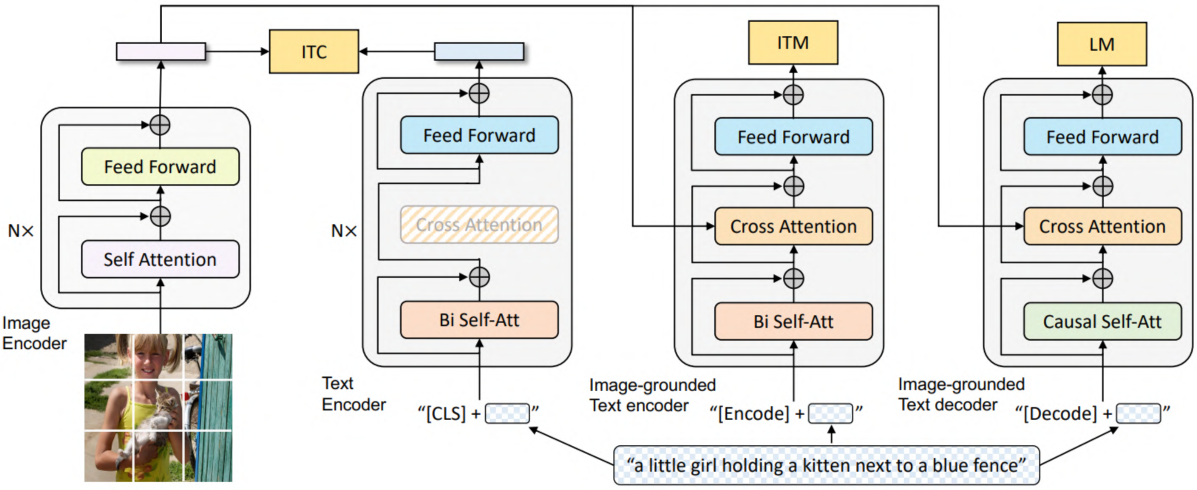


Figure 2.11: BLIP pre-training model architecture: a multimodal mixture of encoder-decoder (MED).

OFA. OFA [29] is a sequence-to-sequence pretrained model that unifies modalities and tasks. It performs a lot of cross-modal and uni-modal tasks, including image generation, visual grounding, image captioning, image classification and language modelling (see Figure 2.12). In contrast with the recent VLMs that require large cross-modal datasets, OFA is pretrained on only 20M publicly available image-text pairs. Despite this, OFA achieves SOTA in various cross-modal tasks and competitive performance on uni-modal tasks.

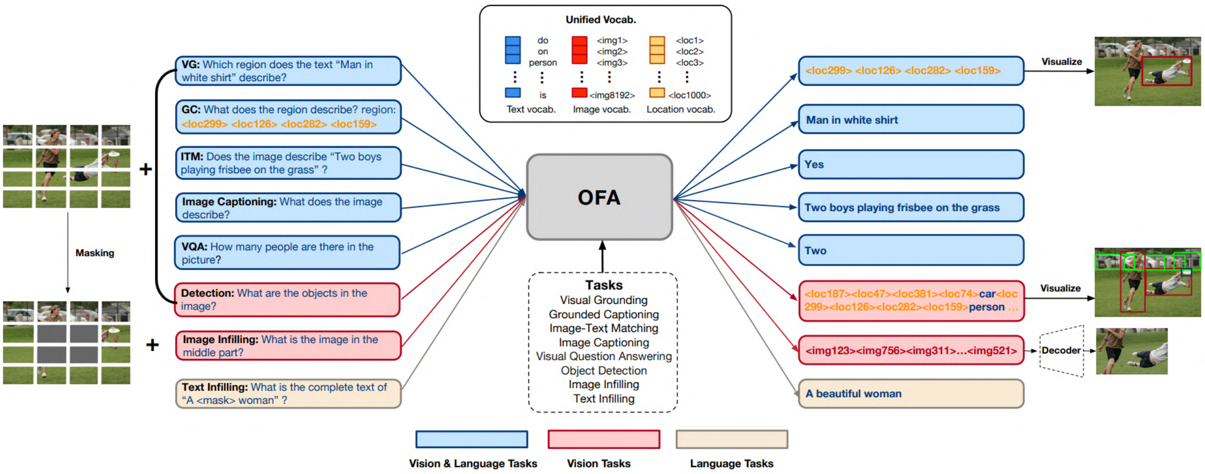

Unified Vocab.
do
on
person
is
Text vocab.
<img1>
<img2>
<img3>
<img8192>
Image vocab.
<loc1>
<loc2>
<loc3>
<loc1000>
Location vocab.
Masking
+
+
VG: Which region does the text "Man in white shirt" describe?
GC: What does the region describe? region: <loc299> <loc126> <loc282> <loc159>
ITM: Does the image describe "Two boys playing frisbee on the grass" ?
Image Captioning: What does the image describe?
VQA: How many people are there in the picture?
Detection: What are the objects in the image?
Image Infilling: What is the image in the middle part?
Text Infilling: What is the complete text of "A <mask> woman" ?
OFA
Tasks
Visual Grounding
Grounded Captioning
Image-Text Matching
Image Captioning
Visual Question Answering
Object Detection
Image Infilling
Text Infilling
<loc299> <loc126> <loc282> <loc159>
Visualize
Man in white shirt
Yes
Two boys playing frisbee on the grass
Two
<loc187><loc47><loc381><loc74>car<loc299><loc126><loc282><loc159>person ...
Visualize
<img123><img756><img311>...<img521>
Decoder
A beautiful woman
Vision & Language Tasks
Vision Tasks
Language Tasks

Figure 2.12: OFA pretraining tasks: visual grounding, grounded captioning, image-text matching, image captioning, VQA, object detection, image infilling and text infilling.

2.1.2 Multimodal RNNs

Multimodal RNNs were the SOTA approach for vision-language tasks before transformers. Two sequence-based models are included in Winoground [9] evaluation, VSE++ [37] and VSRN [38]. Both models minimize the hardest negative score. which is the highest-scoring image-caption pair that is not correct. Both models use a GRU citechung2014gru to get language embeddings.

VSE++. VSE++ [37] uses a new technique for learning visual-semantic embeddings for cross-modal retrieval, and is based on VSE (see Figure 2.13). VSE's image encoder is a linear projection of the embedding from a backbone (either ResNet152 [63] or VGG19 [64]). VSE++ is trained on COCO and Flickr30K datasets, obtaining state-of-the-art results on image-text retrieval.

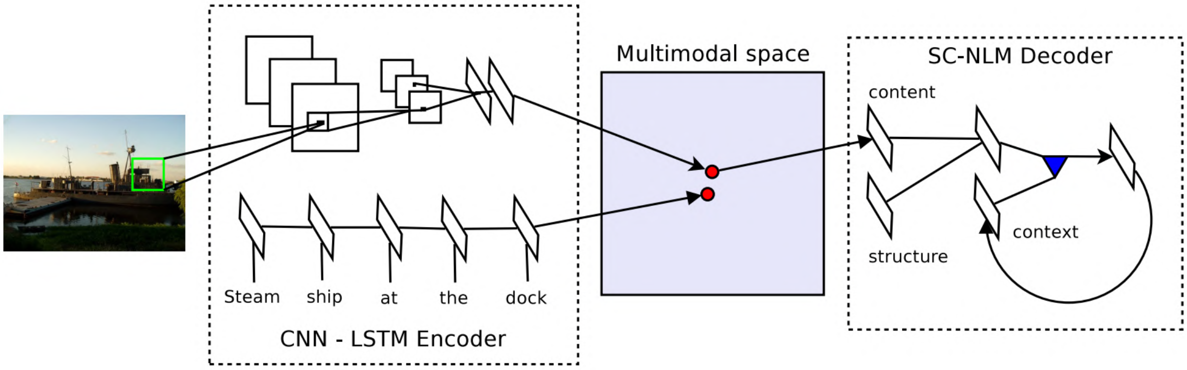


Figure 2.13: VSE model architecture. The encoder is composed of a CNN and an LSTM for learning a joint image-sentence embedding. The decoder is an NLM that combines structure and content vectors for generating words one by one.

VSRN. VSRN [38] is a simple and interpretable reasoning model to generate a visual representation that captures key objects and semantic concepts of a scene (see Figure 2.14). A Faster R-CNN is used to get a sequence of features which are fed into a GRU to obtain image embeddings. VSRN is trained on COCO and Flickr30K datasets, outperforming previous models on image-text retrieval.

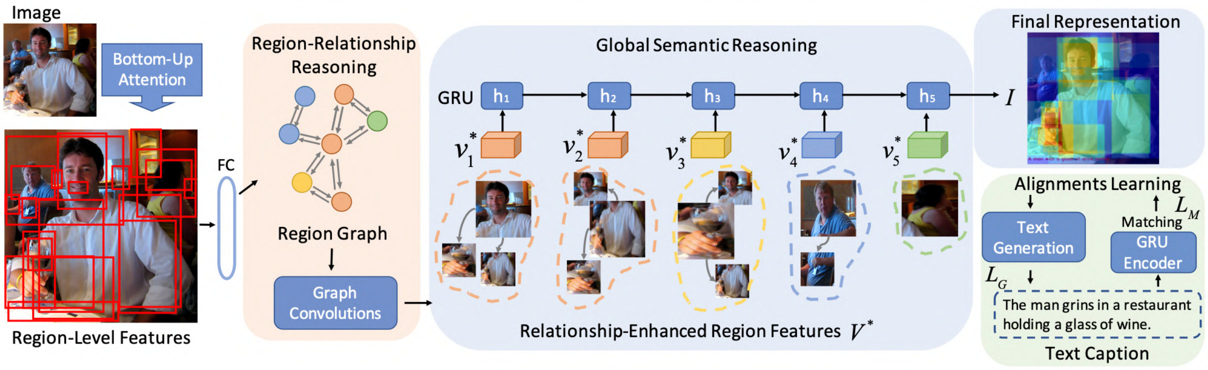


Figure 2.14: An overview of VSRN (Visual Semantic Reasoning Network).

2.1.3 Diffusion Models

Diffusion models are trained to denoise random gaussian noise step by step, to get a sample image. Neural networks are trained to predict a way to slightly denoise the picture in each step. As we can see in Figure 2.15, after a certain number of steps, a sample is obtained.

Diffusion models have obtained SOTA results on image generation. However, one downside of diffusion models is that the reverse denoising process is slow. In addition, these models consume a lot of memory because they work in pixel space. Therefore, it is challenging to train these models and also to use them for inference.

Consequently, most of the recent diffusion models, e.g. DALLE-2 [12] and IMAGEN [13], are unfortunately not accessible to the community. The most popular exception is Stable Diffusion [17], which has been open sourced and can be used on a single GPU.

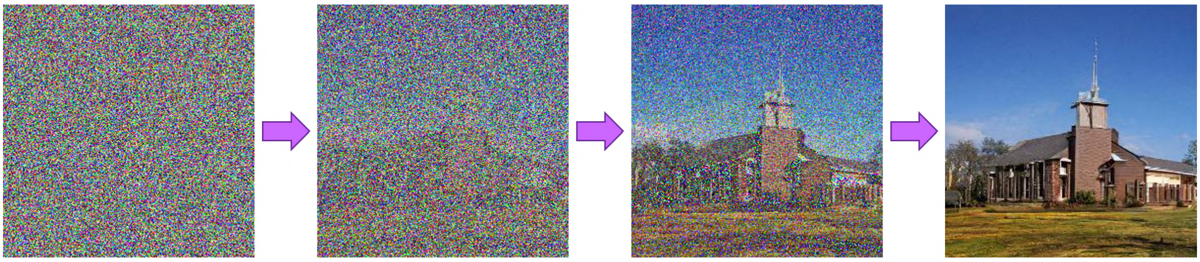


Figure 2.15: In the diffusion process random images are denoised in multiple steps to get a sample image.

Stable Diffusion. Stable Diffusion is based on a type of diffusion model called Latent Diffusion [17]. Latent diffusion reduces the memory and compute complexity by applying the diffusion process over a lower dimensional latent space. There are three main components in latent diffusion: an autoencoder (VAE), a U-Net and a text-encoder (CLIP).

The autoencoder (VAE). The VAE [65] has two parts, an encoder and a decoder, as we can see in Figure 2.16. During latent diffusion training, the encoder maps the images to a latent space for the forward diffusion process, which applies more noise at each step. During inference, the decoder maps the latents generated by the reverse diffusion process back to the images. The encoder and decoder are trained jointly to minimize the reconstruction error.

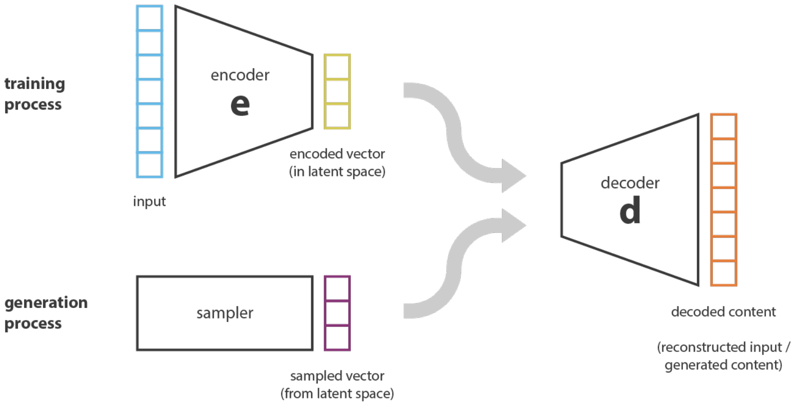

training process
input
encoder
e
encoded vector (in latent space)
decoder
d
decoded content
(reconstructed input / generated content)
generation process
sampler
sampled vector (from latent space)

Figure 2.16: Variational Autoencoder (VAE) training and generation processes.

The U-Net. The U-Net [66] also has an encoder part and a decoder part, as shown in Figure 2.17. The encoder has several ResNet blocks which half the image size by 2. The decoder does the opposite process to upsample the image to the initial size. The U-Net outputs the noise residual which can be used to compute the denoised image representation. To prevent the U-Net from losing important information while downsampling, shortcut connections are usually added from the downsample path to the corresponding layers in the upsample path. Moreover, the output of the stable diffusion U-Net is conditioned on text-embeddings via cross-attention layers.

The text-encoder (CLIP). The CLIP [30] text-encoder transforms the input prompt into an embedding for the U-Net. Stable Diffusion does not train the text-encoder during training and uses an already trained CLIP text encoder.

With the previous components we nearly have the full Stable Diffusion inference architecture Figure 2.18. The stable diffusion model takes a latent seed and a text prompt as input. The latent seed is used to generate initial random latents. The output of the U-Net is used to compute a denoised image representation with a scheduler algorithm. This process is repeated many to get better representations in each iteration. Finally, the latent image representation is decoded by the VAE decoder.

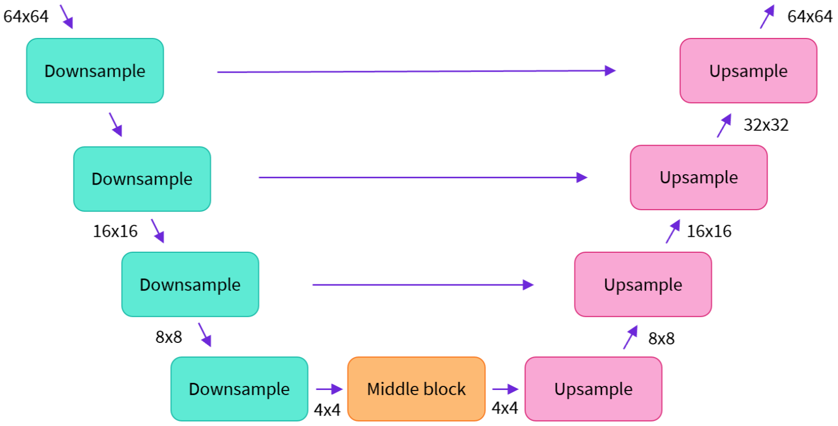


Figure 2.17: The architecture of the U-Net model.

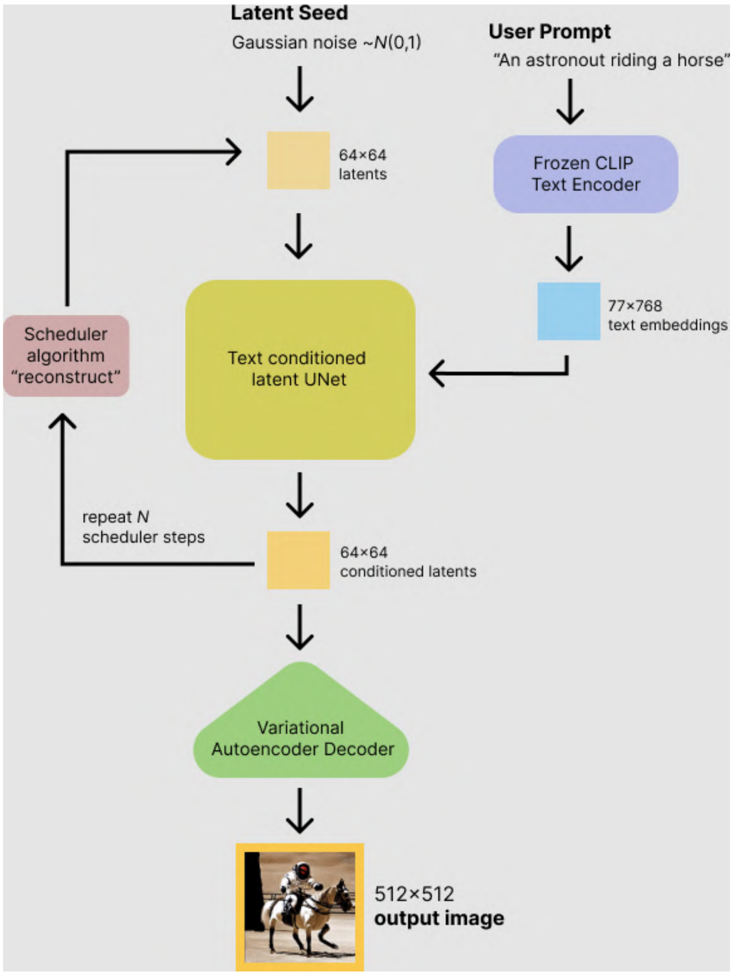


Figure 2.18: Stable Diffusion inference architecture.

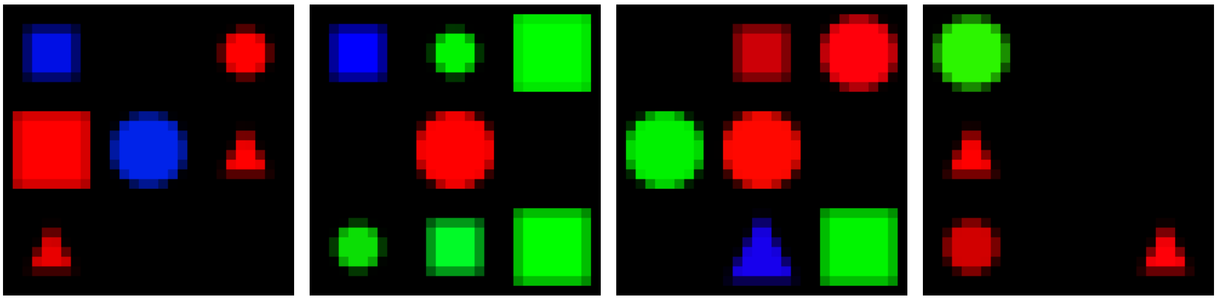
2.2 Visual Reasoning Datasets

This section includes information about visual reasoning datasets. Sections 2.2.1 and 2.2.2 introduce some of the existing Synthetic and Natural Visual Reasoning Datasets. Section 2.2.3 explains the two datasets that we have chosen for Compositional and Spatial Reasoning.

2.2.1 Synthetic Visual Reasoning Datasets

Multimodal training datasets with images and descriptions that include spatial relations tend to be small. Synthetic visual reasoning datasets have been proposed to overcome this problem. These datasets enable full control of dataset generation, easing spatial reasoning capability probing on VLMs. Some examples of synthetic datasets include SHAPES [67], CLEVR [68], NLVR [69] and SPARTQA [70].

SHAPES is a dataset of synthetic images designed to benchmark understanding of spatial and logical relations among multiple objects [67]. The dataset consists of complex yes or no questions about arrangements of colored shapes. Each image is a 3×3 grid of objects. Each object is characterized by shape (circle, square, triangle), colour (red, green, blue) and size (small, big). Figure 2.19 shows some example images and questions.



(a) Q: is a red shape blue? **A:** no

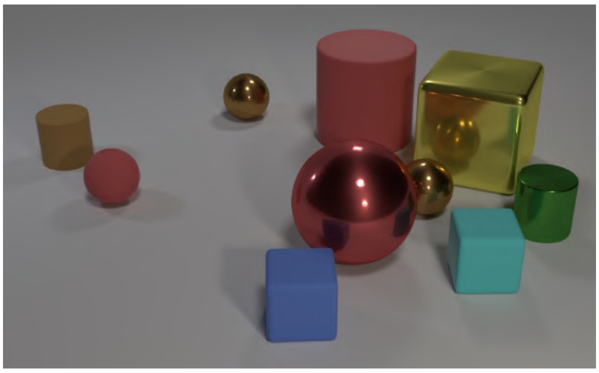
(b) Q: is there a blue shape below a square? **A:** no

(c) Q: is there a red shape above a circle? **A:** yes

(d) Q: is there a red shape below a triangle? **A:** yes

Figure 2.19: Example images, questions and answers from SHAPES.

CLEVR was one of the pioneering works on testing **compositional language and elementary visual reasoning** [68]. However, it presents two major drawbacks: i) questions not only cover spatial grounding but some other concepts such as compositional language and attribute identification, and ii) spatial relations are limited to four, i.e. left, right, behind and in front. A sample image and questions are shown in Figure 2.20.



Q: Are there an equal number of large things and metal spheres? **A:** no

Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere? **A:** small

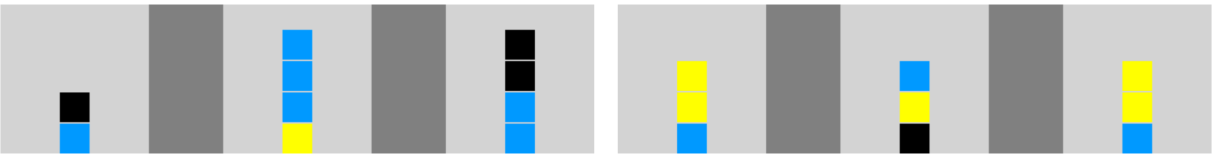
Q: There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere? **A:** no

Q: How many objects are either small cylinders or metal things? **A:** 6

Figure 2.20: A sample image, questions and answers from CLEVR. Questions test aspects of visual reasoning such as attribute identification, counting, comparison, spatial relations, and logical operations.

NLVR contains natural language sentences grounded in images [69]. The task is to determine whether a sentence is true about a visual input. The data was collected through crowdsourcing, and solving the task requires reasoning about sets of objects, comparisons, and spatial relations. Figure 2.21 shows two examples from NLVR.

SPARTQA provides a synthetic **question-answering** dataset that is specially focused on spatial reasoning capabilities [70]. SPARTQA is built on NLVR's images containing more objects with richer spatial structures (Figure 2.22). Questions require deeper reasoning and have four types: *find relation* (FR), *find blocks* (FB), *choose object* (CO), and *yes/no* (YN), which allows for more fine-grained analysis of

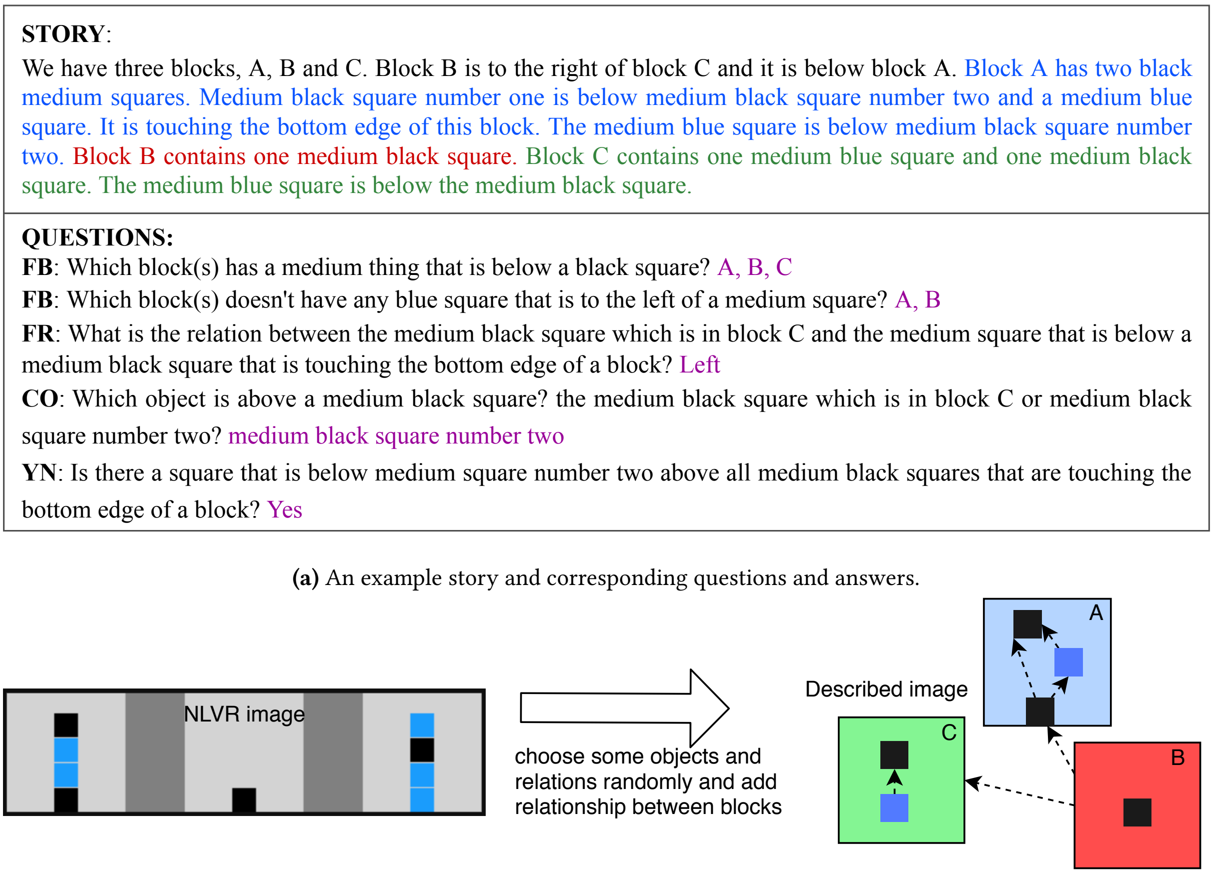


(a) There is at least one tower with four blocks with a yellow block at the base and a blue block below the top block.

(b) There is exactly one tower with a blue block at the base and yellow block at the top.

Figure 2.21: Example sentences and images from NLVR. Each image includes three boxes with different object types. The left sentence is true, while the right is false.

models' capabilities. However, it contains only text and no images, and therefore it does not provide any means to ground spatial concepts.



STORY:
We have three blocks, A, B and C. Block B is to the right of block C and it is below block A. Block A has two black medium squares. Medium black square number one is below medium black square number two and a medium blue square. It is touching the bottom edge of this block. The medium blue square is below medium black square number two. Block B contains one medium black square. Block C contains one medium blue square and one medium black square. The medium blue square is below the medium black square.

QUESTIONS:
FB: Which block(s) has a medium thing that is below a black square? A, B, C
FB: Which block(s) doesn't have any blue square that is to the left of a medium square? A, B
FR: What is the relation between the medium black square which is in block C and the medium square that is below a medium black square that is touching the bottom edge of a block? Left
CO: Which object is above a medium black square? the medium black square which is in block C or medium black square number two? medium black square number two
YN: Is there a square that is below medium square number two above all medium black squares that are touching the bottom edge of a block? Yes

(a) An example story and corresponding questions and answers.

NLVR image
choose some objects and relations randomly and add relationship between blocks
Described image
A
C
B

(b) An example NLVR image and the scene created in Figure 2.22a, where the blocks in the NLVR image are rearranged.

Figure 2.22: Example from SpartQA. We can see an automatically generated story and corresponding questions and answers.

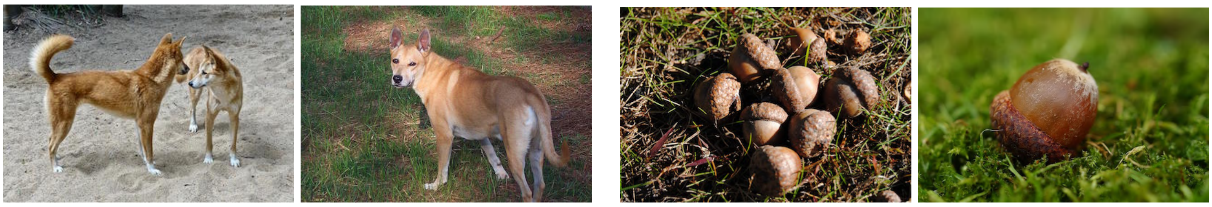
A very recent work proposes a method called **Pseudo-Q** to **automatically create synthetic datasets** that can be used to train visually grounded models [71]. Their method consists of leveraging an off-the-shelf object detector to identify visual objects from unlabeled images, and then creating language queries for these objects that are obtained in an unsupervised fashion with a pseudo-query generation module.

The major drawback of synthetic datasets is that they do not always accurately reflect the challenges of reasoning in the real world. Some aspects that are very important in the real world are not taken into account in synthetic images. For example, the orientations of objects, their context and the viewpoint can affect their spatial relation.

2.2.2 Natural Visual Reasoning Datasets

Many vision-language datasets with natural images also contain spatial relations. For example, NLVR2 [15], MS COCO [39], and VQA [14].

NLVR2 is a dataset for joint reasoning about natural language and images, with a focus on semantic diversity, compositionality, and visual reasoning challenges [15]. There are 9 prevalent linguistic challenges in NLVR2 among which are spatial relations. The examples in Figure 2.23 require addressing challenging semantic phenomena.



(a) The left image contains twice the number of dogs as the right image, and at least two dogs in total are standing.

(b) One image shows exactly two brown acorns in back-to-back caps on green foliage.

Figure 2.23: Two examples from NLVR2, where each caption is paired with two images. The first caption is True and the second one is False.

VQA [14] is a popular vision and language task. Given an image and a question about the image, the task is to provide an accurate answer. VQA is commonly used as a benchmark to evaluate VQA systems. Questions are generally open-ended but multiple choices are provided for some questions. Some examples are shown in Figure 2.24.

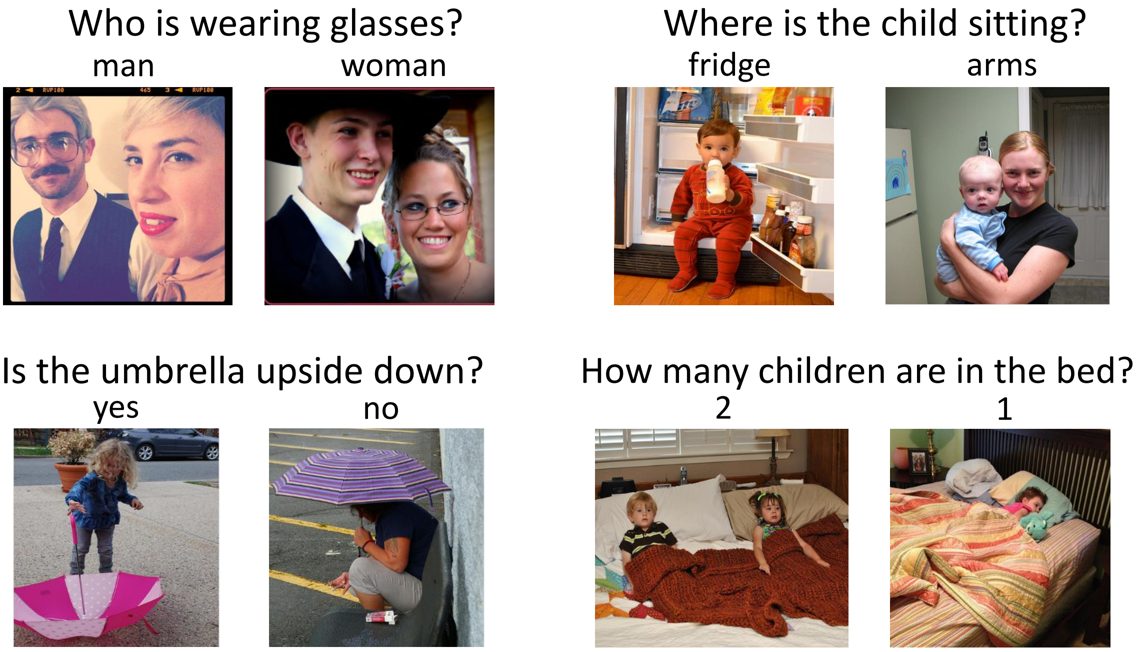

Who is wearing glasses?
man
woman
Where is the child sitting?
fridge
arms
Is the umbrella upside down?
yes
no
How many children are in the bed?
2
1

Figure 2.24: Example images, questions and answers from VQA.

The problem with these datasets is that many different challenges are mixed. Sentences have complex lexical and syntactic information. This makes it hard to identify the exact challenges, preventing categorised analysis.

2.2.3 Compositional and Spatial Reasoning Datasets

To address the problem of mixed challenges, some datasets focus on a single challenge. For instance, **Winoground** [9] focuses on compositional reasoning and **Visual Spatial Reasoning (VSR)** [25] on spatial reasoning. They also contain tags, which enable an in-depth analysis of each visual reasoning challenge. We only provide a short description in this section, but we explain them in detail in the next chapters.

On the one hand, **Winoground** dataset [9] is focused on **evaluating visio-linguistic compositional reasoning** in VLMs. Each instance in the dataset is composed of two images and two captions. Both captions contain a completely identical set of words in a different order. The task is then to match them correctly, which requires the systems to properly deal with composition in natural language. Previous works have shown that language transformers have **difficulties in learning word order** [72, 73]. Winoground provides a means to test whether this is also true for multimodal models. We show some examples in Chapter 3 (Figures 3.1 to 3.3).

On the other hand, **Visual Spatial Reasoning (VSR)** [25], whose objective is to test spatial grounding capabilities by covering 65 different spatial relations over natural images collected from COCO [39]. Given an image, VSR provides a caption which describes a spatial relation between two of the objects that appear in the image. That relation can be real or fake, and that is what the model has to infer. Another advantage of this dataset is that it is annotated by humans. Given its features, we believe VSR is a **good candidate to evaluate spatial grounding in LMs**. We show some examples of VSR in Chapter 4 (Figures 4.1 and 4.2).

3 Winoground Zero-shot Experiments

This chapter describes the Winoground [9] dataset (Section 3.1) and explains the metrics used for evaluation. We also describe a series of previous and new experiments performed over the Winoground dataset using state-of-the-art vision and language models (Section 3.3). The Winoground dataset does not contain a training split, and therefore the experiments are conducted in a zero-shot fashion, where the models are trained on different datasets, and tested on Winoground.

The original Winoground paper included zero-shot experiments with many pre-trained SOTA systems, and they concluded that, surprisingly, none of them does much better than chance [9]. From these experiments, the authors conclude that SOTA models are not as skilled at visio-linguistic compositional reasoning as we might have hoped.

In this work, we extended the previous experiments with new models that obtained better results than those reported in the original paper. In previous experiments, only pre-trained models are tested. We extend this by testing some models that are fine-tuned for specific tasks such as image-text retrieval and visual reasoning. We compare pre-trained versions with fine-tuned versions of the same models and find out that fine-tuning helps.

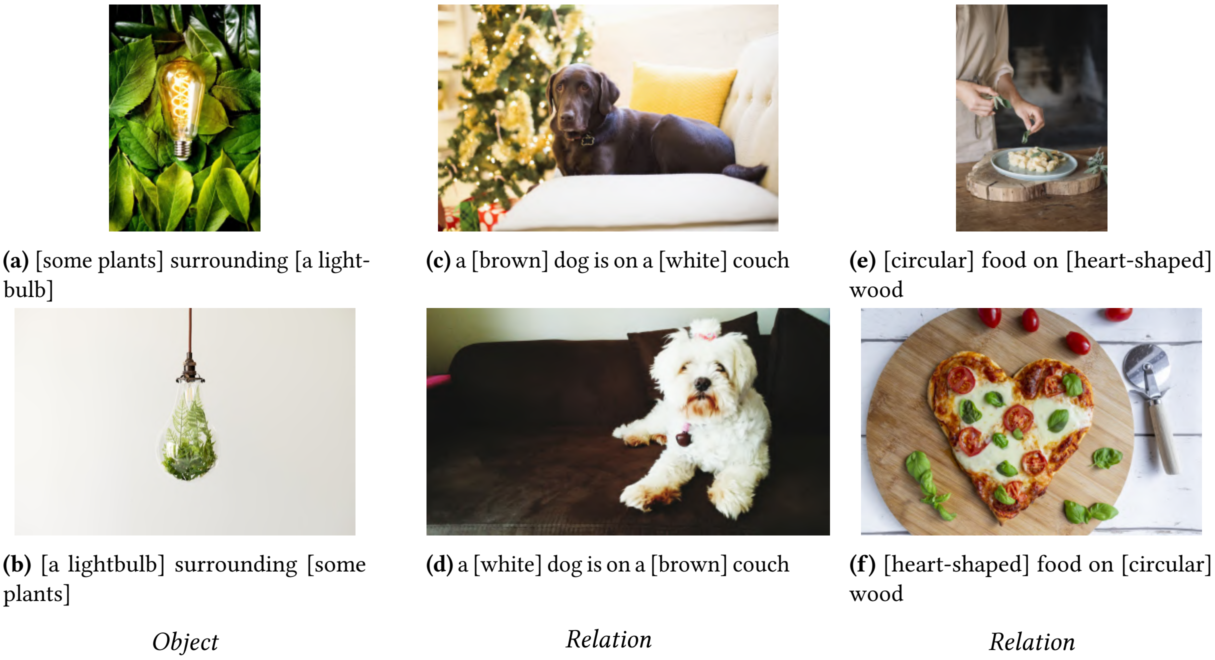
3.1 Winoground Dataset

The Winoground dataset [9] comprises 400 examples that probe different aspects of visio-linguistic compositional reasoning. Each example contains two images and two captions, the goal is to match them correctly. Both captions contain a completely identical set of words or morphemes in a different order. Figures 3.1 to 3.3 show some examples. The dataset was created by expert annotators by designing captions and finding images on Getty Images. All examples are labeled with **linguistic tags** and some include **visual tags**. See Table 3.1 for linguistic and visual tag counts.

Category	Tag	Count
$\text{Linguistic}_{\text{swap-dep.}}$	Object	141
	Relation	233
	Both	26
$\text{Linguistic}_{\text{swap-indep.}}$	1 Main Pred	292
	2 Main Preds	108
Visual	Symbolic	41
	Series	31
	Pragmatics	24

Table 3.1: Linguistic and visual tag counts in the Winoground dataset. Every example has a linguistic tag; only examples that contain visual phenomena have visual tags.

On the one hand, there are 70 **linguistic tags** in total, which can be split into three groups: Object, Relation and Both. **Object** swaps consist in swapping noun phrases that refer to objects. **Relation** swaps reorder words that refer to objects such as verbs, adjectives, prepositions and adverbs. **Both** swaps involve changing both relations and objects. The annotators also tagged examples for **how many main predicates** were in the captions, which is independent of the swap type. See Figures 3.1 and 3.2 for examples of linguistic tags.



(a) [some plants] surrounding [a lightbulb]

(b) [a lightbulb] surrounding [some plants]

Object

(c) a [brown] dog is on a [white] couch

(d) a [white] dog is on a [brown] couch

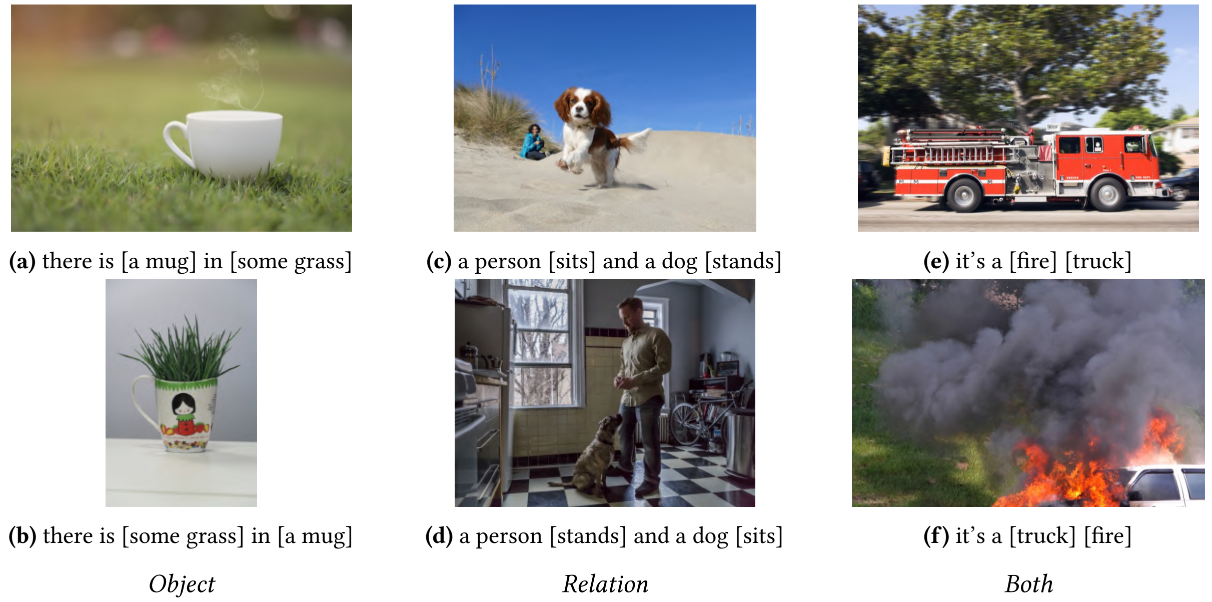
Relation

(e) [circular] food on [heart-shaped] wood

(f) [heart-shaped] food on [circular] wood

Relation

Figure 3.1: Examples from the Winoground dataset for the swap-dependent linguistic tags *Object*, *Relation* and *Relation* from left to right. They are additionally tagged with 1 main predicate.



(a) there is [a mug] in [some grass]

(b) there is [some grass] in [a mug]

Object

(c) a person [sits] and a dog [stands]

(d) a person [stands] and a dog [sits]

Relation

(e) it's a [fire] [truck]

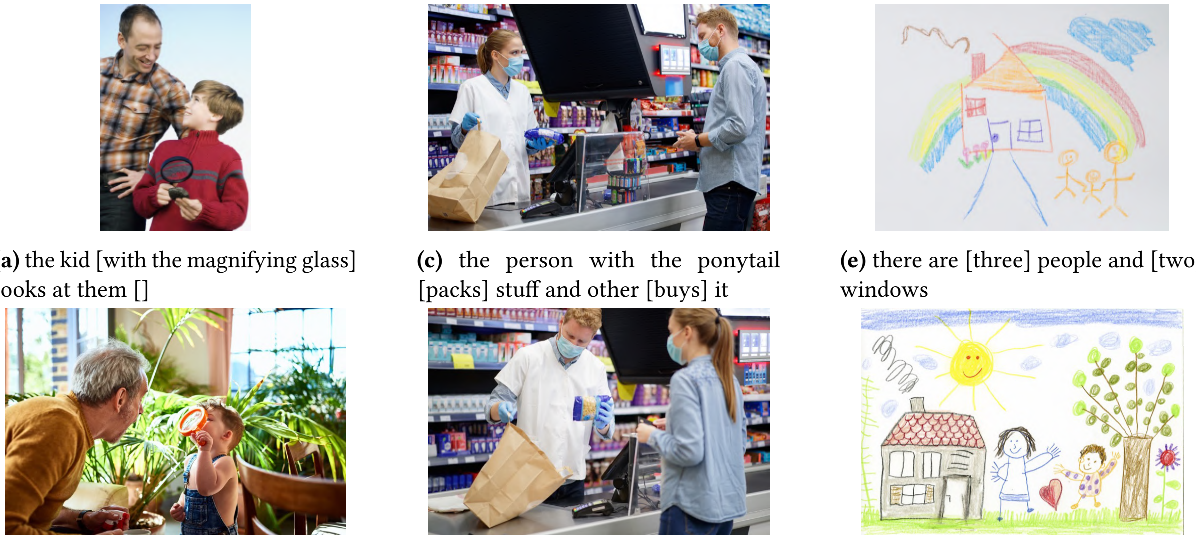
(f) it's a [truck] [fire]

Both

Figure 3.2: Examples from the Winoground dataset for the swap-dependent linguistic tags *Object*, *Relation* and *Both* from left to right. They are additionally tagged with 1, 2 and 1 main predicates from left to right.

On the other hand, there are three non-mutually exclusive **visual reasoning tags**: Pragmatics, Series and Symbolic. **Pragmatics** tag includes images that need to be interpreted non-literally. **Series** tag contains examples where both images come from the same photo series. **Symbolic** tag represents that the images include a symbolic representation. Figure 3.3 shows examples of visual tags.

Winoground is a probing dataset so it prioritizes expert annotations over size. Therefore, there is no training split, all examples are used to evaluate models. The dataset has 400 examples, with 800 unique captions and images. These contain 1600 image-text pairs in total, with 800 correct and 800 incorrect pairings.



(a) the kid [with the magnifying glass] looks at them []

(c) the person with the ponytail [packs] stuff and other [buys] it

(e) there are [three] people and [two] windows

(b) the kid [] looks at them [with the magnifying glass]

(d) the person with the ponytail [buys] stuff and other [packs] it

(f) there are [two] people and [three] windows

Pragmatics

Series

Symbolic

Figure 3.3: Examples from the Winoground dataset for the visual tags *Pragmatics*, *Series* and *Symbolic* from left to right. They are additionally tagged with the *Relation* tag, and 1, 2, and 1 main predicate from left to right.

3.2 Metrics

An example in Winoground is composed of two caption-image pairs: (C_0, I_0) and (C_1, I_1). Metrics must measure models' abilities to match pairs correctly. We compute two types of metrics, **score** and **accuracy**.

Score. Performance on Winoground [9] is computed according to three different score metrics that evaluate different aspects of the models' visio-linguistic reasoning abilities.

The first metric is the **text score**, which measures whether a model can select the correct caption, given an image:

$$ts(C_0, I_0, C_1, I_1) = \begin{cases} 1 & \text{if } s(C_0, I_0) > s(C_1, I_0) \\ & \quad \text{and } s(C_1, I_1) > s(C_0, I_1) \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

The second metric is the **image score**, which measures whether a model can select the correct image, given a caption:

$$is(C_0, I_0, C_1, I_1) = \begin{cases} 1 & \text{if } s(C_0, I_0) > s(C_0, I_1) \\ & \quad \text{and } s(C_1, I_1) > s(C_1, I_0) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Our final metric **group score** combines the previous two, which measures if every combination for a given example is correct:

$$gs(C_0, I_0, C_1, I_1) = \begin{cases} 1 & \text{if } ts(C_0, I_0, C_1, I_1) \\ & \quad \text{and } is(C_0, I_0, C_1, I_1) \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Accuracy. We also add three additional accuracy metrics for additional information. These are similar to the previous ones, but the accuracy is 0.5 when one of the pairs is correct.

The **text accuracy** for an example is computed according to:

$$ta(C_0, I_0, C_1, I_1) = \begin{cases} 1 & \text{if } s(C_0, I_0) > s(C_1, I_0) \\ & \quad \text{and } s(C_1, I_1) > s(C_0, I_1) \\ 0.5 & \text{if } s(C_0, I_0) > s(C_1, I_0) \\ & \quad \text{xor } s(C_1, I_1) > s(C_0, I_1) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

The **image accuracy** for an example is calculated according to:

$$ia(C_0, I_0, C_1, I_1) = \begin{cases} 1 & \text{if } s(C_0, I_0) > s(C_0, I_1) \\ & \quad \text{and } s(C_1, I_1) > s(C_1, I_0) \\ 0.5 & \text{if } s(C_0, I_0) > s(C_0, I_1) \\ & \quad \text{xor } s(C_1, I_1) > s(C_1, I_0) \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

The **group accuracy** in our framework is the mean of both accuracies:

$$ga(C_0, I_0, C_1, I_1) = (ta(C_0, I_0, C_1, I_1) + ia(C_0, I_0, C_1, I_1))/2 \tag{3.6}$$

3.3 Experiments and Results

In this section, we describe a series of previous and new experiments performed over the Winoground dataset using state-of-the-art vision and language models. We compare our results with the previous experiments and with human performance (Section 3.3.1). We also perform analysis by linguistic and visual tags and find that some tags are more challenging for models and others for humans (Sections 3.3.2 and 3.3.3).

3.3.1 Compared To Humans

Previous. We show baseline results from previous experiments [9] in Table 3.2, which includes the following multimodal transformers: CLIP [30], FLAVA [32], LXMERT [11], UniT [33], UNITER [34], VILLA [35], VinVL [36], ViLT [27], VisualBERT [26] and ViLBERT [10]. Several configurations of two types of RNN-based models are also included: VSE++ [37] and VSRN [38].

Human performance was computed using crowd workers on the Amazon Mechanical Turk platform. This establishes a more conservative human baseline than the expert annotator's perfect score [9]. Annotators are shown one image and one caption at a time and have to decide if they match. All 1600 combinations of images and captions are labelled by at least ten annotators. The image-caption score is computed as the ratio of annotators who say that they match.

Table 3.2 shows that there is a large **performance gap** between humans and models. On the one hand, **human performance** is high in all metrics, between 85% and 90% in **scores** and around 93% in **accuracy**. On the other hand, most models perform below random chance in all scores and slightly above random chance in accuracy.

First, only some models are above random chance in **text score**: UNITER, VILLA, VinVL, ViLT, FLAVA, and CLIP. The larger versions of UNITER and VILLA, and VinVL are the models that perform best, and there is still more than 50% difference with human performance.

	Score			Accuracy		
Model	Text	Image	Group	Text	Image	Group
MTurk Human	**89.50**	**88.50**	**85.50**	**93.75**	**93.88**	**93.81**
Random Chance	25.00	25.00	16.67	50.00	50.00	50.00
VinVL	**37.75**	17.75	14.50	**62.75**	**57.75**	**60.25**
UNITER$_{large}$	**38.00**	14.00	10.50	**63.25**	**55.75**	**59.50**
UNITER$_{base}$	**32.25**	13.25	10.00	**60.62**	**55.50**	**58.06**
ViLLA$_{large}$	**37.00**	13.25	11.00	**62.62**	**55.25**	**58.94**
ViLLA$_{base}$	**30.00**	12.00	8.00	**59.62**	**55.00**	**57.31**
VisualBERT$_{base}$	15.50	2.50	1.50	**50.50**	49.88	**50.19**
ViLT (ViT-B/32)	**34.75**	14.00	9.25	**60.50**	**55.38**	**57.94**
LXMERT	19.25	7.00	4.00	**52.12**	**51.88**	**52.00**
ViLBERT$_{base}$	23.75	7.25	4.75	**57.25**	**52.50**	**54.87**
UniT$_{ITM Finetuned}$	19.50	6.25	4.00	**50.25**	**50.75**	**50.50**
FLAVA$_{ITM}$	**32.25**	20.50	14.25	**62.75**	**59.13**	**60.94**
FLAVA$_{ITC}$	**25.25**	13.50	9.00	**59.25**	**55.12**	**57.19**
CLIP (ViT-B/32)	**30.75**	10.50	8.00	**60.38**	**53.25**	**56.81**
VSE++$_{COCO}$ (ResNet)	22.75	8.00	4.00	**51.38**	**50.88**	**51.12**
VSE++$_{COCO}$ (VGG)	18.75	5.50	3.50	**50.38**	49.75	**50.06**
VSE++$_{Flickr30k}$ (ResNet)	20.00	5.00	2.75	**51.50**	**50.25**	**50.88**
VSE++$_{Flickr30k}$ (VGG)	19.75	6.25	4.50	**52.75**	**51.00**	**51.88**
VSRN$_{COCO}$	17.50	7.00	3.75	**50.38**	**51.12**	**50.75**
VSRN$_{Flickr30k}$	20.00	5.00	3.50	**53.25**	**51.75**	**52.50**

Table 3.2: Previous results on the Winoground dataset across the text, image and group score and accuracy metrics. Results above random chance in **bold**.

Second, the performance on **image score** is even worse, where no model performs better than random chance. In contrast, text and image scores in humans are nearly the same. Even the best performing models (FLAVA$_{ITM}$ and VinVL) have nearly 70% difference with the human score.

Last, **group score** is also below random chance for all models, while it is only a bit lower than other scores for humans. Similar to the image score, there is a 70% difference between the human score and the best models, which are again FLAVA$_{ITM}$ and VinVL.

Ours. We show our results in Table 3.3, which includes various configurations of the following multimodal transformers: OFA [29], BLIP [28], CLIP [30], FLAVA [32] and ViLT [27]. OFA and BLIP were not included in the previous experiments. The other models were already included but we test more configurations. For example, we test ViLT models that are finetuned on Flickr30k, COCO, NLVR2 and VSR. All the models except ViLT$_{VSR}$ are already fine-tuned and publicly available.

In the baseline models, only pre-trained models are tested. We extend this by testing some models that are fine-tuned for specific tasks. Those tasks include image-text retrieval and visual reasoning. We compare pre-trained versions with fine-tuned versions of the same models. Our aim is to measure if scores improve by fine-tuning on related tasks.

Depending on the model and setting, the score for an image-text pair is calculated in a different way. For contrastive models, we use cosine similarity between image and text embeddings (CLIP). Other models use the softmaxed probability from the image-text-match classifier (ViLT). BLIP and FLAVA include both options, image-text contrastive (ITC) and image-text matching (ITM) scores. OFA is a generative model, so we have to use the probability of generating that the image and text match. For models fine-tuned on visual reasoning tasks, we take the probability of the True label as a score. Due to its generative nature, we decided to test OFA, hoping that it would have better spatial reasoning skills.

We test 6 different versions of **ViLT**. The first one is the pre-trained version, without finetuning. Two others are finetuned for retrieval on COCO and Flickr30k. The next one is finetuned for visual reasoning on NLVR2. The last two are finetuned on different splits of VSR. The best one is the one trained on NLVR2, which shows that finetuning on that task helps perform better on Winoground. VSR

	Score			Accuracy		
Model	Text	Image	Group	Text	Image	Group
MTurk Human	**89.50**	**88.50**	**85.50**	**93.75**	**93.88**	**93.81**
Random Chance	25.00	25.00	16.67	50.00	50.00	50.00
ViLT (ViT-B/32)	**27.50**	8.75	6.00	**56.88**	**53.12**	**55.00**
ViLT$_{COCO}$ (ViT-B/32)	**32.75**	13.50	11.25	**61.88**	**56.00**	**58.94**
ViLT$_{Flickr30k}$ (ViT-B/32)	**35.00**	11.50	9.75	**61.62**	**54.50**	**58.06**
ViLT$_{NLVR2}$ (ViT-B/32)	**38.00**	15.25	12.00	**58.75**	**55.62**	**57.19**
ViLT$_{VSR}$ Random (ViT-B/32)	**30.50**	14.50	8.00	**59.00**	**55.75**	**57.38**
ViLT$_{VSR}$ Zero-shot (ViT-B/32)	**29.50**	14.00	9.25	**58.38**	**54.75**	**56.56**
FLAVA$_{ITM}$	**32.25**	20.50	14.25	**62.75**	**59.13**	**60.94**
FLAVA$_{ITC}$	**25.25**	13.50	9.00	**59.25**	**55.12**	**57.19**
CLIP (ViT-B/32)	**30.75**	10.25	8.25	**60.38**	**53.12**	**56.75**
CLIP (ViT-B/16)	25.00	10.25	7.00	**57.88**	**53.75**	**55.81**
CLIP (ViT-L/14)	**28.50**	11.00	8.00	**60.38**	**54.62**	**57.50**
CLIP (ViT-L/14-336)	**27.50**	12.00	8.00	**59.38**	**55.12**	**57.25**
OpenCLIP (ViT-B/32)	**35.00**	11.50	8.25	**62.62**	**54.37**	**58.50**
OpenCLIP (ViT-L/14)	**30.25**	11.75	9.00	**59.88**	**54.12**	**57.00**
OpenCLIP (ViT-H/14)	**33.00**	13.00	10.50	**60.38**	**55.75**	**58.06**
OpenCLIP (ViT-g/14)	**31.00**	10.25	7.75	**60.88**	**54.50**	**57.69**
OFA$_{Tiny}$	20.50	8.00	3.75	**53.50**	**52.00**	**52.75**
OFA$_{Base}$	**26.50**	10.50	7.00	**58.88**	**54.00**	**56.44**
OFA$_{Medium}$	22.75	9.00	5.50	**54.25**	**52.75**	**53.50**
OFA$_{Large}$	**26.00**	8.75	5.75	**58.38**	**52.88**	**55.62**
OFA$_{Huge}$	**36.25**	15.50	13.50	**64.38**	**56.62**	**60.50**
BLIP$_{ITM14M}$ (ViT-B/16)	**39.25**	19.00	15.00	**65.88**	**58.25**	**62.06**
BLIP$_{ITC14M}$ (ViT-B/16)	**32.25**	13.75	10.50	**62.25**	**56.50**	**59.38**
BLIP$_{ITM}$ (ViT-B/16)	**40.50**	20.50	16.50	**66.25**	**59.00**	**62.62**
BLIP$_{ITC}$ (ViT-B/16)	**29.75**	14.50	9.50	**59.88**	**56.12**	**58.00**
BLIP$_{ITM}$ (ViT-B/16) (CapFilt-L)	**37.50**	18.50	14.00	**65.00**	**59.13**	**62.06**
BLIP$_{ITC}$ (ViT-B/16) (CapFilt-L)	**31.50**	10.50	8.50	**61.38**	**53.62**	**57.50**
BLIP$_{ITM}$ (ViT-L/16)	**42.50**	18.25	15.50	**66.88**	**57.25**	**62.06**
BLIP$_{ITC}$ (ViT-L/16)	**33.25**	12.00	9.00	**61.75**	**55.00**	**58.38**
BLIP$_{ITMCOCO}$ (ViT-B/16)	**48.00**	24.50	**20.00**	**69.88**	**61.25**	**65.56**
BLIP$_{ITCCOCO}$ (ViT-B/16)	**37.75**	15.75	12.75	**65.00**	**56.88**	**60.94**
BLIP$_{ITMFlickr30k}$ (ViT-B/16)	**46.25**	24.25	**21.25**	**69.25**	**60.62**	**64.94**
BLIP$_{ITCFlickr30k}$ (ViT-B/16)	**38.25**	15.00	12.25	**65.38**	**56.12**	**60.75**
BLIP$_{ITMCOCO}$ (ViT-L/16)	**46.75**	24.00	**20.50**	**68.88**	**61.00**	**64.94**
BLIP$_{ITCCOCO}$ (ViT-L/16)	**37.75**	13.75	10.50	**64.88**	**55.75**	**60.31**
BLIP$_{ITMFlickr30k}$ (ViT-L/16)	**45.00**	24.75	**20.50**	**68.62**	**60.50**	**64.56**
BLIP$_{ITCFlickr30k}$ (ViT-L/16)	**36.00**	16.25	13.50	**63.38**	**56.75**	**60.06**
BLIP$_{NLVR2}$ (ViT-B/16)	**40.25**	25.00	**18.50**	**64.62**	**61.62**	**63.12**

Table 3.3: Our results on the Winoground dataset across the text, image and group score and accuracy metrics. Results above random chance in **bold**.

fine-tuning also increases scores, but not as much as NLVR2. Finetuning for retrieval is also helpful and improves the results of the pre-trained model. The score of the pre-trained model is lower than the baseline one.

For **FLAVA** and **CLIP** we manage to replicate baseline results. We also test 3 other OpenAI CLIP [30] models with different configurations and find that they all perform similar to the baseline configuration. Finally, we test some new **OpenCLIP** [31] models, that were trained on LAION-2B, a subset of LAION-5B [58] with English captions. These models perform slightly better than OpenAI CLIP models.

We test the 5 model sizes of **OFA**. Considering that this model gets state-of-the-art performance on many tasks, the performance is not very good. Even the biggest model is not better than the best baseline model. OFA is trained to generate "yes" or "no" when given an image and the text "Does the image describe <caption>?". This might explain why it does not perform that well on retrieval and Winoground.

We test many configurations of **BLIP**, which include different training sizes, scoring, vision transformer sizes and finetuning datasets. ITM score is better than ITC score in all the cases. Even the 14M **pretrained-only** model is better than all the previously tested models. When compared with the 129M pretrained model, we find that there is only a small difference in performance. This suggests that **pretraining on more data** might not be enough to increase performance. Using a **bigger vision backbone** (ViT-L/16) does not improve results either, getting similar or worse results when compared to ViT-B/16. Finally, using **caption filtering** (CapFilt-L) also provides worse results. This suggests that improving pretraining captions is not

Finetuning BLIP on different tasks increases results significantly. **Finetuning for retrieval** on COCO and Flickr30k improves the results a lot, reaching nearly above random performance in text, image and group scores. **Finetuning on NLVR2** is also very helpful, but a bit less than retrieval. This contrasts with ViLT, which gets the best results when finetuned on NLVR2. The best BLIP scores are much better than previous models, 10% in text score, 4% in image score and 7% in group score.

However, even the best model is still **far from human performance** in text, image and group scores. There is still a 40% gap in text scores, and 64% in image and group scores. If we look at accuracy metrics, the gap is reduced, but the difference is still very big. Image score remains much lower than text score for all the models.

3.3.2 Results By Linguistic Tag

Previous. Table 3.4 shows results from previous experiments [9] by linguistic tags. The highest **human performance** for **swap-dependent linguistic tags** is on **object**, followed by **relation** and then **both**. For the **swap-independent linguistic tags**, humans do better on examples with two main predicates, which tend to be longer and more complicated. We include correlation analysis between caption perplexity and model scores in Appendix A.1. Appendix A.2 includes correlation analysis between pretraining dataset size and model scores.

Models perform much worse in all tags, but they show the **opposite pattern**. They perform better on examples with simpler and shorter sentences which often have morpheme-level swaps. Examples with the **both** tag have some of the shortest and least compositional captions. Many models get better than random performance on this tag, and CLIP even reaches human performance on text score. Image score remains lower than text score for all tags and models.

	Object			Relation			Both			1 Main Pred			2 Main Preds		
Model	Text	Image	Group	Text	Image	Group	Text	Image	Group	Text	Image	Group	Text	Image	Group
MTurk Human	**92.20**	**90.78**	**88.65**	**89.27**	**90.56**	**86.70**	**76.92**	**57.69**	**57.69**	**87.33**	**85.62**	**82.53**	**95.37**	**96.30**	**93.52**
VinVL	**36.88**	17.73	14.18	**37.77**	17.60	14.16	**42.31**	19.23	**19.23**	**39.38**	21.23	**17.47**	**33.33**	8.33	6.48
UNITER$_{large}$	**39.01**	12.77	9.93	**36.05**	14.16	9.87	**50.00**	19.23	**19.23**	**40.07**	16.44	13.36	**32.41**	7.41	2.78
UNITER$_{base}$	**34.04**	11.35	9.22	**30.04**	14.16	10.30	**42.31**	15.38	11.54	**35.27**	14.73	11.99	24.07	9.26	4.63
ViLLA$_{large}$	**36.88**	14.89	11.35	**37.34**	12.88	11.16	**34.62**	7.69	7.69	**39.73**	17.12	14.38	**29.63**	2.78	1.85
ViLLA$_{base}$	**33.33**	15.60	9.93	**27.04**	9.01	6.01	**38.46**	19.23	15.38	**33.22**	14.04	10.27	21.30	6.48	1.85
VisualBERT$_{base}$	19.15	2.13	0.71	12.88	2.15	1.72	19.23	7.69	3.85	16.44	2.74	1.71	12.96	1.85	0.93
ViLT (ViT-B/32)	**31.91**	15.60	9.22	**36.91**	11.59	8.15	**30.77**	**26.92**	**19.23**	**35.27**	17.12	11.64	**33.33**	5.56	2.78
LXMERT	22.70	9.22	6.38	17.60	5.58	2.58	15.38	7.69	3.85	19.18	8.56	5.14	19.44	2.78	0.93
ViLBERT$_{base}$	**29.08**	10.64	7.09	19.31	3.00	1.72	**34.62**	**26.92**	**19.23**	23.97	8.90	5.82	23.15	2.78	1.85
UniT$_{ITM finetuned}$	17.73	5.67	2.13	18.03	4.72	3.43	**42.31**	23.08	**19.23**	21.58	6.85	4.11	13.89	4.63	3.70
FLAVA$_{ITM}$	**31.91**	23.40	14.89	**30.04**	16.31	12.02	**53.85**	**42.31**	**30.77**	**36.30**	24.66	**17.81**	21.30	9.26	4.63
FLAVA$_{ITC}$	23.40	19.15	11.35	23.61	8.58	5.58	**50.00**	**26.92**	**26.92**	**26.37**	16.44	10.62	22.22	5.56	4.63
CLIP (ViT-B/32)	**34.75**	7.80	6.38	22.75	8.58	5.58	**80.77**	**42.31**	**38.46**	**35.27**	13.01	10.27	18.52	3.70	1.85
VSE++$_{COCO}$ (ResNet)	21.99	6.38	1.42	23.61	9.01	5.58	19.23	7.69	3.85	25.00	9.59	4.79	16.67	3.70	1.85
VSE++$_{COCO}$ (VGG)	17.73	2.13	2.13	18.45	7.30	3.86	**26.92**	7.69	7.69	18.49	4.79	2.74	19.44	7.41	5.56
VSE++$_{Flickr30k}$ (ResNet)	20.57	6.38	3.55	18.88	4.29	2.15	**26.92**	3.85	3.85	21.58	6.51	3.42	15.74	0.93	0.93
VSE++$_{Flickr30k}$ (VGG)	17.73	4.96	2.84	19.74	6.87	5.15	**30.77**	7.69	7.69	20.55	6.16	4.79	17.59	6.48	3.70
VSRN$_{COCO}$	15.60	4.96	2.13	18.88	7.73	4.72	15.38	11.54	3.85	17.12	7.19	3.77	18.52	6.48	3.70
VSRN$_{Flickr30k}$	16.31	4.96	2.13	21.03	4.29	3.86	**30.77**	11.54	7.69	20.89	5.82	3.77	17.59	2.78	2.78

Table 3.4: Previous results by linguistic tag. Results above chance are in **bold**.

Ours. Table 3.5 shows results from our experiments by linguistic tags. We see a similar pattern when compared with previous experiments. Models perform worse on more complicated sentences. BLIP is the best model in nearly all tags, performing above chance in most of them. The only exception is CLIP, which is better in **both** tag scores. Even the best BLIP model is far from random chance on image and group scores of examples with **2 main preds**, the tag where human performance is best. Finetuning helps with most tags and ITM remains better than ITC is most tags, with a few exceptions in both tag scores.

	Object			Relation			Both			1 Main Pred			2 Main Preds		
Model	Text	Image	Group	Text	Image	Group	Text	Image	Group	Text	Image	Group	Text	Image	Group
MTurk Human	**92.20**	**90.78**	**88.65**	**89.27**	**90.56**	**86.70**	**76.92**	**57.69**	**57.69**	**87.33**	**85.62**	**82.53**	**95.37**	**96.30**	**93.52**
ViLT (ViT-B/32)	**29.08**	10.64	4.96	**26.18**	7.73	6.44	**30.77**	7.69	7.69	**30.14**	10.62	7.53	20.37	3.70	1.85
ViLT$_{COCO}$ (ViT-B/32)	**33.33**	15.60	12.77	**30.90**	10.73	9.01	**46.15**	**26.92**	**23.08**	**36.64**	15.75	14.04	22.22	7.41	3.70
ViLT$_{Flickr30k}$ (ViT-B/32)	**32.62**	14.89	11.35	**35.62**	8.15	7.73	**42.31**	23.08	**19.23**	**36.99**	14.38	11.99	**29.63**	3.70	3.70
ViLT$_{NLVR2}$ (ViT-B/32)	**39.01**	16.31	14.18	**36.48**	14.59	10.30	**46.15**	15.38	15.38	**39.73**	18.15	15.07	**33.33**	7.41	3.70
ViLT$_{VSR}$ Random (ViT-B/32)	**34.75**	17.73	9.93	**27.47**	11.16	6.01	**34.62**	**26.92**	15.38	**32.53**	16.44	9.59	25.00	9.26	3.70
ViLT$_{VSR}$ Zero-shot (ViT-B/32)	**34.75**	19.86	14.18	**26.18**	10.73	6.44	**30.77**	11.54	7.69	**32.19**	15.75	11.30	22.22	9.26	3.70
FLAVA$_{ITM}$	**31.91**	23.40	14.89	**30.04**	16.31	12.02	**53.85**	**42.31**	**30.77**	**36.30**	24.66	**17.81**	21.30	9.26	4.63
FLAVA$_{ITC}$	23.40	19.15	11.35	23.61	8.58	5.58	**50.00**	**26.92**	**26.92**	**26.37**	16.44	10.62	22.22	5.56	4.63
CLIP (ViT-B/32)	**35.46**	7.80	6.38	22.32	7.73	5.58	**80.77**	**46.15**	**42.31**	**35.62**	13.01	10.62	17.59	2.78	1.85
CLIP (ViT-B/16)	**27.66**	10.64	5.67	19.31	6.44	4.29	**61.54**	**42.31**	**38.46**	**30.14**	11.99	8.90	11.11	5.56	1.85
CLIP (ViT-L/14)	**27.66**	8.51	5.67	**25.75**	9.87	6.44	**57.69**	**34.62**	**34.62**	**30.14**	13.01	9.93	24.07	5.56	2.78
CLIP (ViT-L/14-336)	**32.62**	12.77	9.22	21.03	8.15	4.29	**57.69**	**42.31**	**34.62**	**30.48**	14.04	10.62	19.44	6.48	0.93
OpenCLIP (ViT-B/32)	**39.01**	11.35	9.93	**30.90**	9.44	5.15	**50.00**	**30.77**	**26.92**	**37.33**	13.36	10.27	**28.70**	6.48	2.78
OpenCLIP (ViT-L/14)	**34.04**	12.06	9.22	**25.32**	8.58	6.01	**53.85**	**38.46**	**34.62**	**32.19**	14.73	11.64	25.00	3.70	1.85
OpenCLIP (ViT-H/14)	**39.01**	12.77	10.64	**26.61**	9.44	6.87	**57.69**	**46.15**	**42.31**	**36.30**	17.12	14.38	24.07	1.85	0.00
OpenCLIP (ViT-g/14)	**33.33**	9.93	6.38	**25.75**	8.58	6.44	**65.38**	**26.92**	**26.92**	**35.96**	12.67	9.93	17.59	3.70	1.85
OFA$_{Tiny}$	22.70	6.38	2.13	17.17	6.87	3.43	**38.46**	**26.92**	15.38	23.97	8.22	4.45	11.11	7.41	1.85
OFA$_{Base}$	**25.53**	14.18	7.09	24.46	6.87	5.15	**50.00**	23.08	**23.08**	**28.77**	12.67	8.56	20.37	4.63	2.78
OFA$_{Medium}$	19.86	7.80	4.26	22.32	7.73	4.72	**42.31**	**26.92**	**19.23**	24.32	10.96	6.85	18.52	3.70	1.85
OFA$_{Large}$	**26.24**	10.64	5.67	24.03	5.15	3.86	**42.31**	**30.77**	**23.08**	**29.45**	10.96	7.53	16.67	2.78	0.93
OFA$_{Huge}$	**40.43**	18.44	15.60	**30.90**	11.59	9.87	**61.54**	**34.62**	**34.62**	**39.73**	19.18	**16.78**	**26.85**	5.56	4.63
BLIP$_{ITM14M}$ (ViT-B/16)	**41.84**	23.40	**17.73**	**36.05**	14.59	11.59	**53.85**	**34.62**	**30.77**	**43.84**	23.63	**18.49**	**26.85**	6.48	5.56
BLIP$_{ITC14M}$ (ViT-B/16)	**34.04**	13.48	9.93	**28.33**	12.02	9.44	**57.69**	**30.77**	**23.08**	**37.67**	16.44	13.01	17.59	6.48	3.70
BLIP$_{ITM}$ (ViT-B/16)	**46.10**	22.70	**17.73**	**35.62**	17.60	14.16	**53.85**	**34.62**	**30.77**	**45.89**	**25.34**	**20.55**	**25.93**	7.41	5.56
BLIP$_{ITC}$ (ViT-B/16)	**34.75**	14.18	9.22	**25.32**	13.73	8.58	**42.31**	23.08	**19.23**	**33.56**	16.10	10.62	19.44	10.19	6.48
BLIP$_{ITM}$ (ViT-B/16) (CapFilt-L)	**39.01**	19.86	12.77	**34.76**	15.88	12.45	**53.85**	**34.62**	**34.62**	**41.10**	22.60	**17.12**	**27.78**	7.41	5.56
BLIP$_{ITC}$ (ViT-B/16) (CapFilt-L)	**36.88**	12.77	9.22	**26.18**	8.58	7.30	**50.00**	15.38	15.38	**35.96**	13.36	10.96	19.44	2.78	1.85
BLIP$_{ITM}$ (ViT-L/16)	**41.84**	19.86	**17.02**	**40.77**	16.31	13.73	**61.54**	**26.92**	**23.08**	**45.55**	23.29	**20.21**	**34.26**	4.63	2.78
BLIP$_{ITC}$ (ViT-L/16)	**34.04**	14.18	11.35	**30.90**	9.01	6.01	**50.00**	**26.92**	**23.08**	**36.99**	14.04	10.96	23.15	6.48	3.70
BLIP$_{ITMCOCO}$ (ViT-B/16)	**42.55**	**26.95**	**19.15**	**49.79**	21.89	**19.31**	**61.54**	**34.62**	**30.77**	**48.97**	**29.79**	**24.66**	**45.37**	10.19	7.41
BLIP$_{ITCCOCO}$ (ViT-B/16)	**36.88**	19.15	14.18	**36.05**	11.59	10.30	**57.69**	**34.62**	**26.92**	**41.78**	18.84	15.07	**26.85**	7.41	6.48
BLIP$_{ITMFlickr30k}$ (ViT-B/16)	**49.65**	**28.37**	**22.70**	**42.49**	19.74	**18.45**	**61.54**	**42.31**	**38.46**	**51.03**	**28.42**	**26.03**	**33.33**	12.96	8.33
BLIP$_{ITCFlickr30k}$ (ViT-B/16)	**36.88**	17.02	10.64	**36.48**	12.02	11.16	**61.54**	**30.77**	**30.77**	**40.75**	17.12	13.70	**31.48**	9.26	8.33
BLIP$_{ITMCOCO}$ (ViT-L/16)	**48.94**	**25.53**	**20.57**	**44.64**	22.32	**20.60**	**53.85**	**30.77**	**19.23**	**51.03**	**28.42**	**23.97**	**35.19**	12.04	11.11
BLIP$_{ITCCOCO}$ (ViT-L/16)	**36.88**	14.18	11.35	**36.05**	11.16	7.30	**57.69**	**34.62**	**34.62**	**41.10**	16.44	13.36	**28.70**	6.48	2.78
BLIP$_{ITMFlickr30k}$ (ViT-L/16)	**46.10**	22.70	16.31	**42.06**	24.89	**21.46**	**65.38**	**34.62**	**34.62**	**50.34**	**29.11**	**24.66**	**30.56**	12.96	9.26
BLIP$_{ITCFlickr30k}$ (ViT-L/16)	**39.01**	19.86	15.60	**30.47**	11.59	9.44	**69.23**	**38.46**	**38.46**	**39.38**	20.55	**17.12**	**26.85**	4.63	3.70
BLIP$_{NLVR2}$ (ViT-B/16)	**42.55**	23.40	**19.86**	**36.48**	**25.32**	**16.74**	**61.54**	**30.77**	**26.92**	**42.47**	**27.74**	**21.23**	**34.26**	17.59	11.11

Table 3.5: Our results by linguistic tag. Results above chance are in **bold**.

3.3.3 Results By Visual Tag

Previous. Table 3.6 shows results from previous experiments [9] by visual tags. First, humans and models are especially good at the **symbolic** examples. Some models even get above random chance in image and group scores but are still very far from humans. Next, human performance is very low on the **pragmatics** tag. Ten human crowdworkers probably didn't capture slight pragmatics preferences that the expert linguist annotators agreed on. Models are also quite bad on this tag, text scores are quite good but many of them get 0% scores on image and group scores. There are a few exceptions that get good scores, even getting close to humans. Finally, humans are very good on **series** tag while models do worst on this tag. Similar to pragmatics, many get a 0% group score. This means that models are always choosing the same image regardless of the caption. This is understandable because images from the same series tend to be very similar between them.

Ours. Table 3.7 shows results from our experiments by visual tags. We see a similar pattern when compared with previous experiments. BLIP is again the best model in all tags, performing above chance in most of them. However, even the best BLIP model is far from random chance on image and group scores of examples with **series** tag. It also struggles a lot in **pragmatics**, where there are only a few exceptions that surpass random chance in group score. Finetuning helps with most tags and ITM remains better than ITC, with big differences in most cases.

	Symbolic			Pragmatics			Series		
Model	Text	Image	Group	Text	Image	Group	Text	Image	Group
MTurk Human	**96.43**	**92.86**	**92.86**	**58.82**	**41.18**	**41.18**	**95.65**	**91.30**	**91.30**
VinVL	25.00	17.86	14.29	**29.41**	5.88	5.88	**34.78**	17.39	13.04
UNITER$_{large}$	**39.29**	**28.57**	**17.86**	**35.29**	0.00	0.00	4.35	8.70	0.00
UNITER$_{base}$	**46.43**	14.29	14.29	**29.41**	17.65	11.76	8.70	8.70	0.00
ViLLA$_{large}$	**39.29**	14.29	10.71	17.65	0.00	0.00	17.39	4.35	0.00
ViLLA$_{base}$	**42.86**	17.86	14.29	**29.41**	5.88	5.88	13.04	8.70	4.35
VisualBERT$_{base}$	**28.57**	0.00	0.00	5.88	0.00	0.00	13.04	0.00	0.00
ViLT (ViT-B/32)	**28.57**	17.86	10.71	**35.29**	0.00	0.00	**26.09**	0.00	0.00
LXMERT	**28.57**	3.57	3.57	17.65	5.88	0.00	8.70	4.35	0.00
ViLBERT$_{base}$	**28.57**	10.71	7.14	**29.41**	5.88	5.88	13.04	0.00	0.00
UniT$_{ITM finetuned}$	14.29	10.71	7.14	17.65	5.88	5.88	21.74	4.35	4.35
FLAVA$_{ITM}$	25.00	**28.57**	**17.86**	17.65	**29.41**	11.76	17.39	8.70	0.00
FLAVA$_{ITC}$	17.86	10.71	10.71	11.76	23.53	5.88	17.39	4.35	4.35
CLIP (ViT-B/32)	**39.29**	3.57	3.57	**35.29**	5.88	5.88	8.70	0.00	0.00
VSE++$_{COCO}$ (ResNet)	**32.14**	10.71	10.71	23.53	11.76	0.00	13.04	4.35	4.35
VSE++$_{COCO}$ (VGG)	17.86	14.29	7.14	17.65	0.00	0.00	13.04	4.35	4.35
VSE++$_{Flickr30k}$ (ResNet)	21.43	3.57	0.00	23.53	0.00	0.00	17.39	4.35	0.00
VSE++$_{Flickr30k}$ (VGG)	**28.57**	10.71	10.71	11.76	0.00	0.00	13.04	4.35	0.00
VSRN$_{COCO}$	7.14	3.57	0.00	11.76	0.00	0.00	13.04	0.00	0.00
VSRN$_{Flickr30k}$	21.43	3.57	3.57	**35.29**	11.76	5.88	8.70	4.35	4.35

Table 3.6: Previous results by visual tag. Results above chance are in **bold**.

	Symbolic			Pragmatics			Series		
Model	Text	Image	Group	Text	Image	Group	Text	Image	Group
MTurk Human	**96.43**	**92.86**	**92.86**	**58.82**	**41.18**	**41.18**	**95.65**	**91.30**	**91.30**
ViLT (ViT-B/32)	21.43	7.14	3.57	17.65	5.88	5.88	17.39	8.70	4.35
ViLT$_{COCO}$ (ViT-B/32)	21.43	10.71	10.71	**29.41**	17.65	5.88	21.74	8.70	4.35
ViLT$_{Flickr30k}$ (ViT-B/32)	**28.57**	7.14	7.14	23.53	0.00	0.00	**26.09**	4.35	4.35
ViLT$_{NLVR2}$ (ViT-B/32)	**42.86**	10.71	10.71	**41.18**	0.00	0.00	17.39	13.04	4.35
ViLT$_{VSR}$ Random (ViT-B/32)	**28.57**	14.29	7.14	**29.41**	11.76	5.88	**30.43**	21.74	8.70
ViLT$_{VSR}$ Zero-shot (ViT-B/32)	25.00	10.71	7.14	**35.29**	23.53	11.76	**30.43**	8.70	0.00
FLAVA$_{ITM}$	25.00	**28.57**	**17.86**	17.65	**29.41**	11.76	17.39	8.70	0.00
FLAVA$_{ITC}$	17.86	10.71	10.71	11.76	23.53	5.88	17.39	4.35	4.35
CLIP (ViT-B/32)	**35.71**	3.57	3.57	**35.29**	5.88	5.88	13.04	0.00	0.00
CLIP (ViT-B/16)	21.43	3.57	3.57	**29.41**	11.76	11.76	4.35	4.35	0.00
CLIP (ViT-L/14)	**28.57**	10.71	3.57	23.53	17.65	11.76	13.04	8.70	4.35
CLIP (ViT-L/14-336)	**28.57**	14.29	7.14	17.65	17.65	5.88	13.04	4.35	0.00
OpenCLIP (ViT-B/32)	**35.71**	10.71	10.71	23.53	5.88	5.88	17.39	4.35	0.00
OpenCLIP (ViT-L/14)	**32.14**	3.57	3.57	**35.29**	11.76	5.88	8.70	4.35	0.00
OpenCLIP (ViT-H/14)	**35.71**	10.71	10.71	**35.29**	**29.41**	**17.65**	17.39	0.00	0.00
OpenCLIP (ViT-g/14)	**39.29**	10.71	7.14	**41.18**	17.65	11.76	21.74	0.00	0.00
OFA$_{Tiny}$	21.43	7.14	7.14	11.76	17.65	0.00	21.74	8.70	0.00
OFA$_{Base}$	**28.57**	10.71	10.71	23.53	5.88	5.88	21.74	13.04	4.35
OFA$_{Medium}$	**28.57**	10.71	7.14	17.65	5.88	5.88	13.04	8.70	4.35
OFA$_{Large}$	**28.57**	14.29	10.71	**29.41**	0.00	0.00	13.04	0.00	0.00
OFA$_{Huge}$	**39.29**	14.29	14.29	11.76	11.76	5.88	17.39	4.35	4.35
BLIP$_{ITM14M}$ (ViT-B/16)	**46.43**	17.86	**17.86**	**35.29**	11.76	11.76	17.39	4.35	0.00
BLIP$_{ITC14M}$ (ViT-B/16)	**32.14**	14.29	10.71	**29.41**	0.00	0.00	13.04	0.00	0.00
BLIP$_{ITM}$ (ViT-B/16)	**50.00**	17.86	**17.86**	**29.41**	5.88	5.88	13.04	4.35	0.00
BLIP$_{ITC}$ (ViT-B/16)	**39.29**	10.71	7.14	5.88	11.76	0.00	4.35	8.70	0.00
BLIP$_{ITM}$ (ViT-B/16) (CapFilt-L)	**42.86**	17.86	14.29	23.53	17.65	**17.65**	17.39	4.35	0.00
BLIP$_{ITC}$ (ViT-B/16) (CapFilt-L)	**42.86**	0.00	0.00	17.65	0.00	0.00	4.35	0.00	0.00
BLIP$_{ITM}$ (ViT-L/16)	**53.57**	25.00	**25.00**	**29.41**	5.88	0.00	**26.09**	4.35	0.00
BLIP$_{ITC}$ (ViT-L/16)	**39.29**	17.86	14.29	**41.18**	11.76	11.76	8.70	4.35	4.35
BLIP$_{ITMCOCO}$ (ViT-B/16)	**53.57**	17.86	**17.86**	**58.82**	17.65	**17.65**	**39.13**	8.70	0.00
BLIP$_{ITCCOCO}$ (ViT-B/16)	25.00	10.71	7.14	**35.29**	5.88	5.88	17.39	8.70	4.35
BLIP$_{ITMFlickr30k}$ (ViT-B/16)	**53.57**	21.43	**21.43**	**35.29**	11.76	11.76	**26.09**	4.35	4.35
BLIP$_{ITCFlickr30k}$ (ViT-B/16)	**35.71**	10.71	10.71	23.53	17.65	11.76	17.39	4.35	0.00
BLIP$_{ITMCOCO}$ (ViT-L/16)	**39.29**	**35.71**	**25.00**	**58.82**	23.53	**17.65**	**26.09**	4.35	0.00
BLIP$_{ITCCOCO}$ (ViT-L/16)	**46.43**	14.29	14.29	17.65	5.88	5.88	13.04	0.00	0.00
BLIP$_{ITMFlickr30k}$ (ViT-L/16)	**39.29**	**28.57**	**25.00**	**47.06**	11.76	5.88	**30.43**	8.70	4.35
BLIP$_{ITCFlickr30k}$ (ViT-L/16)	**39.29**	14.29	14.29	**47.06**	5.88	5.88	21.74	13.04	13.04
BLIP$_{NLVR2}$ (ViT-B/16)	**57.14**	21.43	10.71	**41.18**	5.88	5.88	21.74	17.39	4.35

Table 3.7: Our results by visual tag. Results above chance are in **bold**.

4 VSR Zero-shot Experiments

In the previous chapter, we have explained zero-shot experiments on Winoground, a dataset for visio-linguistic compositional reasoning. We managed to improve previous results, but there is still a large performance gap between humans and models. In this chapter, we will focus on Visual Spatial Reasoning [25], a dataset for spatial reasoning (Section 4.1). Unlike Winoground, VSR contains training and validation splits and can be used to train models (Section 4.2). However, we mainly focus on zero-shot experiments with models that are trained on other datasets. We explain previous and new experiments we performed and the results we obtained in VSR (Section 4.3).

4.1 VSR Dataset

The objective of VSR is to **test spatial grounding** capabilities by covering **65 spatial relations** over natural images from COCO. Given an image and a caption which describes a spatial relation between two objects, the model has to infer if the relation is true or false.

A **contrastive caption generation** approach was used in VSR to avoid choosing too many trivial relations. First, a pair of images that contain the same two concepts are selected from COCO. Second, an annotator had to choose a spatial relation that made the caption template correct for one image but incorrect for the other. Finally, every item is reviewed by at least two additional human annotators. If the agreement between annotators is not high enough, the data point is excluded.

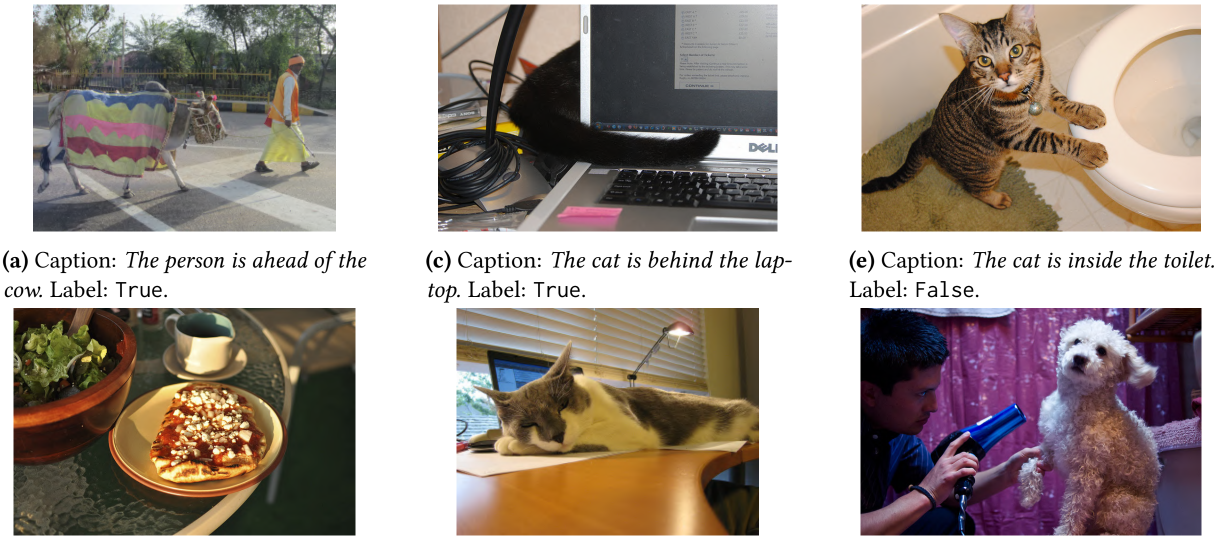
To get a more high-level understanding of the relations, they are grouped in **meta categories** [74]: Adjacency, Directional, Orientation, Projective, Proximity, Topological and Unallocated (see Table 4.1). We show some examples to understand the differences between relation categories in Figure 4.1.

Category	Spatial Relations
Adjacency	Adjacent to, alongside, at the side of, at the right side of, at the left side of, attached to, at the back of, ahead of, against, at the edge of
Directional	Off, past, toward, down, deep down*, up*, away from, along, around, from*, into, to*, across, across from, through*, down from
Orientation	Facing, facing away from, parallel to, perpendicular to
Projective	On top of, beneath, beside, behind, left of, right of, under, in front of, below, above, over, in the middle of
Proximity	By, close to, near, far from, far away from
Topological	Connected to, detached from, has as a part, part of, contains, within, at, on, in, with, surrounding, among, consists of, out of, between, inside, outside, touching
Unallocated	Beyond, next to, opposite to, after*, among, enclosed by

Table 4.1: The available 71 spatial relations. 65 of them appear in the final dataset. Relations with $*$ are not used.

In Figure 4.1 we show examples of Adjacency, Projective and Topological meta categories. **Adjacency** examples involve identifying what is ahead of the cow and which is the edge of the table. The **Projective** images are paired with the same caption, but have different labels. **Topological** examples require understanding what being inside and touching are.

In Figure 4.2 Adjacency, Projective and Orientation meta categories. The first **Adjacency** example is tricky, it requires knowing which is the right side of the bench. The second one is even more difficult because the cow both the cow appears in the car's side mirror. **Projective** examples involve knowing where is the front of the person and below the cat. **Orientation** examples require understanding the orientations of the hair drier and the fire hydrant.



(a) Caption: *The person is ahead of the cow.* Label: `True`.

(b) Caption: *The pizza is at the edge of the dining table.* Label: `True`.

(c) Caption: *The cat is behind the laptop.* Label: `True`.

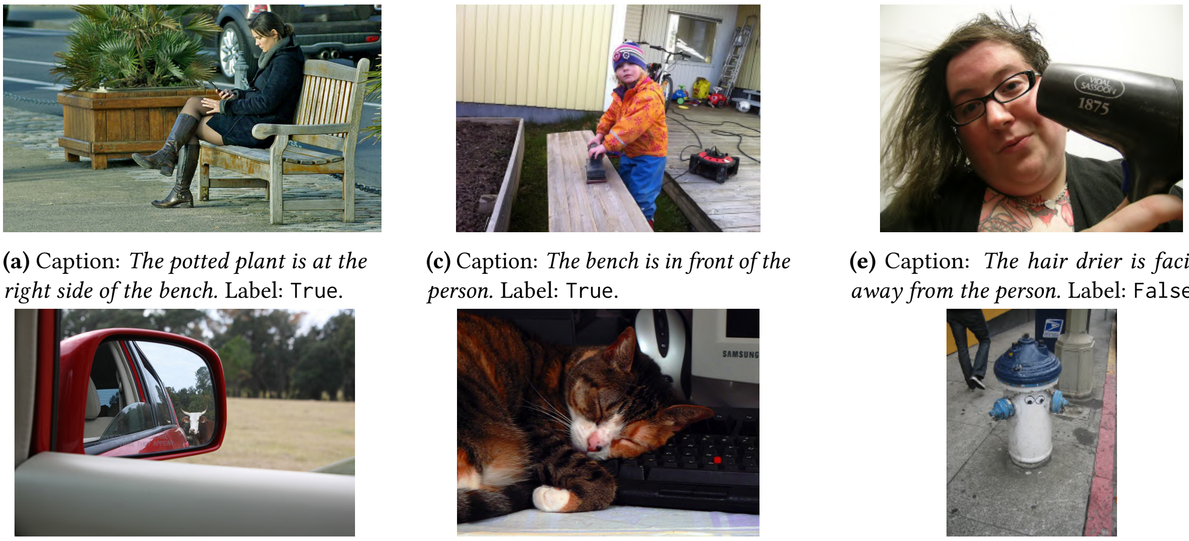
(d) Caption: *The cat is behind the laptop.* Label: `False`.

(e) Caption: *The cat is inside the toilet.* Label: `False`.

(f) Caption: *The person is touching the hair drier.* Label: `True`.

Adjacency *Projective* *Topological*

Figure 4.1: Examples from the VSR dataset for the relation meta categories *Adjacency*, *Projective* and *Topological* from left to right.



(a) Caption: *The potted plant is at the right side of the bench.* Label: `True`.

(b) Caption: *The cow is at the back of the car.* Label: `True`.

(c) Caption: *The bench is in front of the person.* Label: `True`.

(d) Caption: *The keyboard is below the cat.* Label: `True`.

(e) Caption: *The hair drier is facing away from the person.* Label: `False`.

(f) Caption: *The fire hydrant is facing away from the person.* Label: `True`.

Adjacency *Projective* *Orientation*

Figure 4.2: Examples from the VSR dataset for the relation meta categories *Adjacency*, *Projective* and *Orientation* from left to right.

4.2 Dataset Splits

The VSR dataset has two types of splits [25], random and zero-shot. The statistics of the two splits are shown in Table 4.2.

Random split. The dataset is split randomly into train/dev/test with a ratio of 70%/10%/20%. All the validated data points are used in this split.

split	train	dev	test	total
random	7,083	1,012	2,024	10,119
zero-shot	5,440	259	731	6,430

Table 4.2: Data statistics of the *random* and *zero-shot* splits.

Zero-shot split. It is a concept zero-shot split where train/dev/test have no overlapping concepts. That is, each concept can only appear in one of the sets. This is done by randomly grouping concepts into three sets with a ratio of 50%/20%/30%. This is a more challenging setup because the model has to learn concepts and relations in a compositional way instead of remembering the co-occurrence of the two. Moreover, having less training data is a disadvantage for the models, since not all the data can be used in this setting.

4.3 Experiments and Results

In this section, we describe a series of previous and new experiments performed over VSR dataset using state-of-the-art vision and language models. We compare our results with the previous experiments and with human performance (Section 4.3.1). We also perform analysis by relation and relation meta category (Sections 4.3.2 and 4.3.3).

4.3.1 Compared To Humans

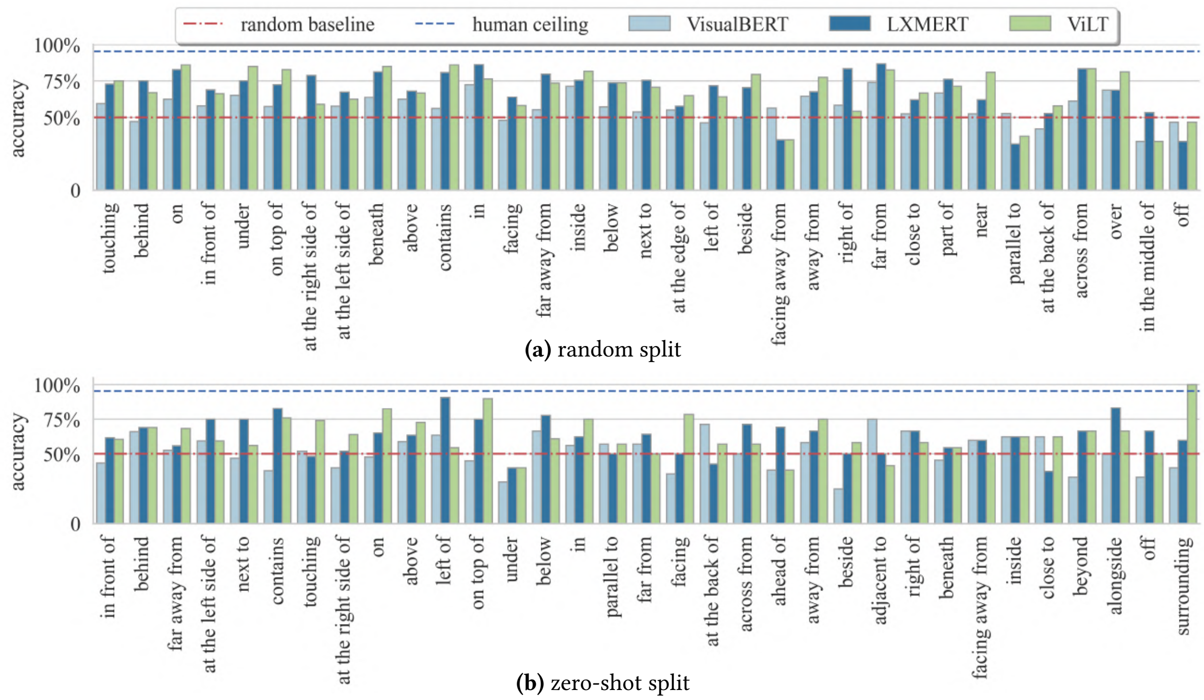
Previous. VSR authors [25] test three popular VLMs: VisualBERT [26], LXMERT [11], and ViLT [27]. All three models are stacked Transformers [75] that take image and text pairs as input. The difference mainly lies in how or whether they encode position information of objects. Checkpoints are saved every 100 iterations and the best checkpoint on the dev set is used for testing. All models are run three times using three random seeds. The only metric used for evaluation is **accuracy**.

We show previous results in Table 4.3, which includes development and test performance of random and zero-shot splits [25]. On **random split**, LXMERT and ViLT are the best models , reaching 70% of accuracy in dev and test. VisualBERT is below 60% , just slightly better than random chance. On **zero-shot split**, performance declines significantly and the best model LXMERT only obtains 63.2% accuracy in test. This means that concept zero-shot learning is fundamentally a hard task for current models. When compared to **human performance**, there is a gap of more than 20% with the best models. Two annotators labelled 500 examples from the test set to calculate human performance.

	random split		zero-shot split	
model↓	dev	test	dev	test
human		95.4		
VisualBERT	$59.2_{\pm 0.9}$	$57.4_{\pm 0.9}$	$57.4_{\pm 2.2}$	$54.0_{\pm 1.3}$
LXMERT	$\mathbf{73.8}_{\pm 1.2}$	$\mathbf{72.5}_{\pm 1.4}$	$\mathbf{69.2}_{\pm 1.0}$	$\mathbf{63.2}_{\pm 1.7}$
ViLT	$71.9_{\pm 1.3}$	$71.0_{\pm 0.7}$	$66.7_{\pm 1.7}$	$62.4_{\pm 1.5}$

Table 4.3: Previous model performance on VSR. Results of both random and zero-shot splits, both validation and tests are listed.

Sensitiveness to random seeds. In general, models have larger standard deviations on the zero-shot split, probably because the zero-shot dev/test sets are smaller. The gap between dev and test sets becomes much greater on zero-shot split likely because of the same reason. Due to the **fluctuations**, authors recommend always reporting the average performance of three runs to make sure the conclusion is reliable [25].



(a) random split

(b) zero-shot split

Figure 4.3: Previous model performance by relation on the random (upper) and zero-shot (lower) split test sets. Relation order sorted by frequency (high to low from left to right). Only relations with more than 15 and 5 occurrences on the random and zero-shot tests respectively are shown.

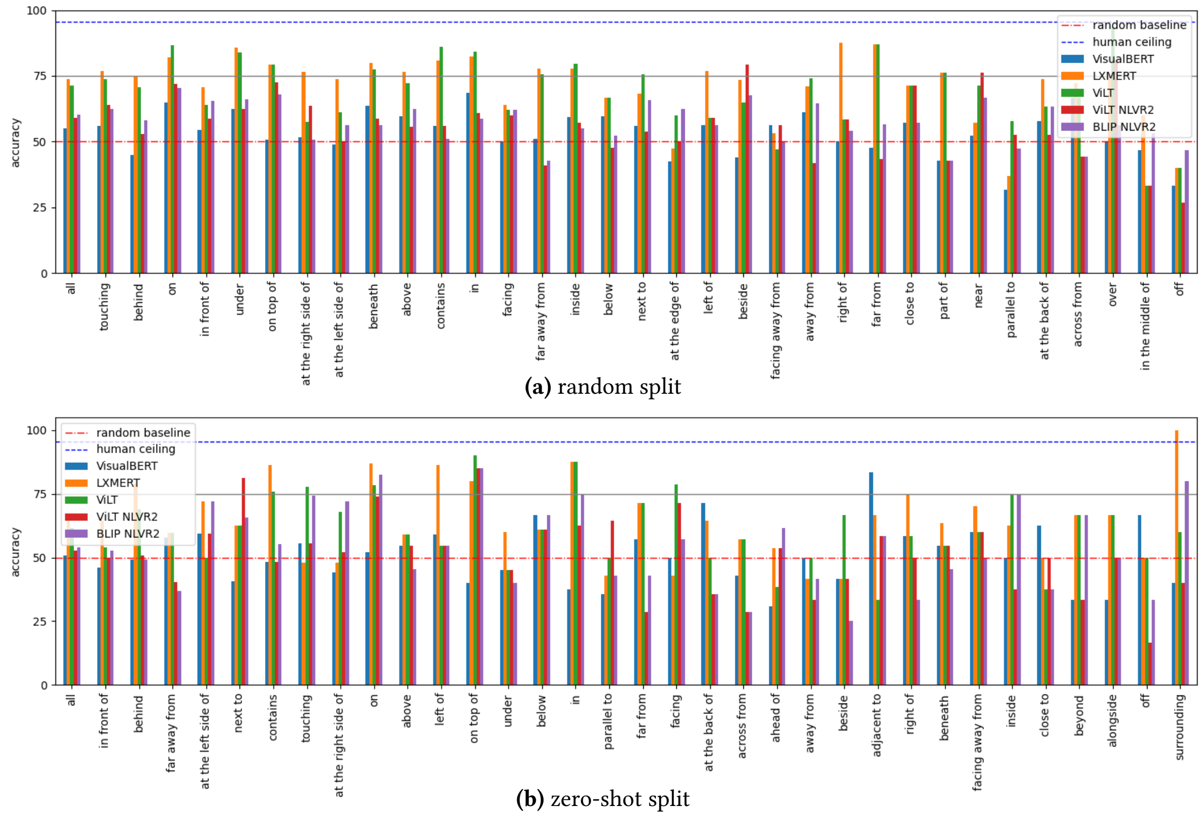
Explicit positional information matters. LXMERT and ViLT outperform VisualBERT by more than 10% on both splits [25]. This is expected because LXMERT and ViLT encode explicit positional information, and VisualBERT does not. LXMERT has position features as part of the input which encodes the relative coordinates of objects. ViLT slices an image into patches and uses positional encoding to signal the patches' relative positions.

Ours. We first test the same previous models. We also evaluate ViLT [27] and BLIP [28] models that have been fine-tuned on NLVR2. We show our results in Table 4.4, which includes development and test performance of random and zero-shot splits. Results are very similar to previous ones for VisualBERT, LXMERT and ViLT, the differences can be attributed to fluctuations. Regarding zero-shot NLVR2 results, performance drops a lot. This is understandable because NLVR2 contains some spatial relations, but are only a small part of the dataset. Moreover, NLVR2 examples contain two images, and VSR examples have only one image. To evaluate on VSR, we need to pass the same image twice, or change the caption to mention one of the images.

	random split		zero-shot split	
model↓	dev	test	dev	test
human		95.4		
VisualBERT	60.1	55.1	56.8	50.8
LXMERT	**73.3**	**73.9**	**70.3**	**65.5**
ViLT	72.7	71.2	66.0	61.6
ViLT NLVR2	57.9	59.1	56.4	52.8
BLIP NLVR2	60.9	60.1	57.9	53.9

Table 4.4: Our model performance on VSR. Results of both random and zero-shot splits, both validation and tests are listed.

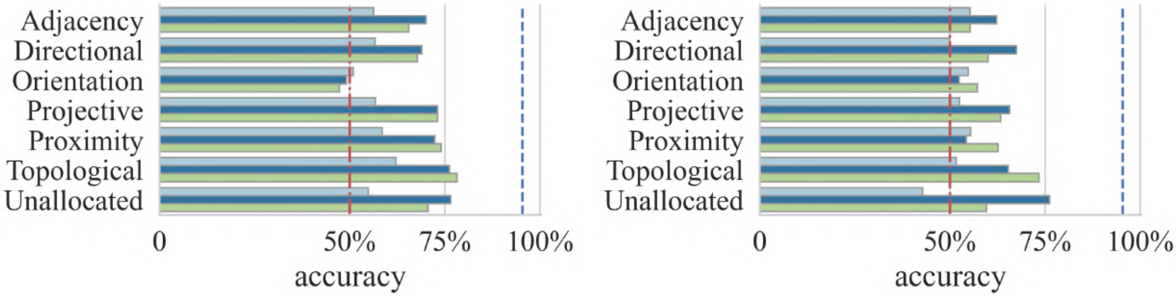
We see that there is still a gap between random and zero-shot splits when testing models that are



(a) random split

(b) zero-shot split

Figure 4.4: Our model performance by relation on the random (upper) and zero-shot (lower) split test sets. Relation order sorted by frequency (high to low from left to right). Only relations with more than 15 and 5 occurrences on the random and zero-shot tests respectively are shown.



(a) random split

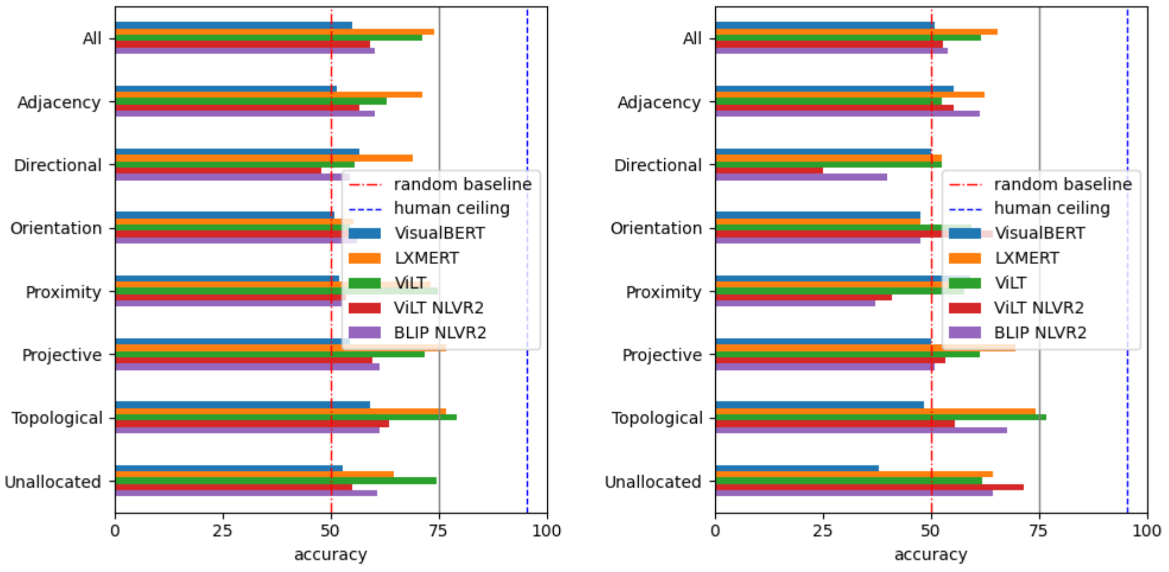
(b) zero-shot split

Figure 4.5: Previous model performance by meta categories of relations, on the random (left) and zero-shot (right) split test sets.

trained on NLVR2. This suggests that the dev and test sets of the zero-shot split are more difficult than random ones. Therefore, the difference in accuracy can not be attributed only to the unseen concepts. The gap between dev and test sets is also maintained in the zero-shot split, which might mean that the test set is inherently more difficult.

4.3.2 Results By Relation

Previous. Figure 4.3 shows performance by relation of previous models on random and zero-shot splits [25]. Only the most common relations are shown and they are sorted from left to right by frequency. It seems that there is no correlation between performance and frequency. This suggests that some relations are harder than others, regardless of the training examples.



(a) random split

(b) zero-shot split

Figure 4.6: Our model performance by meta categories of relations, on the random (left) and zero-shot (right) split test sets.

Any relation that requires recognising **orientations** or **facing directions** of objects is very hard, e.g., "facing", "facing away from", "parallel to" and "at the back of". For example, LXMERT failed on the examples in Figures 4.2c and 4.2e, which require understanding the front of a hair drier and a person respectively.

Left and right relations such as "at the left/right side of" and "left/right of" are also difficult because they can refer to either viewer's or object's **reference frames**. For instance, in Figure 4.2a, all three models predicted `False` since the potted plant is at the left of the bench if the viewer is the reference frame. However, if using the bench as the reference frame, the potted plant is at the right.

While generally speaking orientation has been hard, the tested VLMs do well on some seemingly hard cases. As an example, all models correctly predicted Figure 4.2f, a case that requires compositional zero-shot generalisation capability. Models need to generalise the concept of "face" to a fire hydrant by identifying eyes.

Ours. We show performance by relation of our models on random and zero-shot splits in Figure 4.4. Only the most common relations are shown and they are sorted from left to right by frequency. Results for VisualBERT, LXMERT and ViLT are similar to the previous experiments. LXMERT and ViLT are clearly better than VisualBERT in both random and zero-shot splits, with very few exceptions. ViLT and BLIP models that are fine-tuned on NLVR2 are generally better than VisualBERT but worse than the other models. There are some relations where they get similar and a few relations where they even get better results. This could be because these relations might be more common in NLVR2. We show VSR result tables of each split by relation in Appendix B.1.

4.3.3 Results By Meta Category

Previous. Previous results by relation meta category are shown in Figure 4.5. Relations are grouped into categories to get a more high-level understanding of the relations' performance. Table 4.1 shows relations included in each meta category: "Adjacency", "Directional", "Orientation", "Projective", "Proximity", "Topological" and "Unallocated". **"Orientation"** is the worst category on both splits, and on average all performances are at chance level [25].

Performance decreases when comparing **random** and **zero-shot** splits for almost all categories and models. “Proximity” is the category that decreases the most, from close to 75% accuracy in random split to chance level in the zero-shot split. “Proximity” contains relations like “close to”, “near” and “far from”. Authors think it is due to proximity being relative and very dependent on the concept and its context [25]. As zero-shot split concepts are different in each set, models have more difficulties.

Ours. Figure 4.6 shows our results by relation meta category. Results for VisualBERT, LXMERT and ViLT are similar to the previous experiments. LXMERT and ViLT are superior to the other models for both splits in most categories. NLVR2 models are generally better than VisualBERT in the random split. In the zero-shot split there are bigger differences between the models. NLVR2 models are very bad in the ”Directional” and ”Proximity” categories and better than other models in the ”Unallocated” category. We show VSR result tables of each split by relation meta category in Appendix B.2.

5 Synthetic Dataset Creation

As we have seen in Chapter 3 models perform badly in Winoground. The problem is that only zero-shot can be done because Winoground has no training data. Annotating examples for Winoground is very time-consuming and therefore creating a big dataset for training is very difficult. Therefore, compositional reasoning has to be learned either in pretraining or when fine-tuning with other datasets. However, there is no training dataset that focuses on compositional reasoning. A solution could be to create a **synthetic dataset** for compositional reasoning and fine-tune the models with it.

We explore three different options for synthetic dataset generation: Text-to-Image Generation (5.1), Image Captioning (5.2) and Image Retrieval (5.3). These experiments also allow us to gain more insight into the dataset and the tested models.

5.1 Text-to-Image Generation

As noted, large generative **text-to-image diffusion models**, like like Stable Diffusion [17] are able to generate stunning images. They are known to possess some visual-reasoning skills [18]. However, a recent work [19] has shown that they **struggle to understand the composition of some concepts**, such as confusing the attributes and relations of different objects. Here we want to know if they are good enough for synthetic dataset generation for compositional reasoning. First, we generate some images from Winoground captions automatically (Section 5.1.1). Then, we also do a manual qualitative evaluation of the generated images (Section 5.1.2).

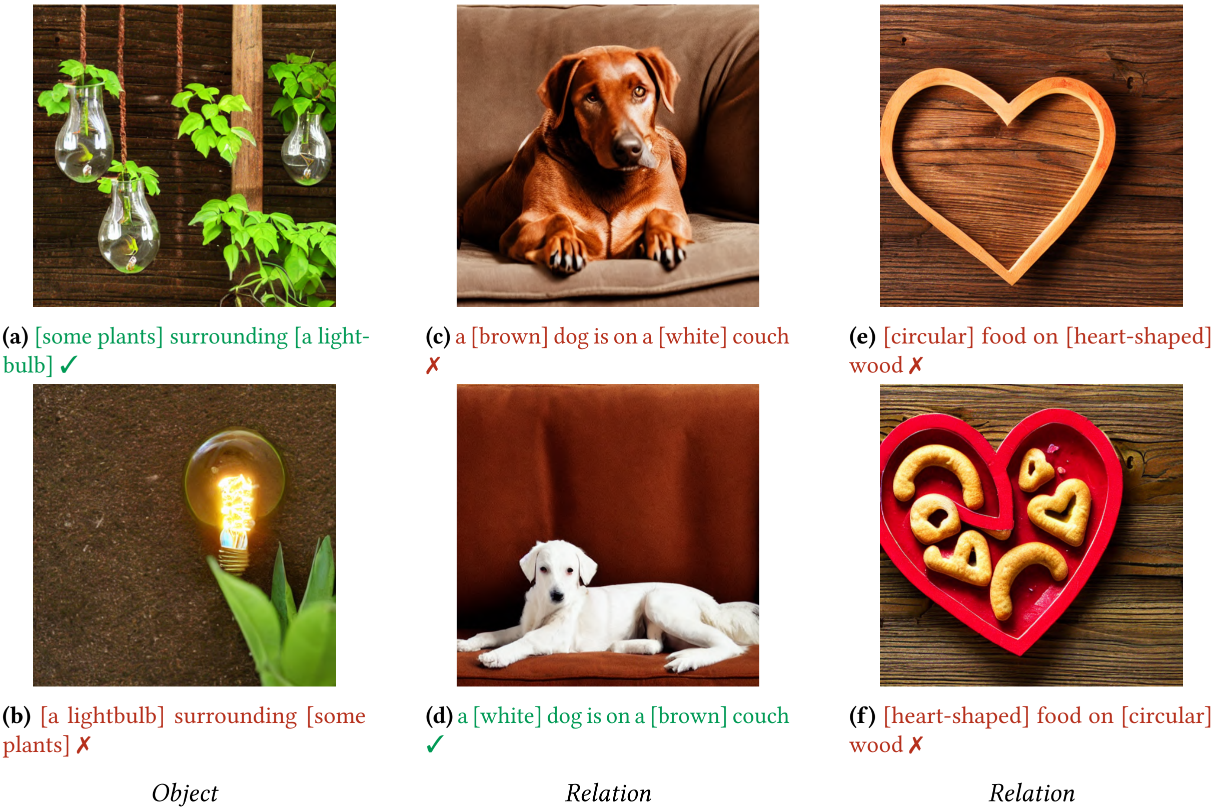
5.1.1 Automatic Generation

With the aim of evaluating the compositional ability of diffusion models, we used the state-of-the-art Stable Diffusion model [17] to generate images from Winoground captions. We generate images 9 images for each Winoground caption, which results in a total of $800 * 9 = 7200$ images. Here we will have a look at a few examples of generated images to compare them with the original images. We select the same caption pairs that were used to present the Winoground dataset in the previous section. Caption pairs with the same words are shown in columns. Only the first generated image is shown in the examples, which is not necessarily the best one.

In Figure 5.1, there are two correct and four incorrect images. First, the top image in the left column is correct, but the bottom one is wrong. Then, the colour of the dog is correct in the middle column images, but the colour of the couch is mistaken in the top image. Finally, the two images in the right column are wrong, food is missing in the first one and the shape is wrong in the second one.

In Figure 5.2, there are four correct and two incorrect images. First, both images of the left column correspond to the first caption, so the bottom image is wrong. Next, both images in the middle column correspond to the second caption, so the top one is wrong. Finally, the third pair is the only one that is completely correct.

In Figure 5.3, all of the six images are wrong and do not correspond to any of the captions. First, both examples in the left column are wrong, there is no magnifying glass in any of them. Next, in the middle column only the person with the ponytail is shown and other things are missing. Finally, both images in the right column only show three windows, there is no people in neither of them.



(a) [some plants] surrounding [a lightbulb] ✓

(b) [a lightbulb] surrounding [some plants] ✗

(c) a [brown] dog is on a [white] couch ✗

(d) a [white] dog is on a [brown] couch ✓

(e) [circular] food on [heart-shaped] wood ✗

(f) [heart-shaped] food on [circular] wood ✗

Object *Relation* *Relation*

Figure 5.1: Stable Diffusion examples for the swap-dependent linguistic tags *Object*, *Relation* and *Relation* from left to right. They are additionally tagged with 1 main predicate. Correct examples are marked in green ✓ and incorrect ones in red ✗.

5.1.2 Manual Evaluation

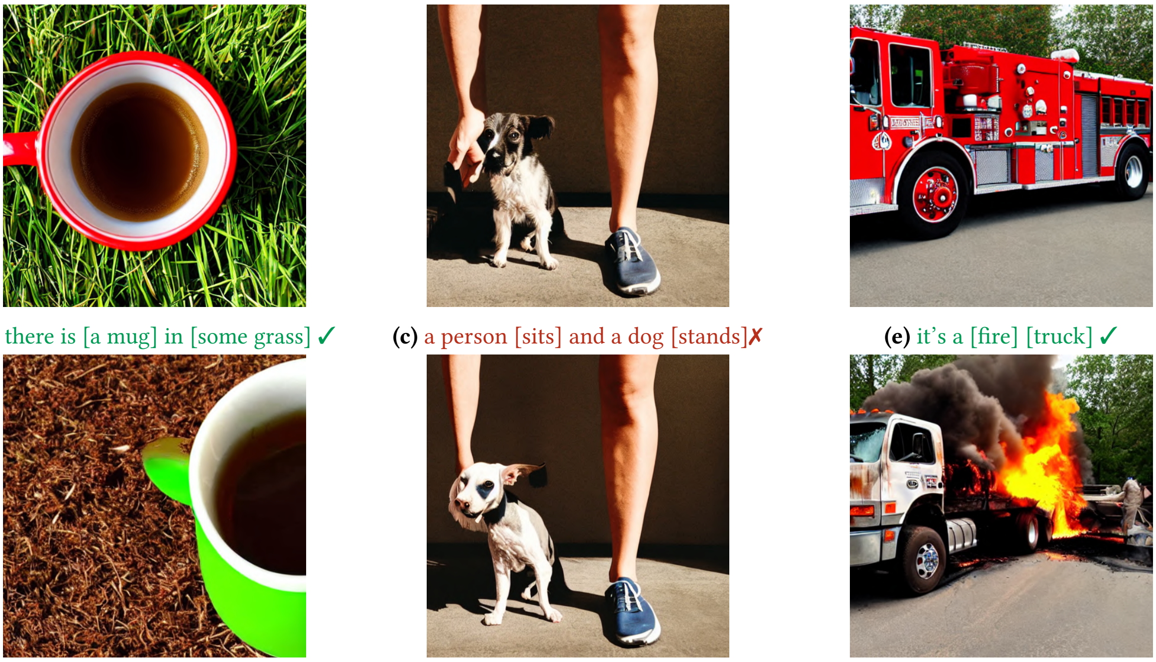
In this section we will explain the manual annotation that was performed to evaluate generated images. We used Label Studio [76] to annotate images generated by Stable Diffusion. As annotating all the images would take a very long time, we choose to annotate only one image per caption.

In each annotation there are two captions from Winoground and two images generated with Stable Diffusion. Each image is created from one caption but the order of the images is random. Each image is annotated separately, we included the two images together to speed up annotation. The annotators have to choose which text corresponds to each image: the first caption, the second caption, both or none. An screenshot of the annotation interface can be seen in Figure 5.4.

There were 6 annotators in total and each one annotated 50 examples, for a total of 300 annotated examples. Each example includes two images, for a total of 600 annotated images. There are 400 examples in total, so we considered that 300 is a large enough subset to be able to draw significant conclusions from.

The statistics of the annotation task are shown in Table 5.1. The general conclusion is that Stable Diffusion is not good at this task. Most of the images do not match any of the captions, 351 out of 600. There are only 25 images that match both captions. The remaining images match one caption or the other (224), but there are many that match the incorrect caption (94). If we take into account image pairs, there are only a few correct ones, 23 out of 300.

Therefore, using a diffusion model to create a synthetic dataset might not be robust enough. It would require generating many images to get correct ones, and manual filtering to discard the wrong images. In the future, we could test better diffusion models that make less mistakes. In Section 5.3, we will test another approach to obtain similar images. Instead of generating new images, they are retrieved from a



(a) there is [a mug] in [some grass] ✓

(b) there is [some grass] in [a mug] ✗

Object

(c) a person [sits] and a dog [stands]✗

(d) a person [stands] and a dog [sits] ✓

Relation

(e) it's a [fire] [truck] ✓

(f) it's a [truck] [fire] ✓

Both

Figure 5.2: Stable Diffusion examples for the swap-dependent linguistic tags *Object*, *Relation* and *Both* from left to right. They are additionally tagged with 1, 2 and 1 main predicates from left to right. Correct examples are marked in green ✓ and incorrect ones in red ✗.

	Caption 0	Caption 1	Both	None	All
Caption 0	65	48	12	175	300
Caption 1	46	65	13	176	300
All	111	113	25	351	600

Table 5.1: Statistics of the annotations. Rows shows the caption used for generation and columns show the annotation choice.

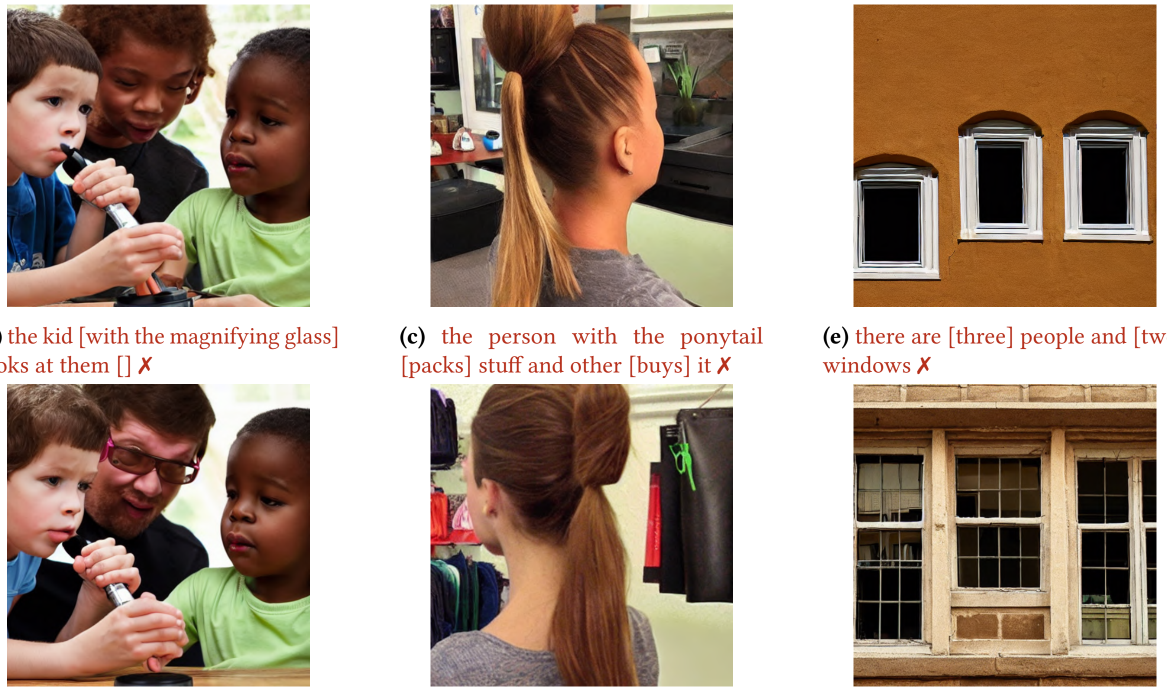
huge dataset.

5.2 Image Captioning

In this section, we explain how we evaluate automatic image captioning for synthetic dataset creation. Captioning could be applied for generating automatic descriptions of images much faster than humans. This would reduce the work needed to create training datasets targeted for compositional or spatial reasoning.

In this work, we used OFA [29] and BLIP [28] models of different sizes to generate captions for all Winoground images. We chose these models because they are SOTA in image captioning and we also use them in other evaluations.

To know how good generated captions are, we can compare them with the real captions using automatic metrics. To evaluate the quality of the generated captions, we calculated the BLEU score [77] between the generated caption and the caption Winoground, shown in Table 5.2. Scores are very low for all models, considering that BLEU scores range from 0 to 100. This indicates that the captions generated by these models are very different from the real captions. We find BLIP gets better scores than OFA and bigger models are generally better.



(a) the kid [with the magnifying glass] looks at them [] ✗

(c) the person with the ponytail [packs] stuff and other [buys] it ✗

(e) there are [three] people and [two] windows ✗

(b) the kid [] looks at them [with the magnifying glass] ✗

(d) the person with the ponytail [buys] stuff and other [packs] it ✗

(f) there are [two] people and [three] windows ✗

Pragmatics *Series* *Symbolic*

Figure 5.3: Stable Diffusion examples for the visual tags *Pragmatics*, *Series* and *Symbolic* from left to right. They are additionally tagged with the *Relation* tag, and 1, 2, and 1 main predicate from left to right. Correct examples are marked in green ✓ and incorrect ones in red ✗.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
OFA$_{Tiny}$	14.40	5.76	2.50	1.30
OFA$_{Base}$	16.68	7.12	3.26	1.58
OFA$_{Medium}$	16.28	6.47	2.84	1.39
OFA$_{Large}$	15.10	6.45	3.03	1.53
OFA$_{Huge}$	15.73	6.94	3.06	1.35
BLIP (ViT-B/16)	17.80	8.10	3.96	2.01
BLIP (ViT-L/16)	17.96	8.31	4.36	2.50

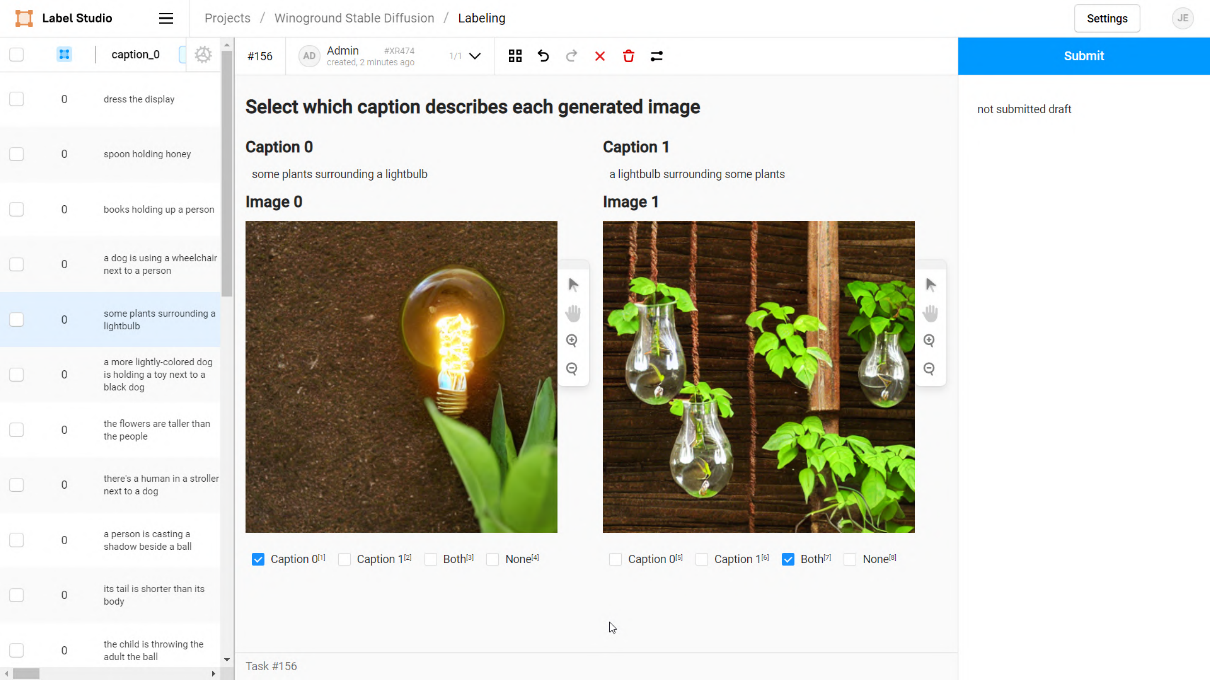
Table 5.2: Image captioning BLEU scores of OFA and BLIP models.

One reason for this could be that the real Winoground captions are not typical captions. They are hand-crafted so that they contain the same words in a different order. Another reason could be that these models are not good at describing these types of images that require compositional reasoning. We have previously seen that these models are bad at matching Winoground images with captions.

Analysing the captions manually would be necessary to know how good they really are. We show some caption examples of the best performing model, BLIP large, so that we can compare them with the original ones. The complete caption files for all the models can be found in the GitHub repository.

In Figure 5.5, all the captions are correct. They describe the images correctly. Some descriptions are more detailed that in the original captions, but some attributes are also missing. For example, the color of the couch and the shape of the cutting wood are not mentioned.

Every caption in Figure 5.6 is also correct. However, some important details that are present in the original captions are missing. In the second pair of images, the first caption does not mention the



□	0	dress the display
□	0	spoon holding honey
□	0	books holding up a person
□	0	a dog is using a wheelchair next to a person
□	0	some plants surrounding a lightbulb
□	0	a more lightly-colored dog is holding a toy next to a black dog
□	0	the flowers are taller than the people
□	0	there's a human in a stroller next to a dog
□	0	a person is casting a shadow beside a ball
□	0	its tail is shorter than its body
□	0	the child is throwing the adult the ball

Select which caption describes each generated image

Caption 0

some plants surrounding a lightbulb

Caption 1

a lightbulb surrounding some plants

Image 0

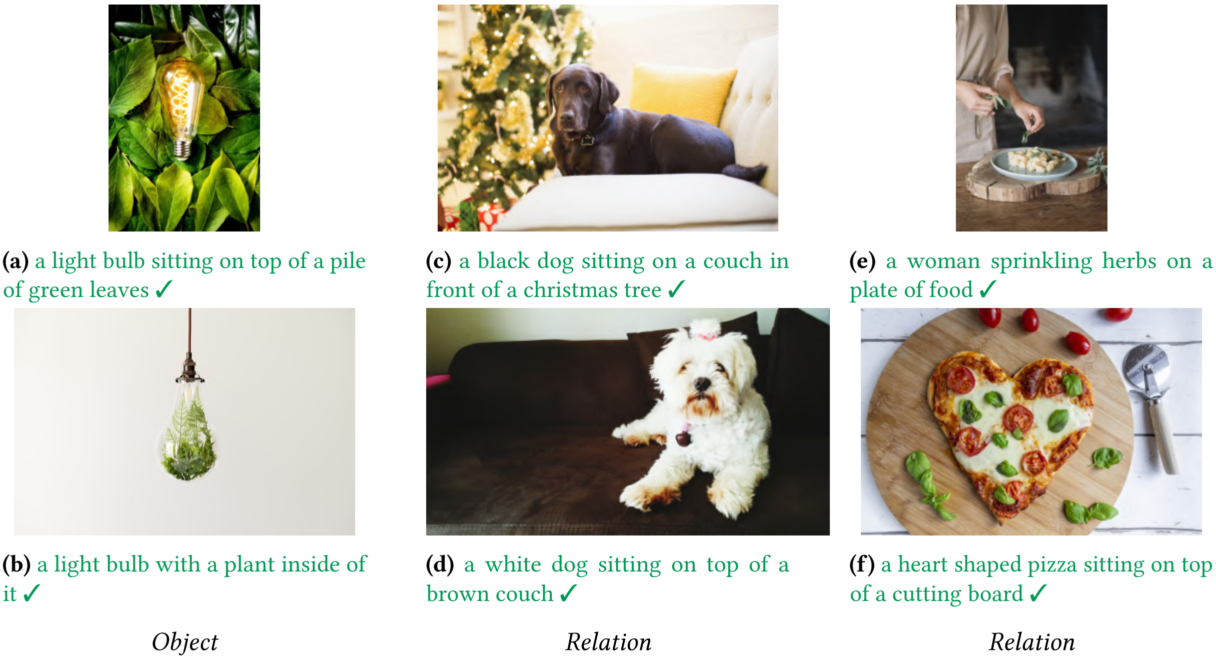
☑ Caption 0[1] □ Caption 1[2] □ Both[3] □ None[4]

Image 1

□ Caption 0[5] □ Caption 1[6] ☑ Both[7] □ None[8]

not submitted draft

Figure 5.4: Label Studio annotation interface



(a) a light bulb sitting on top of a pile of green leaves ✓

(b) a light bulb with a plant inside of it ✓

(c) a black dog sitting on a couch in front of a christmas tree ✓

(d) a white dog sitting on top of a brown couch ✓

(e) a woman sprinkling herbs on a plate of food ✓

(f) a heart shaped pizza sitting on top of a cutting board ✓

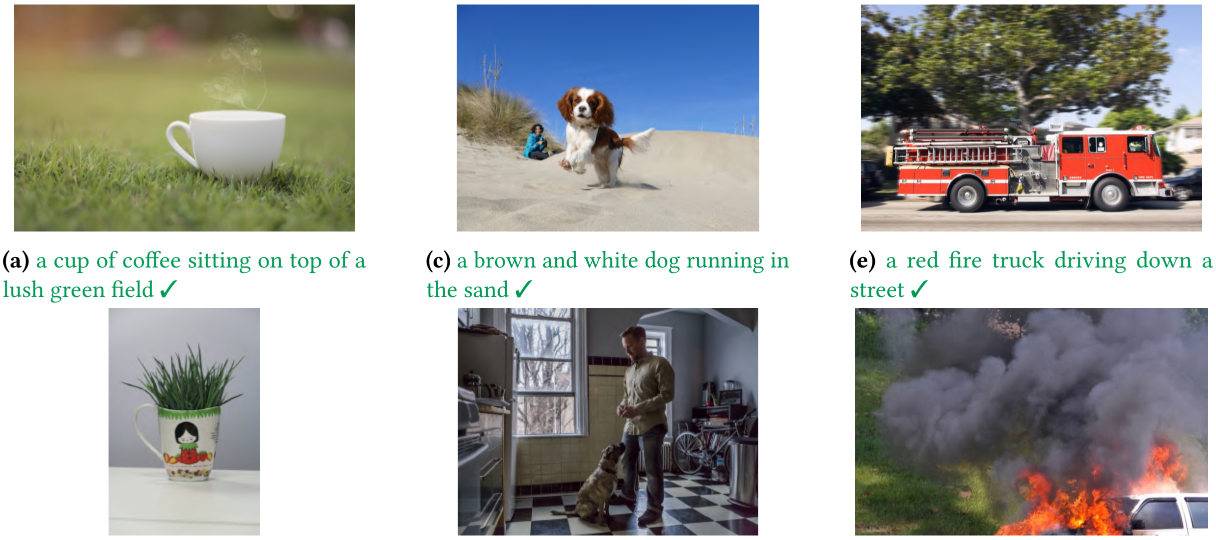
Object *Relation* *Relation*

Figure 5.5: Image Captioning examples from the Winoground dataset for the swap-dependent linguistic tags *Object*, *Relation* and *Relation* from left to right. They are additionally tagged with 1 main predicate. Correct examples are marked in green ✓ and incorrect ones in red ✗.

person, and the second one does not specify that the dog is sitting.

The first pair of captions in Figure 5.7 is wrong. In the first image, the young boy is the one holding the magnifying glass. The second one is completely wrong, they are not sitting at a table and the magnifying glass is not mentioned. Other examples are correct, but more generic than the original captions. The describe the images, without mentioning detains such as the number of people and windows.

The general conclusion is that most captions are quite good. They are very different from the original



(a) a cup of coffee sitting on top of a lush green field ✓

(c) a brown and white dog running in the sand ✓

(e) a red fire truck driving down a street ✓

(b) a cup with a plant in it sitting on a table ✓

(d) a man standing next to a dog in a kitchen ✓

(f) a car is on fire in a field ✓

Object *Relation* *Both*

Figure 5.6: Image Captioning examples from the Winoground dataset for the swap-dependent linguistic tags *Object*, *Relation* and *Both* from left to right. They are additionally tagged with 1, 2 and 1 main predicates from left to right. Correct examples are marked in green ✓ and incorrect ones in red ✗.

ones, but they describe the images correctly. They provide extra information about the images to the models, that is not included in the original captions. They could be used to improve the results of the models by incorporating them into the evaluation process. For example, we could compare the original and generated captions and pair them by similarity.

It seems that these models are not that bad at describing images. However, we have seen that they have more difficulties when matching Winoground images with captions, especially when captions are very similar. This suggests that text encoding might be their biggest limitation, and could be the main reason for their low performance on Winoground.

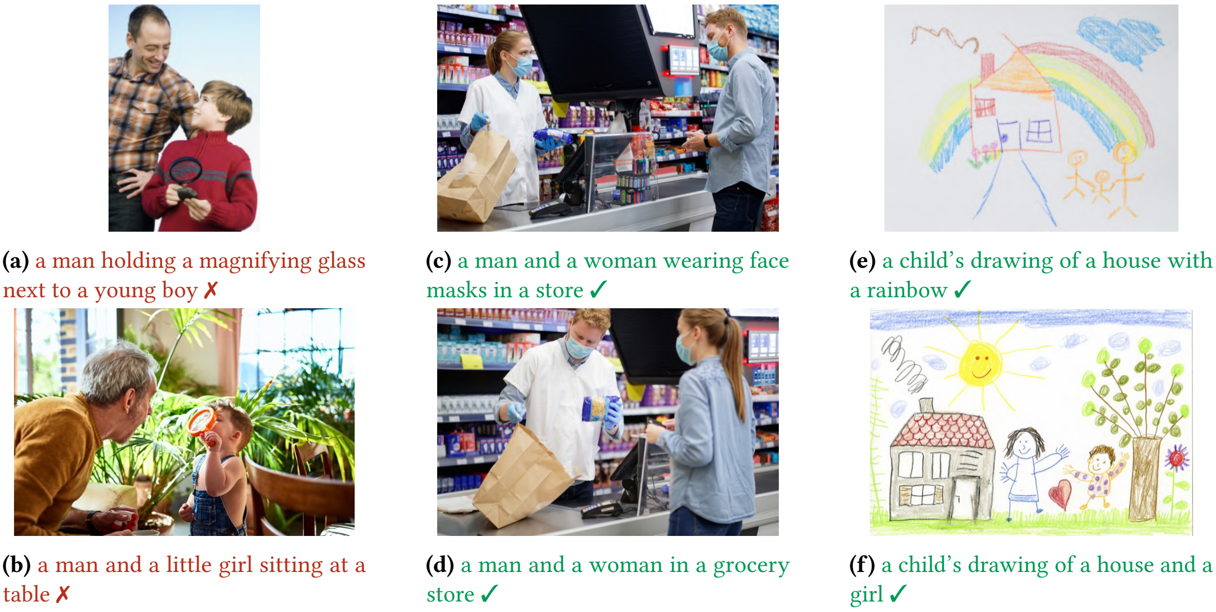
5.3 Image Retrieval

In this section, we explain how we could use image retrieval for synthetic dataset generation. First, we could retrieve images of interest from a large image dataset. For example, we could retrieve many similar images from captions to increase the size of a dataset. We could also change the captions to retrieve images with different objects. Then, captioning could be applied to generate automatic descriptions of images. We would have a large dataset of images with real and generated captions. And this would require a low effort compared to human annotation.

We used CLIP retrieval[1] to retrieve images from LAION-5B [58] dataset. We used Winoground captions and images to get similar images. For each caption and image, we compute its embeddings using CLIP ViT-L-14. Then the system uses a KNN algorithm to retrieve images that have similar embeddings. We can also compute the mean of caption and image embeddings to retrieve images that match both the image and the caption.

We used the python CLIP Client with the default parameters, which retrieves a maximum of 40 images with each query. It also retrieves the original caption of the image and a similarity score. The system also has an aesthetic score that can be used to retrieve better looking images. It also removes

[1] https://github.com/rom1504/clip-retrieval



(a) a man holding a magnifying glass next to a young boy ✗

(b) a man and a little girl sitting at a table ✗

Pragmatics

(c) a man and a woman wearing face masks in a store ✓

(d) a man and a woman in a grocery store ✓

Series

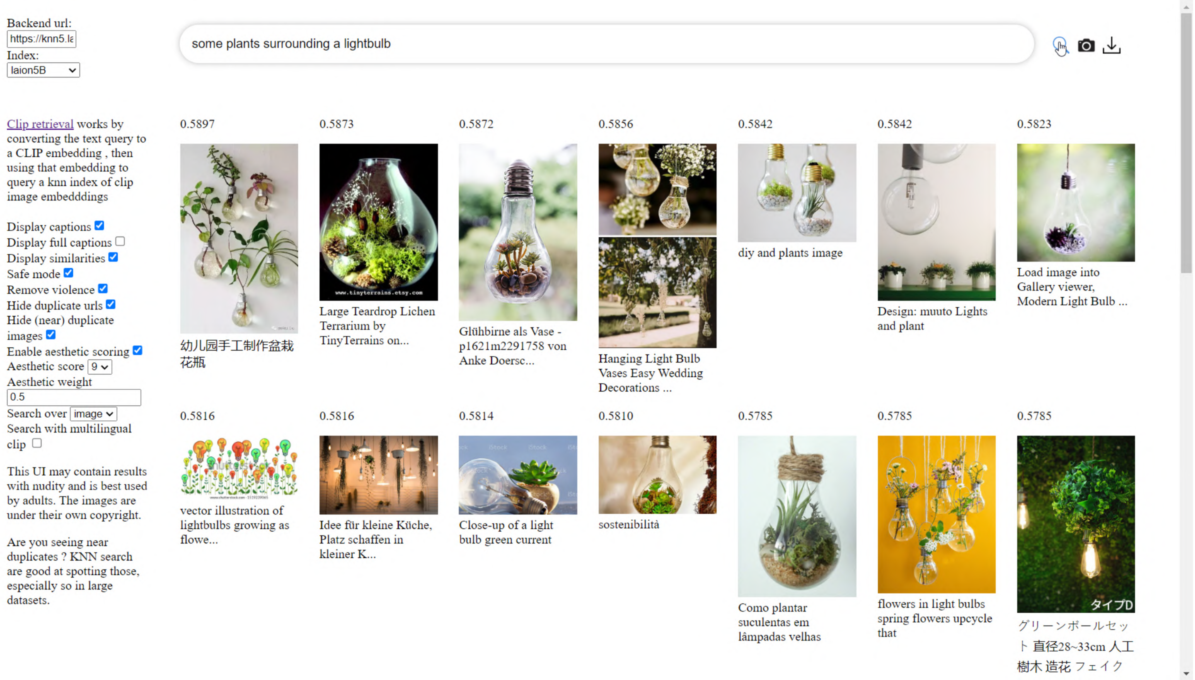
(e) a child's drawing of a house with a rainbow ✓

(f) a child's drawing of a house and a girl ✓

Symbolic

Figure 5.7: Image Captioning examples from the Winoground dataset for the visual tags *Pragmatics*, *Series* and *Symbolic* from left to right. They are additionally tagged with the *Relation* tag, and 1, 2, and 1 main predicate from left to right. Correct examples are marked in green ✓ and incorrect ones in red ✗.

duplicate images and images that contain unsafe content and violence. Figure 5.8 shows an example search with the alternative retrieval interface[2].



Backend url: https://knn5.la

Index: laion5B

some plants surrounding a lightbulb

Clip retrieval works by converting the text query to a CLIP embedding , then using that embedding to query a knn index of clip image embedddings

Display captions ☑
Display full captions ☐
Display similarities ☑
Safe mode ☑
Remove violence ☑
Hide duplicate urls ☑
Hide (near) duplicate images ☑
Enable aesthetic scoring ☑
Aesthetic score 9
Aesthetic weight 0.5
Search over image
Search with multilingual clip ☐

This UI may contain results with nudity and is best used by adults. The images are under their own copyright.

Are you seeing near duplicates ? KNN search are good at spotting those, especially so in large datasets.

0.5897 幼儿园手工制作盆栽花瓶

0.5873 Large Teardrop Lichen Terrarium by TinyTerrains on...

0.5872 Glühbirne als Vase - p1621m2291758 von Anke Doersc...

0.5856 Hanging Light Bulb Vases Easy Wedding Decorations ...

0.5842 diy and plants image

0.5842 Design: muuto Lights and plant

0.5823 Load image into Gallery viewer, Modern Light Bulb ...

0.5816 vector illustration of lightbulbs growing as flowe...

0.5816 Idee für kleine Küche, Platz schaffen in kleiner K...

0.5814 Close-up of a light bulb green current

0.5810 sostenibilità

0.5785 Como plantar suculentas em lâmpadas velhas

0.5785 flowers in light bulbs spring flowers upcycle that

0.5785 グリーンボールセット 直径28~33cm 人工樹木 造花 フェイク

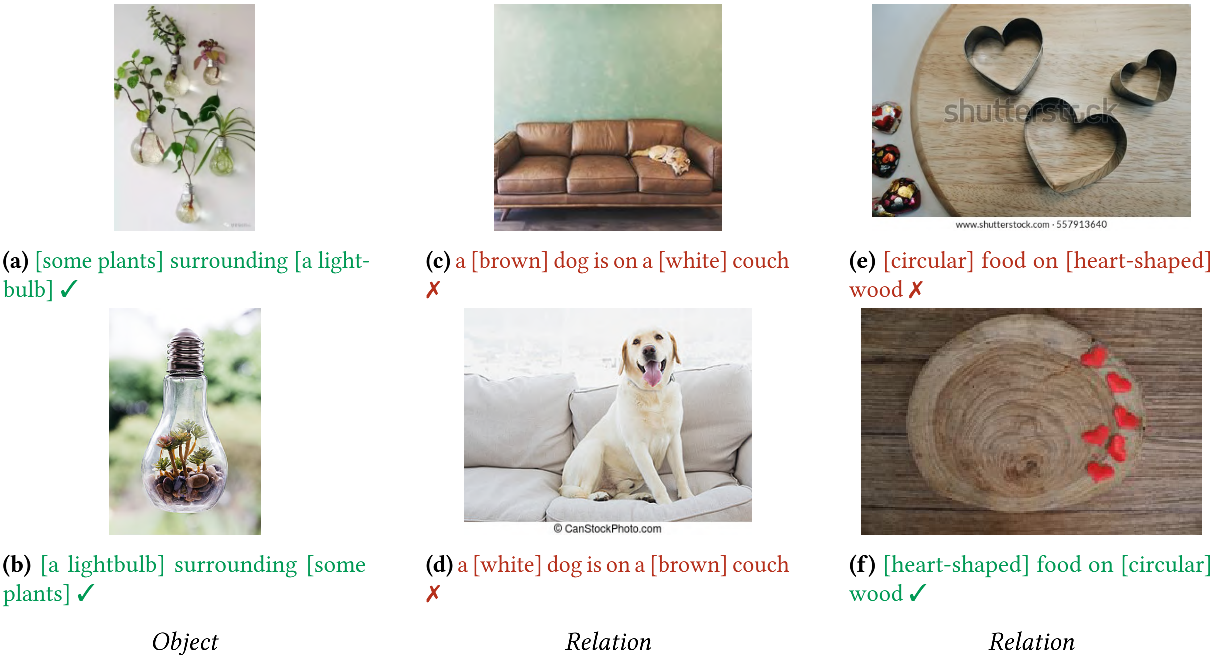
Figure 5.8: CLIP Retrieval interface search example. Many of the images are wrong and correspond to the other caption.

We have selected a few examples of retrieved images to compare them with the original images. These images were retrieved using only the captions. Only the first generated image is shown in the

[2] https://rom1504.github.io/clip-retrieval

examples, which is not necessarily the best one. The complete retrieval file can be found in the GitHub repository.

In Figure 5.9, the first pair of images is correct, both images match the captions. In the second pair, the color of the dog is correct, but the couch has a wrong color. In the third example, the first image has wrong shapes and the second image is wrong.



(a) [some plants] surrounding [a lightbulb] ✓

(b) [a lightbulb] surrounding [some plants] ✓

(c) a [brown] dog is on a [white] couch ✗

(d) a [white] dog is on a [brown] couch ✗

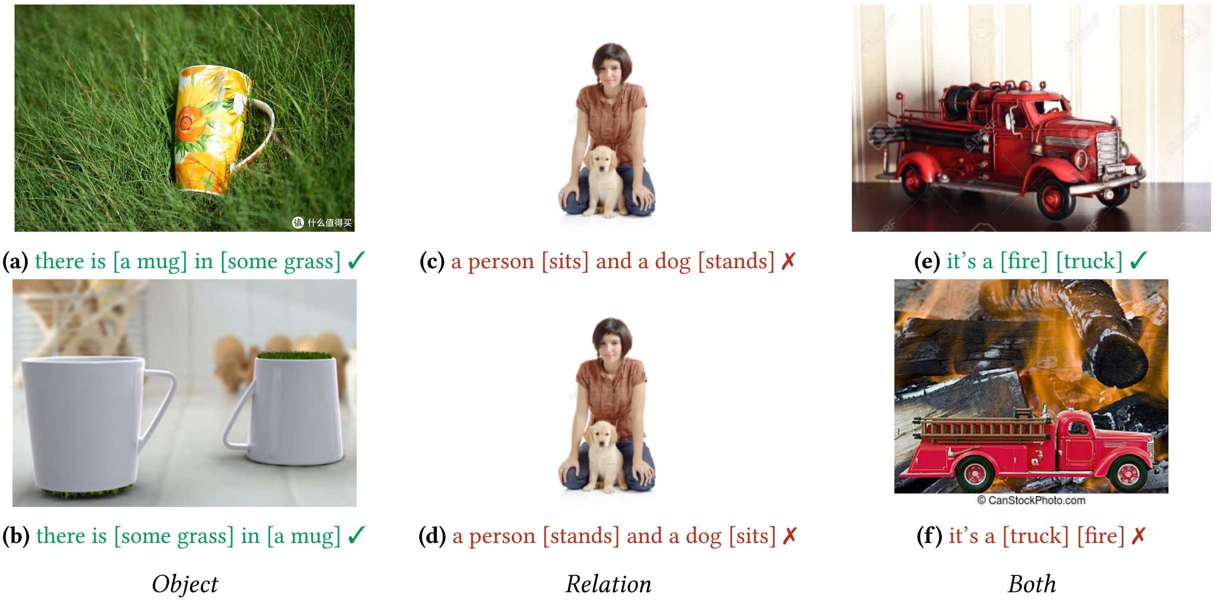
(e) [circular] food on [heart-shaped] wood ✗

(f) [heart-shaped] food on [circular] wood ✓

Object *Relation* *Relation*

Figure 5.9: CLIP Retrieval examples for the swap-dependent linguistic tags *Object*, *Relation* and *Relation* from left to right. They are additionally tagged with 1 main predicate. Correct examples are marked in green ✓ and incorrect ones in red ✗.

In Figure 5.10, the first pair is correct. The second pair is wrong, the same image is retrieved for both captions. The first image in the third example is correct, but the second one is incorrect.



(a) there is [a mug] in [some grass] ✓

(b) there is [some grass] in [a mug] ✓

(c) a person [sits] and a dog [stands] ✗

(d) a person [stands] and a dog [sits] ✗

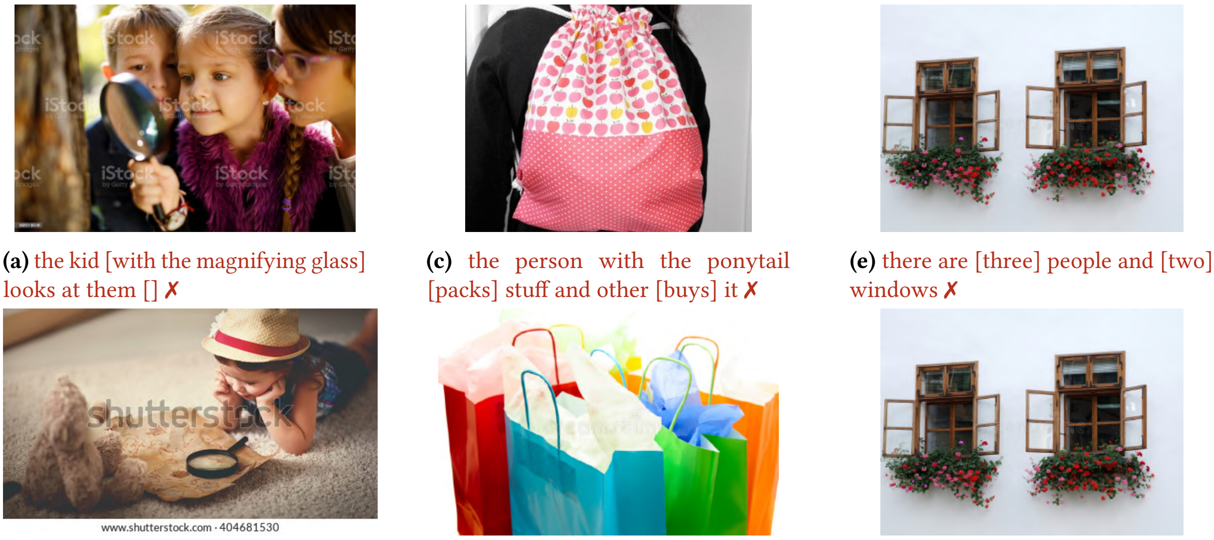
(e) it's a [fire] [truck] ✓

(f) it's a [truck] [fire] ✗

Object *Relation* *Both*

Figure 5.10: CLIP Retrieval examples for the swap-dependent linguistic tags *Object*, *Relation* and *Both* from left to right. They are additionally tagged with 1, 2 and 1 main predicates from left to right. Correct examples are marked in green ✓ and incorrect ones in red ✗.

In Figure 5.11, the first example is wrong, the objects are present but the composition is not correct. In the second pair, some objects are missing in both images. The third pair is wrong, the same image is retrieved, which only contains two windows and no people.



(a) the kid [with the magnifying glass] looks at them [] ✗

(c) the person with the ponytail [packs] stuff and other [buys] it ✗

(e) there are [three] people and [two] windows ✗

(b) the kid [] looks at them [with the magnifying glass] ✗

(d) the person with the ponytail [buys] stuff and other [packs] it ✗

(f) there are [two] people and [three] windows ✗

Pragmatics *Series* *Symbolic*

Figure 5.11: CLIP Retrieval examples for the visual tags *Pragmatics*, *Series* and *Symbolic* from left to right. They are additionally tagged with the *Relation* tag, and 1, 2, and 1 main predicate from left to right. Correct examples are marked in green ✓ and incorrect ones in red ✗.

Similar to image generation, this would also require some filtering because there are many wrong images. As the system retrieves many images, we could select the best ones. The number of retrieved images and the similarity score could also be used as a measure of how common an image is. If there are very few similar images in the dataset, that means that the caption or image is uncommon. This information could be used in Winoground to pair the most common image with the most common caption. However, the system might not be robust enough for this estimation.

6 Conclusions

In this chapter, we provide an overview of the main contributions and conclusions of this work. We managed to accomplish all the objectives of this work. First, we improved the state-of-the-art in compositional reasoning. Next, we performed some zero-shot experiments on spatial reasoning. Finally, we explored three alternatives for synthetic dataset creation: text-to-image generation, image captioning and image retrieval.

Improve the state-of-the-art in compositional reasoning. The original Winoground paper included zero-shot experiments with many pre-trained SOTA systems, and they concluded that, surprisingly, none of them does much better than chance [9]. In this work, we extended the previous experiments with new models that obtained better results than those reported in the original paper. In previous experiments, only pre-trained models are tested. We extend this by testing some models that are fine-tuned for specific tasks such as image-text retrieval and visual reasoning. We compare pre-trained versions with fine-tuned versions of the same models and find out that fine-tuning helps a lot. Our best scores are much better than previous models, 10% in text score, 4% in image score and 7% in group score. However, even the best model is still **far from human performance** in text, image and group scores. There is still a 40% gap in text scores, and 64% in image and group scores.

Perform zero-shot experiments in spatial reasoning. VSR authors [25] train and test three popular VLMs: VisualBERT [26], LXMERT [11], and ViLT [27]. They conclude that there is still a large gap between model and human performance. We extend these experiments and evaluate ViLT [27] and BLIP [28] models fine-tuned on NLVR2. We show that performance drops a lot when evaluating ViLT as zero-shot. This is understandable because spatial relations are only a small part of NLVR2. Moreover, NLVR2 examples contain two images, and VSR examples have only one image. To evaluate on VSR, we need to pass the same image twice, or change the caption to mention one of the images. If we want to improve VSR results in the future, we will need to fine-tune better VLMs such as BLIP directly on VSR.

Investigate text-to-image generative models for synthetic dataset creation. With the aim of evaluating the compositional ability of diffusion models, we used the state-of-the-art Stable Diffusion model [17] to generate images for each Winoground caption. We did a manual evaluation of the generated images to measure their quality and decide if they can be used to create synthetic datasets. The general conclusion is that Stable Diffusion is not good at this task. Most of the generated images do not match the captions. Therefore, using a diffusion model for data augmentation might not be robust enough. It would require generating many images to get the correct ones, and manual filtering to discard the wrong images.

Investigate image captioning for synthetic dataset creation. We used OFA [29] and BLIP [28] models of different sizes to generate captions for all Winoground images. A manual evaluation on a small sample showed that the quality of the generated captions were high. They are very different from the original ones, but they describe the images correctly. They provide extra information about the images to the models, that is not included in the original captions. Captioning could be applied for generating automatic descriptions of images much faster than humans. This would reduce the work needed to create training datasets targeted for compositional or spatial reasoning.

Investigate image retrieval for synthetic dataset creation. We used CLIP retrieval to retrieve images from LAION-5B [58] dataset. We used Winoground captions and images to get similar images. This system could be used to increase the size of our dataset. We could retrieve many similar images for our captions. We could also change the captions to retrieve images with different objects. This would also require some filtering because there are many wrong images.

7 Future Work

This chapter provides an overview of future work areas for further research. We propose four ideas for synthetic dataset generation (Section 7.1): explicit verbalization (7.1.1), text-to-image (7.1.2), image-to-image (7.1.3) and image captioning and retrieval (7.1.4). We also include some ideas for extending current datasets to be multilingual (Section 7.2).

7.1 Synthetic Dataset Creation

As to avoid the scarcity of multimodal datasets that explicitly describe spatial relations, we analyze further options to construct synthetic datasets. We could then use them to train existing language models in a self-supervised way, with the final aim of obtaining spatially grounded language models. We propose four options that could be combined to produce the synthetic datasets: explicit verbalization (7.1.1), text-to-image (7.1.2), image-to-image (7.1.3) and image captioning and retrieval (7.1.4).

In the future, we might investigate the use of **multi-tasking** and **multi-sourcing** to improve generalization properties. In a multi-task training paradigm, the model is forced to learn more than one task simultaneously, therefore improving its generalization capabilities. We could investigate multi-task settings to combine the verbalized dataset, the images and captions produced by the generative VLMs, as well as traditional training data to obtain spatial-aware language models.

We could also improve **zero-shot** and **few-shot** generalization of VLM models to obtain effective models in small data regimes of the spatial reasoning domain. Thus eliminating the necessity of explicitly annotating big quantities of data.

7.1.1 Explicit Verbalization

Explicit verbalization could be used to perform synthetic data generation to learn spatial grounding. Explicit verbalization can be extracted directly from the image. Given an image in an existing dataset that contains spatial relations (MS COCO), we propose to use an object detector to identify the entities in the images. Then, we could create hand-designed verbalization templates to automatically generate textual descriptions of the spatial relations among them.

Finally, we could train existing LMs in a self-supervised way using the synthetic dataset. We could test various methods for verbalization to know which is the right way to verbalize spatial information for effective spatial grounding. Then, we can test if we can improve the state-of-the-art of vision and language models in tasks that require spatial reasoning.

7.1.2 Text-to-Image Generation

Large generative **text-to-image diffusion models**, like DALLE-2 [12] and IMAGEN [13], are able to generate stunning images. They are known to possess some visual-reasoning skills [18]. However, in our work, we have seen that Stable Diffusion makes many mistakes when generating images, which require compositional reasoning.

A recent work [19] has also shown that these models **struggle to understand the composition of some concepts**, such as confusing the attributes and relations of different objects. They propose a new method, where an image is generated by composing a set of diffusion models, with each of them modelling a certain component of the image. Yet, this approach requires modifying the input prompts and is limited only to conjunction (and) and negation (not). Winoground examples are more complex, and can not be simplified to a conjunction of objects.

Another work [20] proposes manipulating cross-attention representations to address three challenging phenomena in Stable Diffusion [17]: **attribute leakage**, **interchanged attributes** and **missing objects** (see Figure 7.1). They achieve better compositional skills in qualitative and quantitative results, leading to a very significant 5-8% advantage in head-to-head user comparison studies. Many of the problems when generating Winoground images are of this type. Therefore, this might improve results when generating Winoground images.

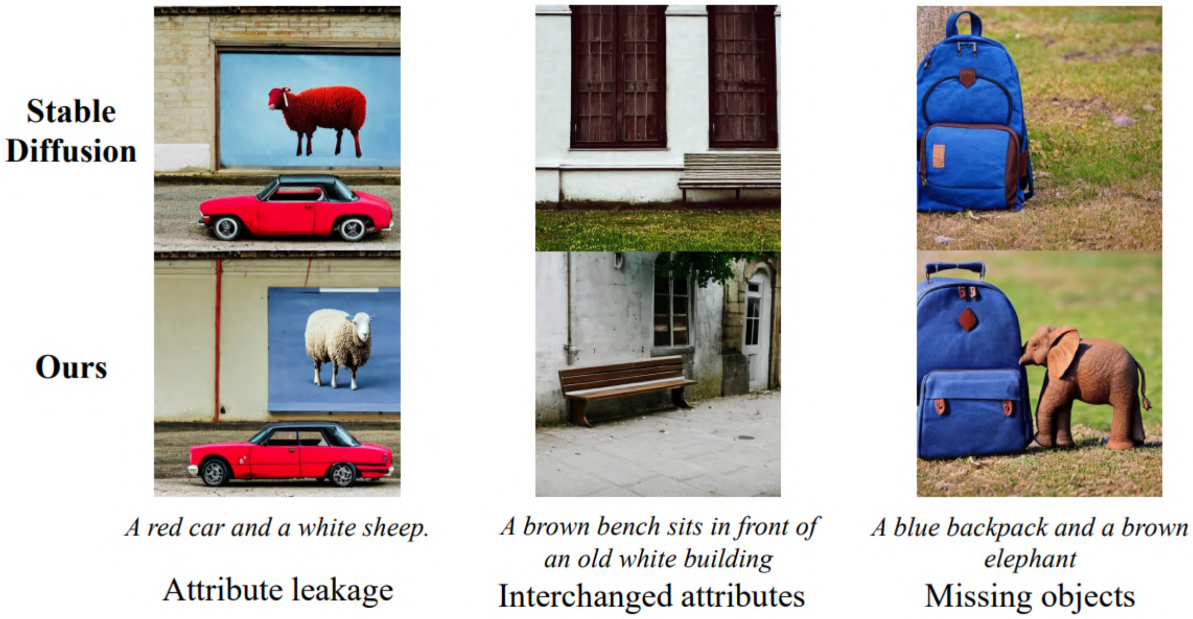

Stable Diffusion
Ours
A red car and a white sheep.
Attribute leakage
A brown bench sits in front of an old white building
Interchanged attributes
A blue backpack and a brown elephant
Missing objects

Figure 7.1: Three challenging phenomena in the compositional generation. Attribute leakage: The attribute of one object appears in another object. Interchanged attributes: the attributes of two or more objects are interchanged. Missing objects: one or more objects are missing.

7.1.3 Image-to-Image Generation

Apart from text-to-image generation, diffusion models can also be used for image-to-image generation. Stable Diffusion [17] could be used to generate new images as a data augmentation technique. For example, multiple image **variations** can be generated from an input image, to get similar images that still match the original caption. We could also change the caption if we are interested in getting similar images with different objects.

The problem with variations is that even a small change in the text prompt can lead to a completely different outcome. Usually, we only want to edit a small part of an image, without affecting the rest of the image. State-of-the-art methods solve this with **in-painting**, by using a spatial mask to localize the edit, leaving the rest of the image untouched. The drawback of this approach is that spatial masks have to be specifically crafted for each edit.

A recent solution called CycleDiffusion [78] can be applied to text-to-image diffusion models such as Stable Diffusion for zero-shot image editing (Figure 7.2). The input is composed of three elements: a source image, a source text and a target text. Within each pair of source and target text, overlapping text spans are marked in purple in the source text and abbreviated as [. . .] in the target text.

Another solution for this is to do image editing only using text prompts, that is, **prompt-to-prompt** editing [79]. This technique allows many types of edits such as localized editing by replacing a word,

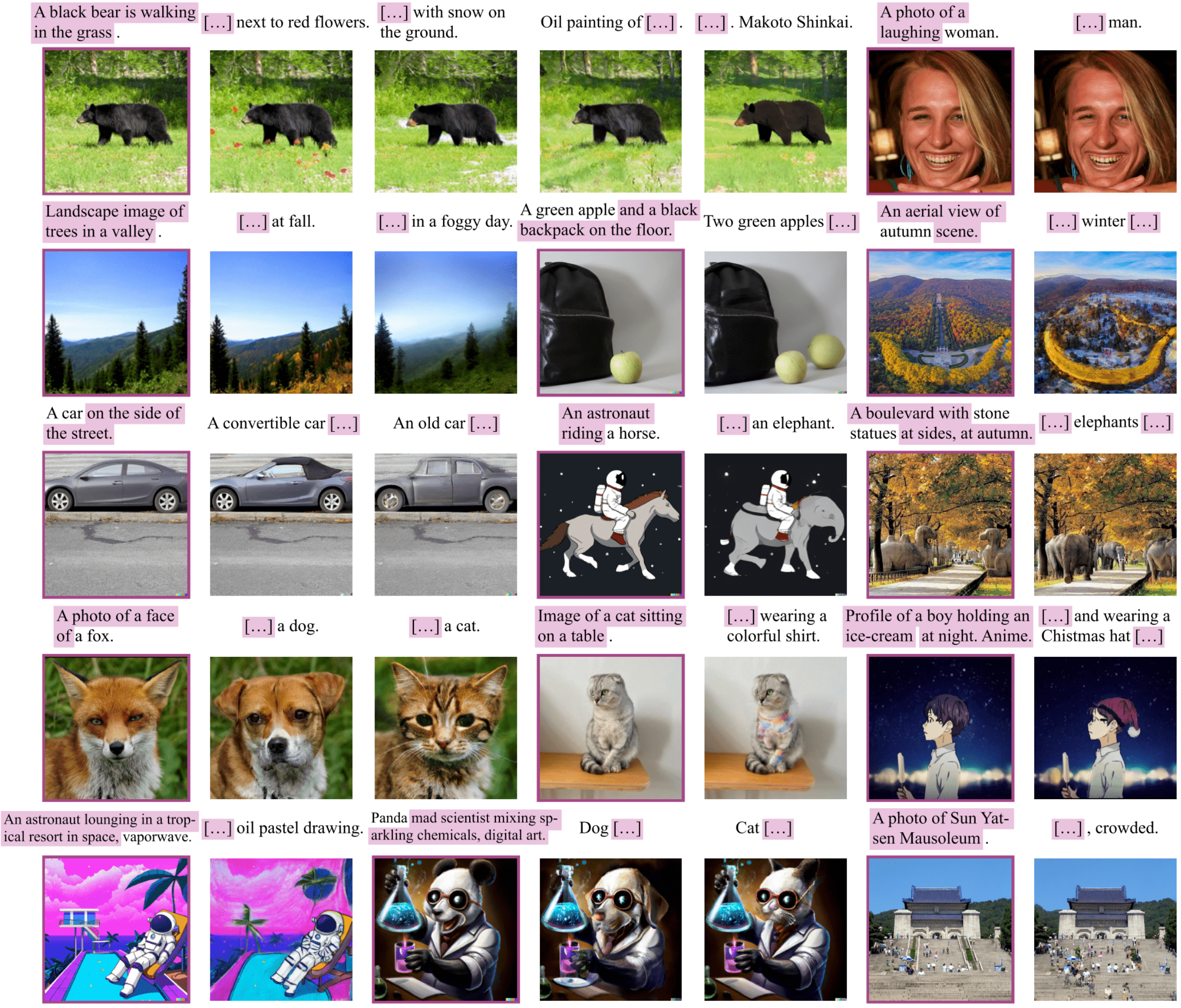

A black bear is walking in the grass .
[...] next to red flowers.
[...] with snow on the ground.
Oil painting of [...] .
[...] . Makoto Shinkai.
A photo of a laughing woman.
[...] man.
Landscape image of trees in a valley .
[...] at fall.
[...] in a foggy day.
A green apple and a black backpack on the floor.
Two green apples [...]
An aerial view of autumn scene.
[...] winter [...]
A car on the side of the street.
A convertible car [...]
An old car [...]
An astronaut riding a horse.
[...] an elephant.
A boulevard with stone statues at sides, at autumn.
[...] elephants [...]
A photo of a face of a fox.
[...] a dog.
[...] a cat.
Image of a cat sitting on a table .
[...] wearing a colorful shirt.
Profile of a boy holding an ice-cream at night. Anime.
[...] and wearing a Chismas hat [...]
An astronaut lounging in a tropical resort in space, vaporwave.
[...] oil pastel drawing.
Panda mad scientist mixing sparkling chemicals, digital art.
Dog [...]
Cat [...]
A photo of Sun Yatsen Mausoleum .
[...] , crowded.

Figure 7.2: With CycleDiffusion text-to-image diffusion models can be used as zero-shot image-to-image editors. Source images are displayed with a purple margin and others are generated target images. CycleDiffusion achieves minimal editing that includes replacing objects, adding objects, changing image styles, and modifying attributes.

and global editing by adding a specification and controlling the extent to which a word is reflected in the image (Figure 7.3).

Imagic [80] only uses an input image and a target text to edit images (see Figure 7.4). Imagic can apply complex semantic edits such as changing the posture and composition of one or multiple objects inside an image while preserving its original characteristics. For example, it can make a standing dog sit down or jump, cause a bird to spread its wings, etc. These are precisely the types of edits that we need to create a synthetic dataset.

UniTune [81] is another new text-driven image editing method. UniTune gets as input an image and a textual edit description, and carries out the edit while maintaining high semantic and visual fidelity to the input image (see Figure 7.5). UniTune proves that large text-to-image diffusion models can be fine-tuned on a single image, encouraging the model to maintain fidelity to the input image while still allowing expressive manipulations.

7.1.4 Image Captioning and Retrieval

In this work, we have seen that generating synthetic captions can be a good approach to automatically describe images at a large scale. We have also seen that image retrieval can be used to retrieve images from a huge dataset given a caption or an image. In the future, captioning could be paired with image

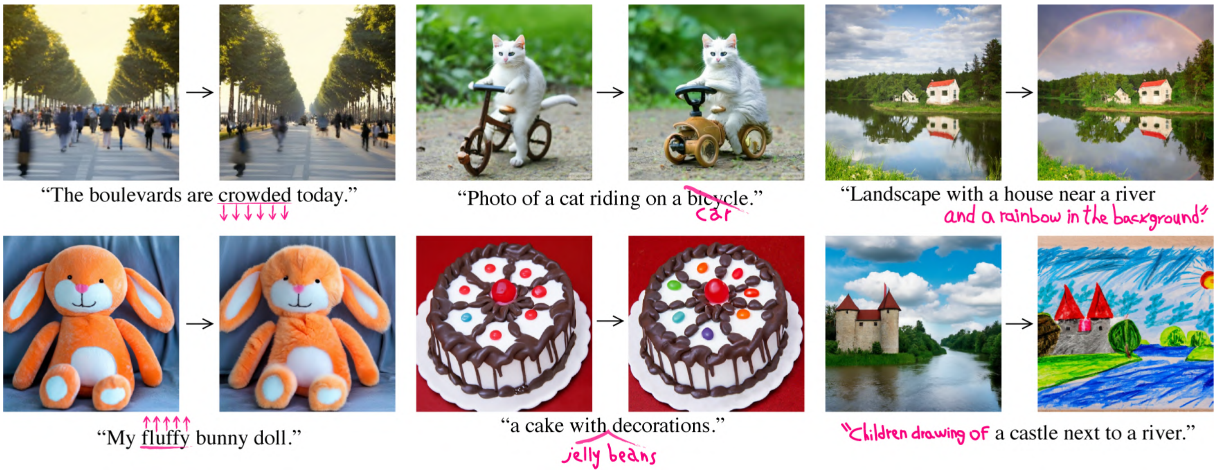

"The boulevards are crowded today."
"Photo of a cat riding on a bicycle."
car
"Landscape with a house near a river
and a rainbow in the background!"
"My fluffy bunny doll."
"a cake with decorations."
jelly beans
"Children drawing of a castle next to a river."

Figure 7.3: Prompt-to-Prompt editing operations: tuning the level of influence of an adjective word (left), making a local modification in the image by replacing or adding a word (middle), or specifying a global modification (right).

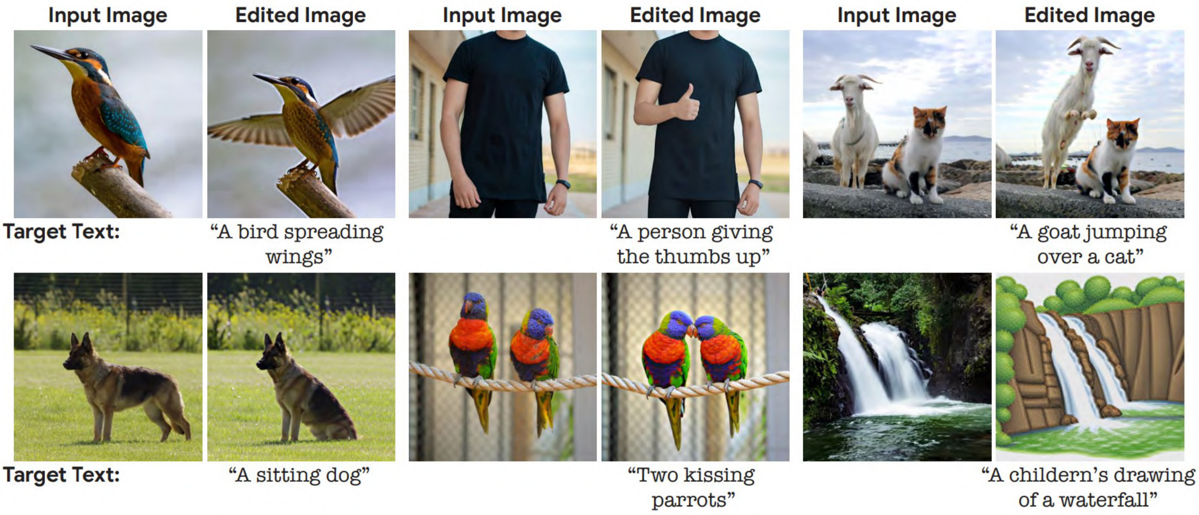


Figure 7.4: Imagic can perform various text-based semantic edits on a single real input image, including highly complex non-rigid changes such as posture changes and editing multiple objects. Here, we show pairs of input images and edited outputs with their respective target texts.

retrieval to create a synthetic dataset. This way, we can retrieve images that we are interested in, and generate good captions for them.

LAION-COCO[1] is a new dataset that follows a similar approach. It is the world's largest dataset of this type, with 600M generated high-quality captions for public images from LAION2B-EN [58]. LAION5B already has natural captions, but these captions are generally not very good and could be completed by synthetic ones. These captions could be used to train VLMs to investigate how they impact the performance of models.

The model used to generate captions is the same one that we used (BLIP L/16), but they do some extra steps to increase caption quality. First, they generate 40 captions at a time, which are later ranked using CLIP L/14 to select the best 5 captions. Then, those captions are ranked using CLIP RN50x64 to select the best one. Finally, a small fine-tuned T0 model is used to repair the grammar and punctuation errors in the texts.

They evaluated these captions by asking human evaluators to guess whether it corresponds to a

[1] https://laion.ai/blog/laion-coco/

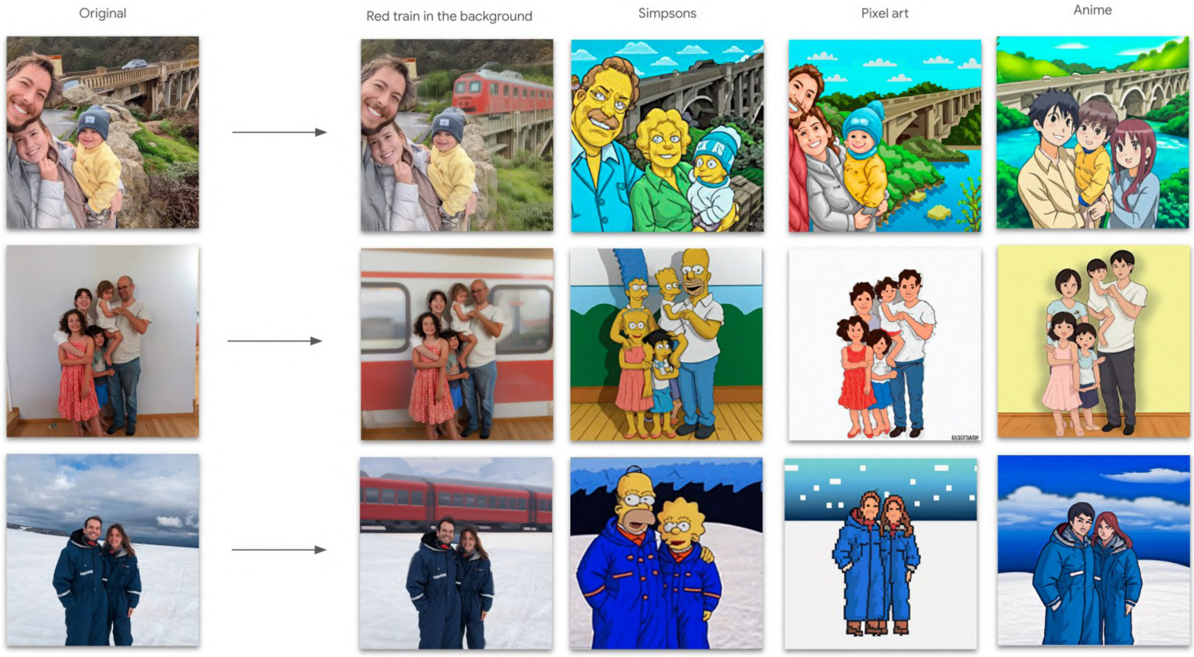


Figure 7.5: Samples showing UniTune's ability to maintain semantic details even across broad visual changes, and to place edits in a logical manner.

human or an AI model. They also asked them to rate the quality on a scale from 0 (bad) to 5 (good). They presented each evaluator with 200 samples, that contained 100 AI-generated and 100 human-written MS COCO captions. They conclude that caption quality is on average pretty close to the human-written captions of MS COCO. Filters could be further improved by rating more images by humans and removing low-score images.

Image captioning could also be used to extend current datasets to other languages. Multilingual VLMs could be used to caption the same image and get captions in many languages. A recent dataset called Crossmodal-3600 aims to evaluate multilingual image captioning [82]. It contains a geographically-diverse set of 3600 images annotated with human-generated reference captions in 36 languages. This dataset could be used to evaluate multilingual models and decide if they are good enough to generate synthetic datasets.

7.2 Multilingual Datasets

Another direction is extending Winoground and VSR to cover more languages and cultures and testing multilingual VLMs. There are already some multilingual visual reasoning datasets. MaRVL (Multicultural Reasoning over Vision and Language) [83] consists of discriminating whether each grounded statement about a pair of images is true or false. It focuses on a typologically diverse set of languages, Indonesian, Mandarin Chinese, Swahili, Tamil, and Turkish. IGLUE (Image-Grounded Language Understanding Evaluation) [84] brings together visual question answering, cross-modal retrieval, grounded reasoning, and grounded entailment tasks across 20 diverse languages.

Winoground is English-only and translation to other languages may be difficult. The key aspect of Winoground is that both captions contain the same words. Translating to other languages and maintaining this characteristic might be very difficult, and probably impossible in some examples. Moreover, expert translation is time-consuming and that limits the size of the translated datasets.

VSR seems to be easier to translate because it has no restrictions about the used words. However, spatial relations can be very different across languages. Some relations that are common in English might not exist in other languages and vice versa. In addition, word order is different from English in many languages, and that hinders template-based caption creation.

Apart from evaluation datasets, good pretraining multilingual datasets are needed to create multilingual VLMs. For example, LAION-5B [58] dataset was automatically collected from the web. It has two multilingual subsets, one with an unknown language (LAION1B-nolang) and the other one with a known (LAION2B-multi). Recently, a new dataset called LAION-translated[2] was released based on the previous ones. Every caption of the original datasets was translated to English with Facebook's M2M100 1.2B model. These captions can be used to train a multilingual VLM using aligned pairs.

[2] https://laion.ai/blog/laion-translated/

A Winoground Discussion

This appendix shows extra Winoground discussion results of previous and our experiments. We include correlation analysis between caption perplexity and model scores (Appendix A.1). Appendix A.2 includes correlation analysis between pretraining dataset size and model scores.

A.1 Caption Complexity

Previous. Table A.1 shows on the left correlations between **GPT-2 caption perplexity** and model scores. Winoground authors found that there is typically a weak correlation between models assigning an image-caption pair a higher score and a caption having low perplexity [9].

Table A.1 shows on the right that **caption length** and lower model performance significantly correlate for the best models, even though the correlation is reversed for humans. The examples with the shortest captions are also the least compositional [9].

	Perplexity		Caption Length					
	Text-Image		Text		Image		Group	
Model	Corr.	p-value	Corr.	p-value	Corr.	p-value	Corr.	p-value
MTurk Human	0.05	0.07	**0.11**	**0.03**	**0.20**	**0.00**	**0.20**	**0.00**
VinVL	**-0.05**	**0.04**	**-0.11**	**0.03**	**-0.18**	**0.00**	**-0.20**	**0.00**
UNITER$_{large}$	-0.01	0.57	-0.08	0.13	-0.06	0.20	**-0.16**	**0.00**
UNITER$_{base}$	-0.03	0.22	**-0.15**	**0.00**	**-0.11**	**0.03**	**-0.14**	**0.00**
ViLLA$_{large}$	-0.02	0.39	-0.05	0.32	**-0.13**	**0.01**	**-0.12**	**0.01**
ViLLA$_{base}$	-0.04	0.13	**-0.14**	**0.01**	**-0.12**	**0.01**	**-0.11**	**0.03**
VisualBERT$_{base}$	-0.04	0.15	-0.09	0.07	-0.07	0.14	-0.06	0.22
ViLT (ViT-B/32)	-0.04	0.16	-0.09	0.06	**-0.20**	**0.00**	**-0.16**	**0.00**
LXMERT	-0.04	0.12	-0.00	0.97	-0.05	0.32	**-0.11**	**0.02**
ViLBERT$_{base}$	-0.04	0.11	-0.09	0.09	**-0.15**	**0.00**	**-0.14**	**0.00**
UniT$_{ITM Finetuned}$	-0.01	0.73	-0.03	0.53	-0.05	0.32	-0.02	0.73
FLAVA$_{ITM}$	-0.03	0.22	**-0.21**	**0.00**	**-0.22**	**0.00**	**-0.23**	**0.00**
FLAVA$_{ITC}$	**-0.06**	**0.01**	**-0.15**	**0.00**	**-0.25**	**0.00**	**-0.19**	**0.00**
CLIP (ViT-B/32)	-0.04	0.09	**-0.27**	**0.00**	**-0.19**	**0.00**	**-0.22**	**0.00**
VSE++$_{COCO}$ (ResNet)	**-0.05**	**0.04**	-0.03	0.60	-0.02	0.74	0.01	0.90
VSE++$_{COCO}$ (VGG)	-0.04	0.08	-0.02	0.65	0.03	0.50	0.03	0.56
VSE++$_{Flickr30k}$ (ResNet)	-0.02	0.43	-0.01	0.80	0.01	0.91	0.02	0.67
VSE++$_{Flickr30k}$ (VGG)	0.01	0.74	-0.09	0.07	-0.07	0.18	**-0.10**	**0.04**
VSRN$_{COCO}$	**-0.07**	**0.01**	-0.03	0.60	-0.05	0.30	-0.05	0.36
VSRN$_{Flickr30k}$	-0.02	0.32	-0.03	0.60	-0.10	0.06	-0.05	0.29

Table A.1: (left) The correlation between model image-caption scores and the caption perplexity from GPT2. (right) The correlation between the model text, image and group scores and the caption length.

Ours. We shows correlations between caption perplexity and our model scores in Table A.2. Similar to previous results, for most models there is a weak correlation. Interestingly, most configurations of the best performing model (BLIP) show some correlation between caption perplexity and image-caption scores. Regarding caption length, there is a significant correlation for most models.

A.2 Pretraining Dataset Size

Previous. Winoground authors found highly significant correlations between the size of the multi-modal pretraining dataset and the scores, if CLIP and FLAVA were removed as outliers. Table A.3 shows these correlations, and Figure A.1 has graphs showing each model's score versus the pretraining data size.

	Perplexity		Caption Length					
	Image-Caption		Text		Image		Group	
Model	Corr.	p-value	Corr.	p-value	Corr.	p-value	Corr.	p-value
MTurk Human	0.05	0.07	**0.11**	**0.03**	**0.20**	**0.00**	**0.20**	**0.00**
ViLT (ViT-B/32)	-0.04	0.08	**-0.12**	**0.02**	-0.07	0.17	-0.05	0.35
ViLT$_{COCO}$ (ViT-B/32)	-0.05	0.06	**-0.21**	**0.00**	**-0.16**	**0.00**	**-0.17**	**0.00**
ViLT$_{Flickr30k}$ (ViT-B/32)	**-0.05**	**0.03**	**-0.11**	**0.03**	**-0.17**	**0.00**	**-0.14**	**0.01**
ViLT$_{NLVR2}$ (ViT-B/32)	0.00	0.95	**-0.13**	**0.01**	**-0.11**	**0.03**	**-0.12**	**0.02**
ViLT$_{VSR}$ Random (ViT-B/32)	**-0.07**	**0.01**	**-0.16**	**0.00**	**-0.19**	**0.00**	**-0.15**	**0.00**
ViLT$_{VSR}$ Zero-shot (ViT-B/32)	**-0.08**	**0.00**	**-0.14**	**0.01**	**-0.15**	**0.00**	**-0.15**	**0.00**
FLAVA$_{ITM}$	-0.03	0.22	**-0.21**	**0.00**	**-0.22**	**0.00**	**-0.23**	**0.00**
FLAVA$_{ITC}$	**-0.06**	**0.01**	**-0.15**	**0.00**	**-0.25**	**0.00**	**-0.19**	**0.00**
CLIP (ViT-B/32)	-0.04	0.10	**-0.28**	**0.00**	**-0.21**	**0.00**	**-0.23**	**0.00**
CLIP (ViT-B/16)	-0.04	0.11	**-0.26**	**0.00**	**-0.22**	**0.00**	**-0.23**	**0.00**
CLIP (ViT-L/14)	-0.03	0.22	**-0.22**	**0.00**	**-0.17**	**0.00**	**-0.18**	**0.00**
CLIP (ViT-L/14-336)	-0.04	0.11	**-0.23**	**0.00**	**-0.22**	**0.00**	**-0.23**	**0.00**
OpenCLIP (ViT-B/32)	-0.04	0.16	**-0.13**	**0.01**	**-0.16**	**0.00**	**-0.18**	**0.00**
OpenCLIP (ViT-L/14)	-0.04	0.14	**-0.22**	**0.00**	**-0.23**	**0.00**	**-0.23**	**0.00**
OpenCLIP (ViT-H/14)	-0.04	0.13	**-0.22**	**0.00**	**-0.26**	**0.00**	**-0.24**	**0.00**
OpenCLIP (ViT-g/14)	-0.04	0.07	**-0.26**	**0.00**	**-0.21**	**0.00**	**-0.20**	**0.00**
OFA$_{Tiny}$	-0.01	0.66	**-0.17**	**0.00**	-0.06	0.24	**-0.12**	**0.02**
OFA$_{Base}$	-0.02	0.43	**-0.15**	**0.00**	**-0.12**	**0.02**	**-0.10**	**0.05**
OFA$_{Medium}$	-0.01	0.77	**-0.11**	**0.03**	**-0.14**	**0.00**	**-0.12**	**0.01**
OFA$_{Large}$	**-0.16**	**0.00**	**-0.18**	**0.00**	**-0.20**	**0.00**	**-0.17**	**0.00**
OFA$_{Huge}$	0.01	0.75	**-0.15**	**0.00**	**-0.17**	**0.00**	**-0.16**	**0.00**
BLIP$_{ITM14M}$ (ViT-B/16)	-0.00	0.85	**-0.22**	**0.00**	**-0.23**	**0.00**	**-0.21**	**0.00**
BLIP$_{ITC14M}$ (ViT-B/16)	-0.00	0.97	**-0.24**	**0.00**	**-0.17**	**0.00**	**-0.17**	**0.00**
BLIP$_{ITM}$ (ViT-B/16)	**-0.05**	**0.04**	**-0.24**	**0.00**	**-0.23**	**0.00**	**-0.22**	**0.00**
BLIP$_{ITC}$ (ViT-B/16)	**-0.06**	**0.02**	**-0.19**	**0.00**	**-0.17**	**0.00**	**-0.13**	**0.01**
BLIP$_{ITM}$ (ViT-B/16) (CapFilt-L)	**-0.10**	**0.00**	**-0.20**	**0.00**	**-0.28**	**0.00**	**-0.23**	**0.00**
BLIP$_{ITC}$ (ViT-B/16) (CapFilt-L)	**-0.10**	**0.00**	**-0.25**	**0.00**	**-0.17**	**0.00**	**-0.15**	**0.00**
BLIP$_{ITM}$ (ViT-L/16)	**-0.07**	**0.01**	**-0.17**	**0.00**	**-0.21**	**0.00**	**-0.19**	**0.00**
BLIP$_{ITC}$ (ViT-L/16)	**-0.08**	**0.00**	**-0.22**	**0.00**	**-0.17**	**0.00**	**-0.17**	**0.00**
BLIP$_{ITMCOCO}$ (ViT-B/16)	-0.04	0.11	**-0.17**	**0.00**	**-0.26**	**0.00**	**-0.22**	**0.00**
BLIP$_{ITCCOCO}$ (ViT-B/16)	**-0.06**	**0.02**	**-0.18**	**0.00**	**-0.26**	**0.00**	**-0.22**	**0.00**
BLIP$_{ITMFlickr30k}$ (ViT-B/16)	-0.04	0.11	**-0.25**	**0.00**	**-0.28**	**0.00**	**-0.28**	**0.00**
BLIP$_{ITCFlickr30k}$ (ViT-B/16)	**-0.07**	**0.00**	**-0.20**	**0.00**	**-0.19**	**0.00**	**-0.18**	**0.00**
BLIP$_{ITMCOCO}$ (ViT-L/16)	**-0.06**	**0.02**	**-0.24**	**0.00**	**-0.23**	**0.00**	**-0.23**	**0.00**
BLIP$_{ITCCOCO}$ (ViT-L/16)	**-0.10**	**0.00**	**-0.21**	**0.00**	**-0.21**	**0.00**	**-0.21**	**0.00**
BLIP$_{ITMFlickr30k}$ (ViT-L/16)	**-0.05**	**0.04**	**-0.27**	**0.00**	**-0.25**	**0.00**	**-0.23**	**0.00**
BLIP$_{ITCFlickr30k}$ (ViT-L/16)	**-0.09**	**0.00**	**-0.24**	**0.00**	**-0.19**	**0.00**	**-0.16**	**0.00**
BLIP$_{NLVR2}$ (ViT-B/16)	-0.00	0.85	**-0.22**	**0.00**	-0.08	0.09	**-0.16**	**0.00**

Table A.2: (left) The correlation between model image-caption scores and the caption perplexity from GPT2. (right) The correlation between the model text, image and group scores and the caption length.

Pretraining	Score	Corr.	p-value
Image	Text	**0.84**	**0.00**
	Image	**0.76**	**0.00**
	Group	**0.75**	**0.00**
Caption	Text	**0.77**	**0.00**
	Image	**0.75**	**0.00**
	Group	**0.71**	**0.00**

Table A.3: Correlations between the number of pretraining images and captions and the model text, image, and group scores. CLIP and FLAVA are excluded as outliers.

Ours. We do the same analysis excluding ViLT as an outlier and find that there is no significant correlation. The models we test are pre-trained on large datasets, and the best ones are not the ones that are trained on most data. For example, BLIP is the best model and uses much less data than OpenCLIP for pre-training. Table A.4 shows these correlations, and Figure A.2 has graphs showing each model's score versus the pretraining data size.

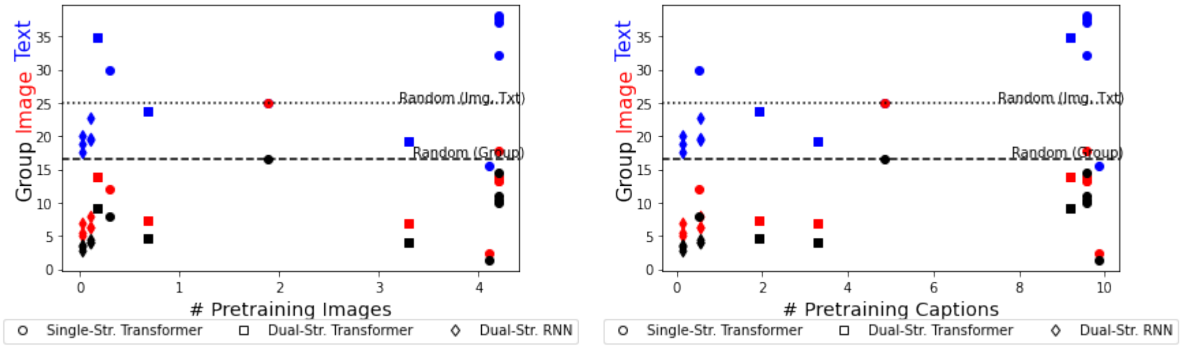


Figure A.1: Graphs of the model performance on Winoground for each model by the number of pretraining images (left) and pretraining captions (right).

Pretraining	Score	Corr.	p-value
Image	Text	-0.11	0.54
	Image	-0.28	0.12
	Group	-0.24	0.18
Caption	Text	-0.11	0.54
	Image	-0.28	0.12
	Group	-0.24	0.18

Table A.4: Correlations between the number of pretraining images and captions and the model text, image, and group scores. ViLT is excluded as outlier.

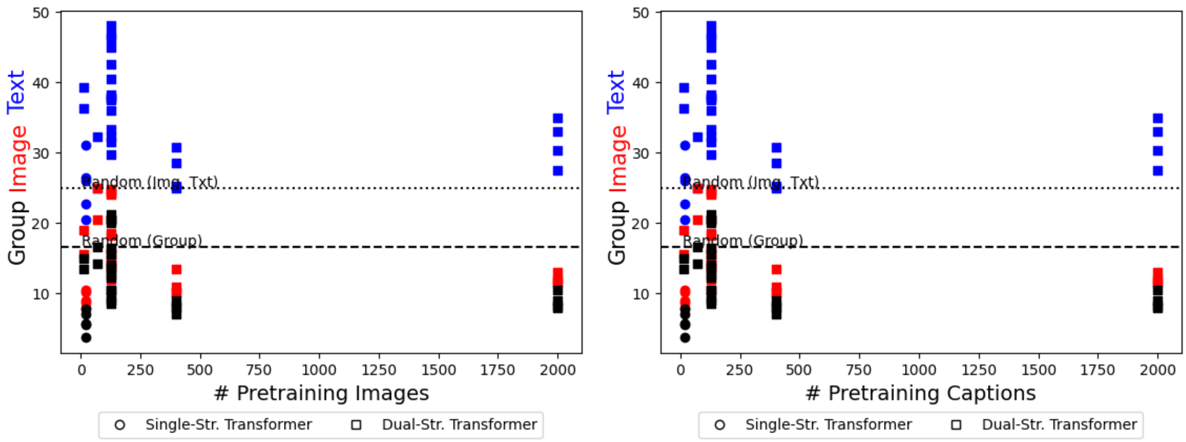


Figure A.2: Graphs of the model performance on Winoground for each model by the number of pretraining images (left) and pretraining captions (right).

B VSR Results

This appendix shows VSR result tables by relation (Appendix B.1) and relation meta category (Appendix B.2).

B.1 Results By Relation

We show performance by relation of our models on random and zero-shot splits in Tables B.1 and B.2. Only the most common relations are shown and they are sorted from top to bottom by frequency.

relation	number	VisualBERT	LXMERT	ViLT	ViLT NLVR2	BLIP NLVR2
all	2024	55.1	73.9	71.2	59.1	60.1
touching	236	55.9	76.7	73.7	64.0	62.3
behind	136	44.9	75.0	70.6	52.9	58.1
on	128	64.8	82.0	86.7	71.9	70.3
in front of	116	54.3	70.7	63.8	58.6	65.5
under	112	62.5	85.7	83.9	62.5	66.1
on top of	87	50.6	79.3	79.3	72.4	67.8
at the right side of	85	51.8	76.5	57.6	63.5	50.6
at the left side of	80	48.8	73.8	61.3	50.0	56.2
beneath	80	63.7	80.0	77.5	58.8	56.2
above	72	59.7	76.4	72.2	55.6	62.5
contains	57	56.1	80.7	86.0	56.1	50.9
in	51	68.6	82.4	84.3	60.8	58.8
facing	50	50.0	64.0	62.0	60.0	62.0
far away from	49	51.0	77.6	75.5	40.8	42.9
inside	49	59.2	77.6	79.6	57.1	55.1
below	42	59.5	66.7	66.7	47.6	52.4
next to	41	56.1	68.3	75.6	53.7	65.9
at the edge of	40	42.5	47.5	60.0	50.0	62.5
left of	39	56.4	76.9	59.0	59.0	56.4
beside	34	44.1	73.5	64.7	79.4	67.6
facing away from	32	56.2	53.1	46.9	56.2	50.0
away from	31	61.3	71.0	74.2	41.9	64.5
right of	24	50.0	87.5	58.3	58.3	54.2
far from	23	47.8	87.0	87.0	43.5	56.5
close to	21	57.1	71.4	71.4	71.4	57.1
part of	21	42.9	76.2	76.2	42.9	42.9
near	21	52.4	57.1	71.4	76.2	66.7
parallel to	19	31.6	36.8	57.9	52.6	47.4
at the back of	19	57.9	73.7	63.2	52.6	63.2
across from	18	66.7	72.2	66.7	44.4	44.4
over	16	50.0	75.0	93.8	81.2	56.2
in the middle of	15	46.7	60.0	33.3	33.3	53.3
off	15	33.3	40.0	40.0	26.7	46.7

Table B.1: Our model performance by relation on the random split test. Only relations with more than 15 occurrences are shown.

B.2 Results By Meta Category

We show performance by relation meta category of our models on random and zero-shot splits in Tables B.3 and B.4. Categories are sorted alphabetically.

relation	number	VisualBERT	LXMERT	ViLT	ViLT NLVR2	BLIP NLVR2
all	731	50.8	65.5	61.6	52.8	53.9
in front of	76	46.1	64.5	53.9	50.0	52.6
behind	71	49.3	78.9	69.0	50.7	49.3
far away from	57	57.9	59.6	59.6	40.4	36.8
at the left side of	32	59.4	71.9	50.0	59.4	71.9
next to	32	40.6	62.5	62.5	81.2	65.6
contains	29	48.3	86.2	75.9	48.3	55.2
touching	27	55.6	48.1	77.8	55.6	74.1
at the right side of	25	44.0	48.0	68.0	52.0	72.0
on	23	52.2	87.0	78.3	73.9	82.6
above	22	54.5	59.1	59.1	54.5	45.5
left of	22	59.1	86.4	54.5	54.5	54.5
on top of	20	40.0	80.0	90.0	85.0	85.0
under	20	45.0	60.0	45.0	45.0	40.0
below	18	66.7	61.1	61.1	61.1	66.7
in	16	37.5	87.5	87.5	62.5	75.0
parallel to	14	35.7	42.9	50.0	64.3	42.9
far from	14	57.1	71.4	71.4	28.6	42.9
facing	14	50.0	42.9	78.6	71.4	57.1
at the back of	14	71.4	64.3	50.0	35.7	35.7
across from	14	42.9	57.1	57.1	28.6	28.6
ahead of	13	30.8	53.8	38.5	53.8	61.5
away from	12	50.0	41.7	50.0	33.3	41.7
beside	12	41.7	41.7	66.7	41.7	25.0
adjacent to	12	83.3	66.7	33.3	58.3	58.3
right of	12	58.3	75.0	58.3	50.0	33.3
beneath	11	54.5	63.6	54.5	54.5	45.5
facing away from	10	60.0	70.0	60.0	60.0	50.0
inside	8	50.0	62.5	75.0	37.5	75.0
close to	8	62.5	50.0	37.5	50.0	37.5
beyond	6	33.3	66.7	66.7	33.3	66.7
alongside	6	33.3	66.7	66.7	50.0	50.0
off	6	66.7	50.0	50.0	16.7	33.3
surrounding	5	40.0	100.0	60.0	40.0	80.0

Table B.2: Our model performance by relation on the zero-shot split test. Only relations with more than 5 occurrences are shown.

category	number	VisualBERT	LXMERT	ViLT	ViLT NLVR2	BLIP NLVR2
All	2024	55.1	73.9	71.2	59.1	60.1
Adjacency	284	51.4	71.1	63.0	56.7	60.2
Directional	90	56.7	68.9	55.6	47.8	54.4
Orientation	112	50.9	55.4	54.5	55.4	56.2
Proximity	123	52.0	73.2	74.8	53.7	52.8
Projective	773	54.5	76.7	71.7	59.8	61.4
Topological	591	59.2	76.8	79.2	63.5	61.4
Unallocated	51	52.9	64.7	74.5	54.9	60.8

Table B.3: Our model performance by relation meta category on the random split test.

category	number	VisualBERT	LXMERT	ViLT	ViLT NLVR2	BLIP NLVR2
All	731	50.8	65.5	61.6	52.8	53.9
Adjacency	114	55.3	62.3	52.6	55.3	61.4
Directional	40	50.0	52.5	52.5	25.0	40.0
Orientation	42	47.6	47.6	59.5	64.3	47.6
Proximity	83	59.0	59.0	57.8	41.0	37.3
Projective	286	50.0	69.6	61.2	53.5	51.0
Topological	124	48.4	74.2	76.6	55.6	67.7
Unallocated	42	38.1	64.3	61.9	71.4	64.3

Table B.4: Our model performance by relation meta category on the zero-shot split test.

Bibliography

[1] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019. See page 1.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. See page 1.

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. See page 1.

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. See page 1.

[5] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5185–5198, 2020. See page 1.

[6] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021. See page 1.

[7] Stephen C Levinson and Stephen C Levinson. *Space in language and cognition: Explorations in cognitive diversity*. Cambridge University Press, 2003. See page 1.

[8] Arjun Akula, Spandana Gella, Yaser Al-Onaizan, Song-Chun Zhu, and Siva Reddy. Words aren't enough, their order matters: On the robustness of grounding visual referring expressions. In *ACL*, 2020. See page 1.

[9] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022. See pages 1, 2, 3, 6, 11, 18, 19, 21, 22, 25, 26, 47, and 55.

[10] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*, 2019. See pages 1, 5, 6, 8, and 22.

[11] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP-IJCNLP*, 2020. See pages 1, 2, 5, 6, 22, 31, and 47.

[12] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. See pages 1, 12, and 49.

[13] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. See pages 1, 12, and 49.

[14] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015. See pages 1, 17.

[15] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018. See pages 1, 17.

[16] Xiao Liu, Da Yin, Yansong Feng, and Dongyan Zhao. Things not written in text: Exploring spatial commonsense from visual signals. *arXiv preprint arXiv:2203.08075*, 2022. See page 1.

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. See pages 1, 2, 5, 12, 13, 37, 47, and 50.

[18] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generative transformers. *arXiv preprint arXiv:2202.04053*, 2022. See pages 1, 37, and 49.

[19] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. *arXiv preprint arXiv:2206.01714*, 2022. See pages 1, 37, and 49.

[20] Anonymous. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *Submitted to The Eleventh International Conference on Learning Representations*, 2023. under review. See pages 1, 50.

[21] Hessam Bagherinezhad, Hannaneh Hajishirzi, Yejin Choi, and Ali Farhadi. Are elephants bigger than butterflies? reasoning about sizes of objects. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. See page 1.

[22] Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. How large are lions? inducing distributions over quantitative attributes. *arXiv preprint arXiv:1906.01327*, 2019. See page 1.

[23] Guillem Collell, Luc Van Gool, and Marie-Francine Moens. Acquiring common sense spatial knowledge through implicit spatial templates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. See page 1.

[24] Aitzol Elu, Gorka Azkune, Oier Lopez de Lacalle, Ignacio Arganda-Carreras, Aitor Soroa, and Eneko Agirre. Inferring spatial relations from textual descriptions of images. *Pattern Recognition*, 113:107847, 2021. See page 1.

[25] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *arXiv preprint arXiv:2205.00363*, 2022. See pages 2, 3, 6, 18, 29, 30, 31, 32, 33, 34, 35, and 47.

[26] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. In *arXiv preprint arXiv:1908.03557*, 2019. See pages 2, 5, 6, 7, 22, 31, and 47.

[27] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021. See pages 2, 5, 6, 9, 22, 23, 31, 32, and 47.

[28] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. See pages 2, 5, 6, 10, 23, 32, 39, and 47.

[29] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*, 2022. See pages 5, 11, 23, 39, and 47.

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. See pages 5, 6, 10, 13, 22, 23, and 24.

[31] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. If you use this software, please cite it as below. See pages 5, 10, and 24.

[32] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *CVPR*, 2022. See pages 5, 6, 9, 22, and 23.

[33] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *arXiv preprint arXiv:2102.10772*, 2021. See pages 5, 6, 7, and 22.

[34] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020. See pages 5, 6, 7, and 22.

[35] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. In *NeurIPS*, 2020. See pages 5, 6, 8, and 22.

[36] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, 2021. See pages 5, 6, 8, and 22.

[37] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *BMVC*, 2018. See pages 5, 11, 12, and 22.

[38] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *ICCV*, 2019. See pages 5, 11, 12, and 22.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. See pages 5, 17, and 18.

[40] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *arXiv preprint arXiv:1602.07332*, 2016. See page 5.

[41] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. See page 5.

[42] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. See page 5.

[43] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *TACL*, 2014. See page 5.

[44] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. See page 5.

[45] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, 2019. See page 5.

[46] Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *ACL*, 2017. See page 5.

[47] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment task for visually-grounded language learning. In *arXiv preprint arXiv:1811.10582*, 2018. See page 5.

[48] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *arXiv preprint arXiv:1606.05250*, 2016. See page 5.

[49] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *arXiv preprint arXiv:1704.05426*, 2017. See page 5.

[50] Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. First quora dataset release: Question pairs, 2017. See page 5.

[51] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *ECCV*, 2020. See page 5.

[52] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *arXiv preprint arXiv:2103.01913*, 2021. See page 5.

[53] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. See page 5.

[54] Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. Redcaps: Web-curated image-text data created by the people. In *NeurIPS Datasets and Benchmarks*, 2021. See page 5.

[55] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. In *Communications of the ACM*, 2016. See page 5.

[56] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013. See page 5.

[57] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. See page 5.

[58] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, mehdi cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Mitchell Wortsman, Richard Vencu, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 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. See pages 5, 10, 24, 42, 47, 52, and 54.

[59] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. See page 6.

[60] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. See pages 6, 10.

[61] Lisa Anne Hendricks, John Mellor, Rosalia Schneider, Jean-Baptiste Alayrac, and Aida Nematzadeh. Decoupling the role of data, attention, and losses in multimodal transformers. In *arXiv preprint arXiv:2102.00529*, 2021. See page 6.

[62] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020. See page 8.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. See page 12.

[64] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for largescale image recognition. In *CVPR*, 2015. See page 12.

[65] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. See page 13.

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. See page 13.

[67] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 39–48. IEEE Computer Society, 2016. See page 15.

[68] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. See page 15.

[69] Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, Vancouver, Canada, July 2017. Association for Computational Linguistics. See page 15.

[70] Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598, Online, 2021. Association for Computational Linguistics. See page 15.

[71] Haojun Jiang, Yuanze Lin, Dongchen Han, Shiji Song, and Gao Huang. Pseudo-q: Generating pseudo language queries for visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15513–15523, 2022. See page 16.

[72] Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. Unnatural language inference. In *ACL*, 2020. See page 18.

[73] Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *EMNLP*, 2021. See page 18.

[74] Cristiane Kutianski Marchi Fagundes, Kristin Stock, and Luciene Delazari. A cross-linguistic study of spatial location descriptions in new zealand english and brazilian portuguese natural language. *Trans. GIS*, 25(6):3159–3187, 2021. See page 29.

[75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. See page 31.

[76] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software, 2020-2022. Open source software available from https://github.com/heartexlabs/label-studio. See page 38.

[77] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational*

Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. See page 39.

[78] Chen Henry Wu and Fernando De la Torre. Unifying diffusion models' latent space, with applications to cyclediffusion and guidance. *arXiv preprint arXiv:2210.05559*, 2022. See page 50.

[79] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. See page 50.

[80] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models, 2022. See page 51.

[81] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning an image generation model on a single image, 2022. See page 51.

[82] Ashish V Thapliyal, Jordi Pont-Tuset, Xi Chen, and Radu Soricut. Crossmodal-3600: A massively multilingual multimodal evaluation dataset. *arXiv preprint arXiv:2205.12522*, 2022. See page 53.

[83] Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. Visually grounded reasoning across languages and cultures. In *EMNLP*, 2021. See page 53.

[84] Emanuele Bugliarello, Fangyu Liu, Jonas Pfeiffer, Siva Reddy, Desmond Elliott, Edoardo Maria Ponti, and Ivan Vulić. IGLUE: A benchmark for transfer learning across modalities, tasks, and languages. *ArXiv preprint*, abs/2201.11732, 2022. See page 53.