PRE-TRAINING METHODS FOR VISION AND LANGUAGE

Hao Tan

A thesis submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2021

Approved by:

Mohit Bansal

Ron Alterovitz

Jason Baldridge

Shahriar Nirjon

Shashank Srivastava

ABSTRACT

Hao Tan: Pre-Training Methods for Vision and Language
(Under the direction of Mohit Bansal)

Vision and language are the primary modalities of our human perception and learning. Recent years have witnessed fast development of methods that connect vision and language. Current deep learning methods are data-hungry, thus pre-training on large-scale data helps warm up the model and shows better fine-tuning results on downstream tasks. However, pre-training frameworks that exploit the power of multi-modality are still underexplored. Specifically, we have the following questions remaining: Could we build large pre-trained models that understand the interactions and alignments between modalities? Could language and vision help the understanding of each other? Could we combine the current diverse methods for vision pre-training and language pre-training? This dissertation aims to answer these questions. I first build a vision-and-language pre-training framework: LXMERT. This pre-training framework learns vision-and-language joint representations from massive data (e.g., MS COCO) and achieves state-of-the-art results on several benchmark tasks such as image question answering and visual reasoning. We also illustrate the importance of single-modality pre-training in vision-and-language tasks. Next, I improve language understanding via dense visual supervision and show its generalization to pure-text tasks. I develop the vokenization method to construct this visual supervision, which learns to retrieve related images for each contextualized token in the sentence. Lastly, current language pre-training and vision pre-training are led by different pretext tasks: language modeling and contrastive learning. I combine these two methods into a unified pre-training framework on videos, such that the pre-trained model could capture both static spatial contents and dynamic temporal interactions.

To my parents.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Prof. Mohit Bansal for all the support and encouragement over these five years. I can't imagine myself in this position without his guidance. This dissertation is the result of his constant support and freedom to explore various challenging problems. I am grateful for all the feedback and the skills that I learned from him which I believe will carry for the rest of my life.

Throughout these five years, I have got various opportunities to work with a lot of amazing people through internships and collaborations. I want to thank all of them for sharing their valuable knowledge and experience with me. Especially, I would like to mention few of them. I thank Trung Bui, Franck Dernoncourt, and Zhe Lin to help me understand the importance of a research project for industry need. I thank Vihan Jain, Eugene Ie, and Jason Baldridge to teach me the scalability of a research project. I thank Chen-Tse Tsai, Yujie He, and Anju Kambadur for providing me a view of NLP research in finance and information. I would also like to thank Thomas Wolf for helping me to better understand the importance of deeper thinking on a research problem.

I would like to thank my committee members (Mohit Bansal, Ron Alterovitz, Jason Baldridge, Shahriar Nirjon, Shashank Srivastava) for their constant support and for providing valuable feedback in completing this dissertation. I also thank the UNC-NLP lab members for always there to help me out and also thank them for having a fun and interactive PhD experience. Among them, I want to share special thank to Licheng Yu who teaches me vision knowledge. I thank Ramakanth Pasunuru for helping me a lot in my PhD studies. I thank Yixin Nie and Jie Lie, who discuss research and brainstorm with me the most.

I would like to thank my parents: Qian Tan and Yingping Zang, my two support pillars. They have always encouraged me to pursue whatever I want to do. Without their support, I wouldn’t have pursued PhD.

PhD is a long journey and there are ups and downs, and without my partner Mengyun, it would have been harder to navigate. I would like to thank her for always being there to support me and share my ups and downs. Also, I am fortunate to have many friends over these five years who have at some scale contributed to a better life as a Phd Student. I would like to thank those friends, specifically, Shi Feng, Zhewei Yao, Hongyuan Mei, Tianze Shi, Wenkang Yu, Sihang Zhou,

I am grateful to be supported by a Bloomberg PhD Fellowship. I am also thankful for the support from other grants: NSF, DARPA, ARO, ONR, IBM, Nvidia, Google, Bloomberg, Salesforce, Facebook, Baidu, and Amazon.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
GLUE	General Language Understanding Evaluation
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
NLVR	Natural Language for Visual Reasoning (Dataset)
RL	Reinforcement Learning
VL	Natural Language Processing
VLP	Vision-and-Language Pre-Training
VQA	Visual Question Answering

CHAPTER 1: INTRODUCTION

Vision and language are the two major modalities for human to perceive the world. We rely on these two modalities to learn and to communicate with others. To build an embodied agent which interact with humans, the ability to handle multiple modalities is necessary. Researchers have studied diverse problems (Kazemzadeh et al., 2014; Chen et al., 2015; Antol et al., 2015; Plummer et al., 2015; Anderson et al., 2018b) in vision-and-language and have built different learning systems (Gerber and Nagel, 1996; Farhadi et al., 2010; Yao et al., 2010; Aker and Gaizauskas, 2010), using deep learning techniques (Vinyals et al., 2015; Xu et al., 2015; Lu et al., 2016; Yang et al., 2016). Current machine learning systems, especially the models with neural networks, are data-hungry and the performance scales well with the amount of data (Sun et al., 2017; Kaplan et al., 2020). However, the amount of clean human-annotated data is limited by the collection budget and is far from saturating the model capacity. In order to deal with this data shortage, the "pre-training and fine-tuning" paradigm is developed and became the primary approach in current machine learning research. In this paradigm, the model is first pre-trained on large-scale less-annotated or unannotated data with pretext tasks (He et al., 2020a). Then these well-initialized models are fine-tuned on downstream tasks, usually with much smaller data size. In the past decade, we have experienced the power of pre-training in vision (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016; Huang et al., 2017) and in language (Mikolov et al., 2013; Pennington et al., 2014; Peters et al., 2018; Devlin et al., 2019a; Radford et al., 2018; Yang et al., 2019; Lan et al., 2019) thus it is a strong belief to see the same improvement in multi-modal tasks. In this thesis, we illustrate the possibility of building vision-and-language pre-training frameworks and present some solutions.

There are three key aspects in building pre-training frameworks: the neural model, the data, and the pre-training (pretext) tasks. Vision-and-language models are usually built with three components: the visual encoder, the language encoder, and the fusion module. The visual encoder and language encoder converts the visual input (e.g., images, videos) and language input (e.g., sentences, paragraphs) to dense vectors. Then the fusion module takes these vectors as input. For detailed model design, the visual encoder usually uses a convolutional neural network (He et al., 2016) as backbone. Recurrent neural networks like LSTM (Hochreiter and Schmidhuber, 1997) were the popular choice for the language encoder until Transformer (Vaswani et al., 2017) becomes a preferring alternative. Fusion module needs to model the interaction between two modalities thus the Transformer model with attention layers (Xu et al., 2015) is suitable. Since pre-training needs large data amount, captioning data is the first choice since they are easier to annotate (Chen et al., 2015; Young et al., 2014; Krishna et al., 2017) or could be gathered from the web (Thomee et al., 2016; Sharma et al., 2018; Radford et al., 2021a; Changpinyo et al., 2021). The captioning data is constructed with pairs of images and sentences, where the sentence describes the visual content of the image. We will discuss the vision-and-language data in Sec. 2.2 in detail. For pre-training methods, the vision and language communities prefer different pretext tasks. Visual models are traditionally pre-trained with classification task on large annotated datasets (Deng et al., 2009) but recently transferred to the unsupervised contrastive learning (Oord et al., 2018; He et al., 2020a; Chen et al., 2020b). For language pre-training, word-level pre-trained vectors (Mikolov et al., 2013; Pennington et al., 2014) are first developed by maximizing the mutual information of word distributions (Levy and Goldberg, 2014; Oord et al., 2018). Since the development of ELMo (Peters et al., 2018), language modeling became the most popular pretext task given its natural connections to human language understanding.

To build a vision-and-language pre-training framework, we need to gather these three pieces as well: suitable models, large-scale data, and appropriate pre-training tasks. LXMERT (Tan and Bansal, 2019) is among the first few realizations of such pre-training frameworks. In this work, we build a full-Transformer (Vaswani et al., 2017) model that the visual encoder, the language encoder, and the fusion module are all built in Transformer. Since Transformer model only takes sequential data as input, it has an issue to directly process the two-dimensional images. We thus use a detection system (Ren et al., 2015; Anderson et al., 2018a) to convert the image into a sequence of objects.[1] For pre-training dataset, we consider the captioning data (Chen et al., 2015; Krishna et al., 2017) and image questions answering data (Goyal et al., 2017b; Hudson and Manning, 2019), which are the public large-scale image-and-sentence data resources. For pre-training tasks, we consider the cross-modality masked language modeling and the cross-modal matching tasks. The first task tries to predict masked information from both modalities, and the second task verifies whether the image and the sentence semantically match. We evaluate our pre-trained LXMERT model on several downstream tasks (e.g., visual question answering, visual reasoning) and observe that the performance is significantly improved.

The vision-and-language pre-training method (Tan and Bansal, 2019) trains all model parameters from scratch, but still has an independent visual module to convert images into objects. This 'bottom-up attention' (Anderson et al., 2018a) visual module is a Faster R-CNN (Ren et al., 2015) trained on Visual Genome (Krishna et al., 2017) detection data. This single-modality pre-trained vision encoder exists in most current vision-and-language systems (Lu et al., 2019; Chen et al., 2020d; Li et al., 2020b), and is biologically plausible that researchers has located the visual system in our brain (Wurtz et al., 2000). From practical considerations, the amount of aligned vision-and-language data is strictly less than the single-modality data (i.e., pure image, pure text). Thus, pre-trained visual system could benefit from larger amount of data. For these reasons, we study the impact of these pre-trained single-modality modules to multimodal tasks. We compare a list of vision modules from the traditional convolution neural networks to recent vision transformers. These models are pre-trained on different datasets as well, including image classification dataset, detection dataset, and also aligned web image-text dataset. As the architecture and visual pre-training methods evolve, we see a significant improvement on downstream vision-and-language tasks. This improvement is almost at the same level as the improvement that vision-and-

[1] Recent work ViT (Dosovitskiy et al., 2021) converts images to a sequence of patches instead and this approch is used in ViLT (Kim et al., 2021) for vision-and-language pre-training.

language pre-training brings. These results show that we still need a careful examination about single-modality training when looking at multimodal tasks.

Previously, we talked about pre-training on vision-and-language data, but only multimodal tasks are considered. However, learning in the multimodal world not only helps a better understanding of multimodal interactions but also is essential in building our communication ability (Bloom, 2002; Bender and Koller, 2020; Bisk et al., 2020). We thus want to show that single-modal tasks can also benefit from multimodal data by integrating more knowledge about the world. We develop a method, vokenization (Tan and Bansal, 2020), to simulate the pointing game for human language learning (Bloom, 2002). Our method first maps each word in the sentence to related images. Then the pre-training task for the language model is to predict these related images. With this external visual supervision, we empirically show that the model has a better performance on pure language tasks. This idea is further extended in our recent works VidLanKD (Tang et al., 2021) where we use knowledge distillation to replace the explicit voken mapping. Concurrently, VirTex (Desai and Johnson, 2020) proposes to pre-train the vision encoder with captioning model, showing the possibility of improving pure vision tasks from multimodal data. CLIP (Radford et al., 2021a) improves it by using a simpler contrastive model and a larger (400M) image-sentence dataset crawled from website.

Lastly, current vision pre-training and language pre-training methods are different from each other, with Contrastive Learning (CL) providing strong results for vision representation learning (Oord et al., 2018; He et al., 2020b; Chen et al., 2020b), and Language Modeling (LM) showing its strength in natural language processing (NLP) pre-training (Peters et al., 2018; Devlin et al., 2019a; Radford et al., 2019). We present VIMPAC (Tan et al., 2021) that combines the language modeling method and constrastive learning in video pre-training since video understanding naturally combines both characteristics of image and text. The 2D processing along the spatial dimensions of the video bears similarity to image processing, while 1D processing along the temporal dimension often involves modeling sequential events and short range coherence. We show that this model could achieve state-of-the-art results on two temporally-heavy action recognition tasks.

With these pre-training frameworks in vision and language, we make progress in moving uni-modal research to multi-modal research. We hope that this research would soon bring an impact to our real life by utilizing multi-modal search, multi-modal recommendation system, and the multi-modal AR/VR systems.

1.1 Thesis Statement

Through using massive data and large-scale models, it is possible to build multimodal pre-training frameworks that benefit both vision-and-language tasks and single-modality tasks.

1.2 Overview of Chapters

The remainder of this dissertation is organized into six chapters. Chapter 2 discusses the related work and background for vision-and-language tasks and data. Chapter 3 presents our work on building the first image-and-text pre-training frameworks, LXMERT. Chapter 4 discusses the impact of single-modality pre-training to multi-modal tasks. Chapter 5 shows how pure-language tasks to be improved from visual supervision by the vokenization method. Chapter 6 illustrates the natural combination of the language pre-training method and vision pre-training methods. Chapter 7 summarizes the contributions herein and discusses the potential opportunities for future work.

CHAPTER 2: BACKGROUND AND RELATED WORK

In this chapter, we discuss the background and related work for vision and language tasks (Sec. 2.1) and data (Sec. 2.2).

2.1 Vision-and-Language Tasks: An Overview

In this section, we give an overview of representative vision-and-language tasks. These vision-and-language tasks are categorized by their input and output types. For each task, we present a short story for their history and recent research progress. Note that it is not ana exclusive list of tasks and there are a lot of other vision-and-language tasks such as multimodal sentiment analysis, multimodal machine translation, e.t.c.

2.1.1 Vision to Language

Image Captioning Given an image as the input, image captioning task aims to generate a corresponding natural language that faithfully describe the visual content. It is useful in building website where alternative texts (known as the 'alt' attribute in HTML) significantly reduce the networking and are friendly to visually-impaired people. As a well-formulated task, it witnessed each progress in vision-and-language research in the deep learning era. Show-and-tell (Vinyals et al., 2015) proposes the encoder-decoder architecture that uses a convolutional neural network (Szegedy et al., 2015) as the vision encoder and a recurrent neural network (Hochreiter and Schmidhuber, 1997) as the language decoder. Show-Attend-and-Tell (Xu et al., 2015) uses attention mechanism (Mnih et al., 2014; Bahdanau et al., 2014) to bridge the two modality. This attention mechanism uses the feature map (i.e., a set of features) of the convolutional neural network, thus could break the information bottleneck introduced by the single-vector image feature. Sequence-level Training (Ranzato et al., 2015; Rennie et al., 2017) computes evaluation metric scores as rewards to the generation, and uses reinforcement learning technique to optimize the model. Bottom-Up Top-Down Attention (Anderson et al., 2018a) uses objects from the detection system as the input to the vision-and-language model, replacing the previously used grid features from convolutional neural networks. As pre-training methods developed, image captioning dataset serves as the major data resources and is used in almost every pre-training frameworks (Tan et al., 2019; Lu et al., 2019; Chen et al., 2020d; Li et al., 2020b). Recent works also show that image captioning data could help pure-language (Tan and Bansal, 2020) and pure-vision (Desai and Johnson, 2020; Radford et al., 2021a) tasks.

2.1.2 Language to Vision

Grounding Visual grounding (more specifically, object localization) maps the words in the sentence to the objects in the image. It builds an explicit connection between the text and image, thus could be used in follow-up processing (e.g., searching, recommendation). Flickr-30K dataset (Young et al., 2014) is an example dataset, but it has a strong bias towards common and large objects (Li et al., 2019a; Kamath et al., 2021). Referring expression (Kazemzadeh et al., 2014) is another types of grounding tasks that needs to differentiate a specific objects within a specific object types. Vision-and-language pre-training (Lu et al., 2019; Chen et al., 2019b) explicitly model this interaction thus showing the state-of-the-art performance.

Conditional Image Generation Conditional image generation aims to generate the image given a natural language text. It is the reverse task of image captioning but is much harder since image generation is more complicated than text generation. It also has an issue in evaluation, since most existing automatic scores focus on the fidelity, and are not good at measuring the semantic correspondence. Multiple works use text-to-image model and GAN loss (Reed et al., 2016; Xu et al., 2018; Zhang et al., 2017; Koh et al., 2021; Zhang et al., 2021a) to train the model. DALL-E (Ramesh et al., 2021) uses VQ-VAE (van den Oord et al., 2017) to quantize the image into discrete tokens then an unified language model could be used as image generation.

Image Retrieval Image retrieval tries to find the matched image from a large set of images given the natural language query. It is directly related to the problem of searching the image on search engines. The retrieval task could be viewed as a finite version of the image generation task. The evaluation metric is deterministic and more accurate, although still has false negative errors. The objective of image retrieval could not be directly optimized given the large negative samples. Hence, a matching score between image and sentence is usually optimized instead. It conducts on the image captioning data by hinge ranking loss (Gordo et al., 2016), binary classification loss (Tan and Bansal, 2019), or multi-way contrastive loss (Radford et al., 2021a). Vision-and-language pre-training frameworks (Tan and Bansal, 2019; Lu et al., 2019) integrate this loss into pre-training thus it naturally shows a large improvement given the bigger data amount. CLIP (Radford et al., 2021a) scales up the retrieval model training with 400M image-sentence pairs and a large batch size of 32,768. With this extreme large model and dataset, CLIP (Radford et al., 2021a) formulate vision tasks as a retrieval problem and shows strong zero-shot ability.

2.1.3 Vision and Language to Others

Visual Question Answering Visual question answering needs to answer a natural language question about the image content. To simplify the task, the datasets (Antol et al., 2015; Goyal et al., 2017b; Hudson and Manning, 2019) are usually presented in the classification version by providing a possible answer set. Due to the simple and faithful evaluation metric along with the large-scale data, visual question answering become a standard dataset to measure the ability of vision-and-language interactions. Previous VQA improvement is based on building better fusion models (Gao et al., 2016; Lu et al., 2016; Yang et al., 2016) with special pooling layers and attention mechanism. Bottom-up attention (Anderson et al., 2018a) proposes to use a detection system trained on fine-grained object annotations (Krishna et al., 2017). After that, different pre-training methods (Tan and Bansal, 2019; Lu et al., 2019) dominate the progress.

Vision-and-Language Navigation Vision-and-language navigation tests the agent's ability to take action according to human instructions, which recently gains popularity in embodied AI (An-

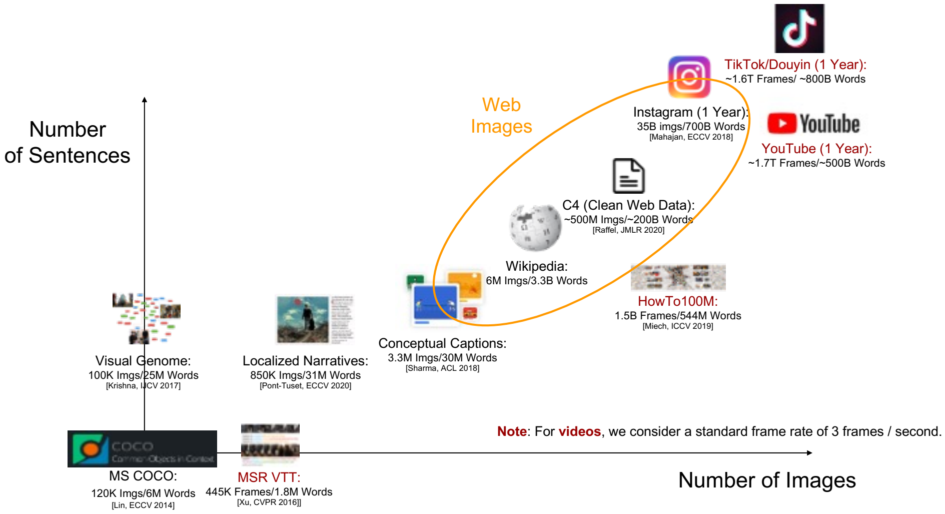
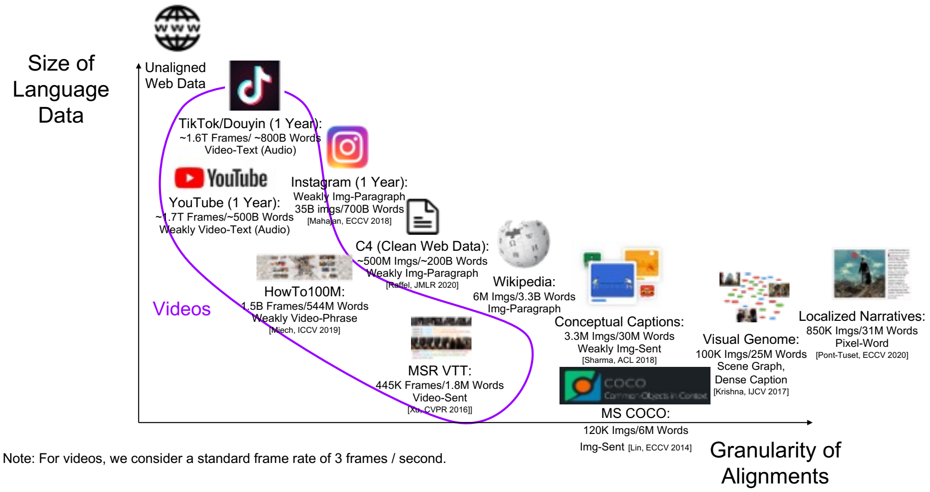


Figure 2.1: The amount of different vision-and-language data.

derson et al., 2018b; Chen et al., 2019a; Jain et al., 2019; Chen et al., 2019a; Qi et al., 2020b; Krantz et al., 2020; Nguyen and Daumé III, 2019; Ku et al., 2020). Specifically, the agent is put at a location in the environment (Chang et al., 2017) and asked to reach a target by following the language instructions. Different from visual question answering where we only face a static image, vision-and-language navigation requires exploring and understanding the dynamic environment to approach the target. Multiple works (Zhu et al., 2020; Hao et al., 2020; Hong et al., 2021) pre-train the model on the domain-specific room-to-room dataset (Anderson et al., 2018b). These domain-specific pre-training shows an improvement over generally-pretrained visoin-and-language systems (Li et al., 2020b; Hong et al., 2021).

2.2 Vision-and-Language Data

The amount of data is the major requirement to single-modality pre-training, e.g., number of images for visual pre-training and number of text tokens for language pre-training. However, the requirement of data in vision-and-language pre-training is more complex and finding suitable datasets actually becomes the bottleneck for vision-and-language research. Besides the need of large amount, the alignment between the visual content and text is also crucial. In this section, we discuss these available data from different viewpoints.



Note: For videos, we consider a standard frame rate of 3 frames / second.

Figure 2.2: The alignment granularity of different vision-and-language data.

2.2.1 Data Amount

As all pre-training methods require, we discuss the amount of data first from different data resources. As shown in Fig. 2.1, we visualize the amount of data by their number of images (the x-axis) and the text-token amounts (the y-axis). We list both image-and-text data and video-and-text data here and compute the amount of images in video dataset with a playing speed of 3 frames per second. In general, academic datasets (e.g., MS COCO, Visual Genome) are much smaller than the data from websites. However, these datasets are human-annotated thus are cleaned for research purpose. Conceptual Captions (Sharma et al., 2018) (CC) provide an opportunity to explore the web-level data in research community. CC heavily cleans the web image-text pairs with multiple cleaning stages. We also estimate the amount of data in three popular multimedia websites: Instagram, YouTube, and TikTok. These data are marginally available to our research community. We see some works exploring the Instagram data from company (Feichtenhofer et al., 2021). There is also a small version of the YouTube dataset, HowTo100M (Miech et al., 2019), which has a focus on instructional videos.

2.2.2 Data Alignment Granularity

Besides the amount of data, we also care about the alignment granularity in building vision-and-language pre-training systems. The alignment granularity could be considered as the 'annotation' of the data that provides supervision to vision-and-language pre-training For example, image captioning data (e.g., MS COCO) has image-to-sentence alignment while the video captioning data (e.g., MSR VTT (Xu et al., 2016)) has video-clip-to-sentence alignment. For other non-standard dataset, we have image-to-paragraph alignment in Wikipedia and video-to-sentence on YouTube. With these alignments, we know that the visual input and text data tell similar things, thus contrastive learning methods could be used. The alignments also tell that the vision and language data share the same context, so it enables the use of multimodal language modeling as the pre-training task as well. We show a series of different data in Fig. 2.2. We kept the y-axis to be the size of language data as in Fig. 2.1. For the x-axis, it indicates the granularity. We could observe that the human-annotated data are usually fine-grained, e.g., Localized Narratives (Pont-Tuset et al., 2019) has pixel-to-word annotation and Visual Genome (Krishna et al., 2017) has object-to-sentence annotation (i.e., dense captions). For web data, it shows a clear trend that larger data usually have weaker alignments and the trade-off between amount and quality naturally arises. Recent works are actively exploring these web data and fighting with this trade-off. Conceptual Captions (Sharma et al., 2018) aim to provide a clean dataset for research purpose. It utilizes a multi-stage filtering and sentence rewriting, but results in a large amount of data (about 99.8%) to be filtered out. In the follow-up work Changpinyo et al. (2021), the pipeline is improved and the dataset is enlarged by four times (i.e., 12M image-sentence pairs). At the same time, CLIP (Radford et al., 2021a) employs a weaker filtering strategy to retain much more data (400M image-sentence pairs). As the research community gradually moves to larger dataset (e.g., the video data in industry), we will face this problem time by time.

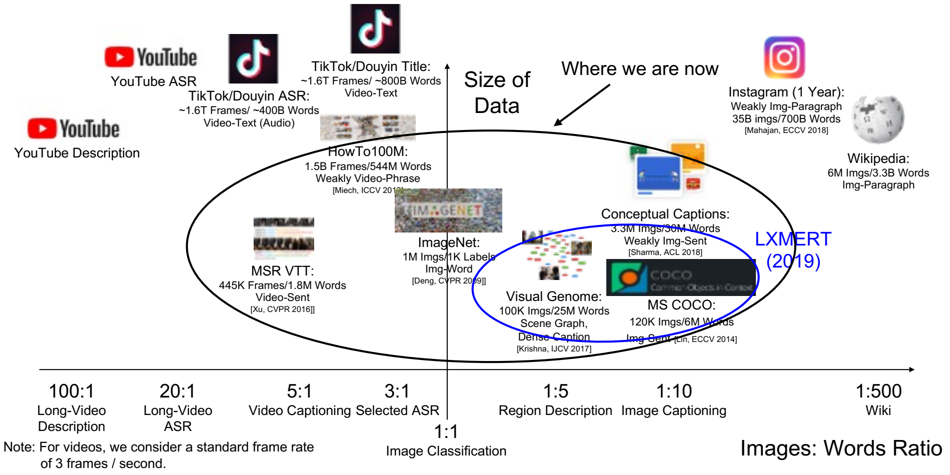


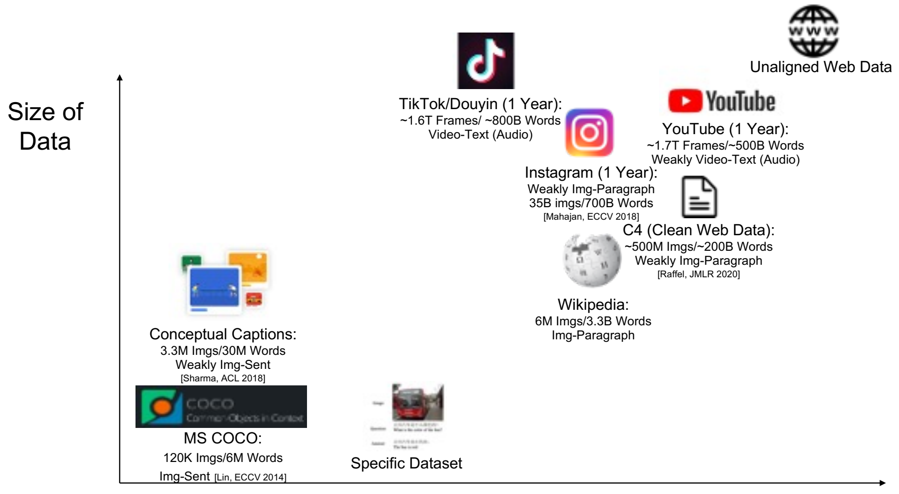
Figure 2.3: The modality balance of different vision-and-language data.

2.2.3 Data Modality Balance

In the previous section, we talked about the alignment's granularity. We take another look at the alignment by showing the modality balance. The modality balance is measured by ratio between paired images and words. As shown in Fig. 2.3, we illustrate the balance of some vision-and-language datasets. Academic datasets focus more on well-balanced data from the range of 10:1 (e.g., MSR VTT video captioning dataset) to 1:10 (e.g., MS COCO image captioning dataset). They mitigate the issue of imbalanced data and are suitable for current vision-and-language models where neural modules for different modalities are also balanced designed. However, the balance between modality in real-life applications is not as well as the ideal academic scenario. For YouTube descriptions, about 100 frames are described by one words in average. On the other side, an image is matched with 500 words in News and Wikipedia articles. As news and videos take a large part of the Internet, these imbalanced datasets proposes new research questions that we are seldom facing now. They might call for a study of new neural models and novel pre-training methods.

2.2.4 Data Multi-linguality

People with different language tries to communicate with each other by building a 'commonly-grounded knowledge space'. Visual objects, actions, and gestures are actively used to reach a



Note: For videos, we consider a standard frame rate of 3 frames / second.

Figure 2.4: The multi-linguality of different vision-and-language data.

consensus since different language are usually mapped to a similar external world. For this reason, vision-and-language data would be a valuable resource to explore the multi-lingual research. We thus illustrate the multi-linguality of each vision-and-language data resources. As shown in Fig. 2.4, we kept the y-axis to be the size of data and use x-axis to indicate the number of languages in each dataset. Current academic datasets (e.g., MS COCO, CC) are built fully on English. Researchers have created multiple specific datasets with other languages (Elliott et al., 2016; Wang et al., 2019d; Ku et al., 2020) but they are usually small. However, the Internet is open to people speaking diverse languages and the web data usually contains tens, hundreds, even thousands of different languages. Although we are currently focusing on English data (Radford et al., 2021a; Feichtenhofer et al., 2021) to prototype the vision-and-language models, it would be a great opportunity to consider different languages when exploring the web data.

CHAPTER 3: VISION-AND-LANGUAGE PRE-TRAINING

3.1 Introduction

Vision-and-language reasoning requires the understanding of visual contents, language semantics, and cross-modal alignments and relationships. There has been substantial past works in separately developing backbone models with better representations for the single modalities of vision and of language. For visual-content understanding, people have developed several backbone models (Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016) and shown their effectiveness on large vision datasets (Deng et al., 2009; Lin et al., 2014; Krishna et al., 2017). Pioneering works (Girshick et al., 2014; Xu et al., 2015) also show the generalizability of these pre-trained (especially on ImageNet) backbone models by fine-tuning them on different tasks. In terms of language understanding, last year, we witnessed strong progress towards building a universal backbone model with large-scale contextualized language model pre-training (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019b), which has improved performances on various tasks (Rajpurkar et al., 2016; Wang et al., 2019a) to significant levels. Despite these influential single-modality works, large-scale pretraining and fine-tuning studies for the modality-pair of vision and language are still under-developed.

Therefore, we present one of the first works in building a pre-trained vision-and-language cross-modality framework and show its strong performance on several datasets. We name this framework “LXMERT: Learning Cross-Modality Encoder Representations from Transformers” (pronounced: ‘leksmert’). This framework is modeled after recent BERT-style innovations while further adapted to useful cross-modality scenarios. Our new cross-modality model focuses on learning vision-and-language interactions, especially for representations of a single image and its descriptive sentence. It consists of three Transformer (Vaswani et al., 2017) encoders: an object relationship encoder, a language encoder, and a cross-modality encoder. In order to better learn the cross-modal alignments between vision and language, we next pre-train our model with five diverse representative tasks: (1) masked cross-modality language modeling, (2) masked object prediction via RoI-feature regression, (3) masked object prediction via detected-label classification, (4) cross-modality matching, and (5) image question answering. Different from single-modality pre-training (e.g., masked LM in BERT), this multi-modality pre-training allows our model to infer masked features either from the visible elements in the same modality, or from aligned components in the other modality. In this way, it helps build both intra-modality and cross-modality relationships.

Empirically, we first evaluate LXMERT on two popular visual question-answering datasets, VQA (Antol et al., 2015) and GQA (Hudson and Manning, 2019). Our model outperforms previous works in all question categories (e.g., Binary, Number, Open) and achieves state-of-the-art results in terms of overall accuracy. Further, to show the generalizability of our pre-trained model, we fine-tune LXMERT on a challenging visual reasoning task, Natural Language for Visual Reasoning for Real (NLVR^2) (Suhr et al., 2019), where we do not use the natural images in their dataset for our pre-training, but fine-tune and evaluate on these challenging, real-world images. In this setup, we achieve a large improvement of 22% absolute in accuracy (54% to 76%, i.e., 48% relative error reduction) and 30% absolute in consistency (12% to 42%, i.e., 34% relative error reduction). Lastly, we conduct several analysis and ablation studies to prove the effectiveness of our model components and diverse pre-training tasks by removing them or comparing them with their alternative options. Especially, we use several ways to take the existing BERT model and its variants, and show their ineffectiveness in vision-and-language tasks, which overall proves the need of our new cross-modality pre-training framework. We also present several attention visualizations for the different language, object-relationship, and cross-modality encoders.

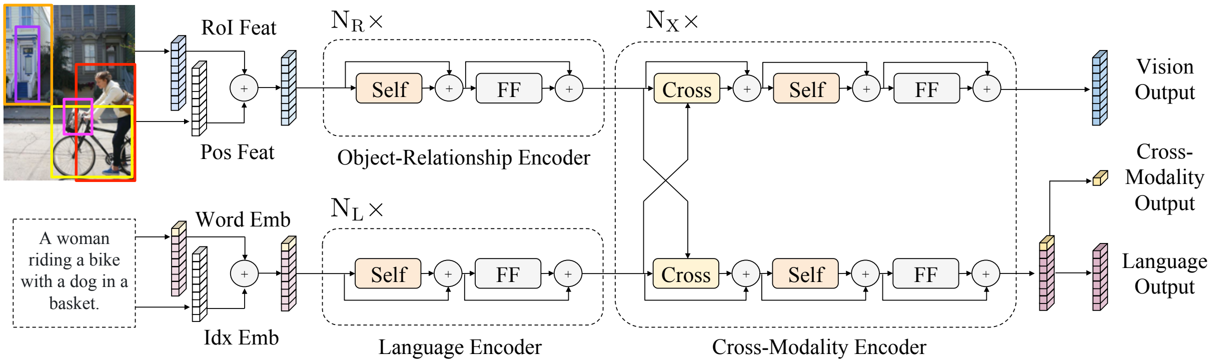


Figure 3.1: The LXMERT model for learning vision-and-language cross-modality representations. ‘Self’ and ‘Cross’ are abbreviations for self-attention sub-layers and cross-attention sub-layers, respectively. ‘FF’ denotes a feed-forward sub-layer.

3.2 Model Architecture

We build our cross-modality model with self-attention and cross-attention layers following the recent progress in designing natural language processing models (e.g., transformers (Vaswani et al., 2017)). As shown in Fig. 3.1, our model takes two inputs: an image and its related sentence (e.g., a caption or a question). Each image is represented as a sequence of objects, and each sentence is represented as a sequence of words. Via careful design and combination of these self-attention and cross-attention layers, our model is able to generate language representations, image representations, and cross-modality representations from the inputs. Next, we describe the components of this model in detail.

3.2.1 Input Embeddings

The input embedding layers in LXMERT convert the inputs (i.e., an image and a sentence) into two sequences of features: word-level sentence embeddings and object-level image embeddings. These embedding features will be further processed by the latter encoding layers.

Word-Level Sentence Embeddings A sentence is first split into words $\{w_1, \ldots, w_n\}$ with length of n by the same WordPiece tokenizer (Wu et al., 2016) in Devlin et al. (2019b). Next, as shown in Fig. 3.1, the word w_i and its index i (w_i's absolute position in the sentence) are projected to vectors by embedding sub-layers, and then added to the index-aware word embeddings:

$$\begin{aligned}
\hat{w}_i &= \mathrm{WordEmbed}\left(w_i\right) \\
\hat{u}_i &= \mathrm{IdxEmbed}\left(i\right) \\
h_i &= \mathrm{LayerNorm}\left(\hat{w}_i + \hat{u}_i\right)
\end{aligned}$$

Object-Level Image Embeddings Instead of using the feature map output by a convolutional neural network, we follow Anderson et al. (2018a) in taking the features of detected objects as the embeddings of images. Specifically, the object detector detects m objects $\{o_1, \dots, o_m\}$ from the image (denoted by bounding boxes on the image in Fig. 3.1). Each object o_j is represented by its position feature (i.e., bounding box coordinates) p_j and its 2048-dimensional region-of-interest (RoI) feature f_j. Instead of directly using the RoI feature f_j without considering its position p_j in Anderson et al. (2018a), we learn a position-aware embedding v_j by adding outputs of 2 fully-connected layers:

$$\begin{aligned}
\hat{f}_j &= \mathrm{LayerNorm}\left(W_{\mathrm{F}} f_j + b_{\mathrm{F}}\right) \\
\hat{p}_j &= \mathrm{LayerNorm}\left(W_{\mathrm{P}} p_j + b_{\mathrm{P}}\right) \\
v_j &= \left(\hat{f}_j + \hat{p}_j\right)/2 \qquad (3.1)
\end{aligned}$$

In addition to providing spatial information in visual reasoning, the inclusion of positional information is necessary for our masked object prediction pre-training task (described in Sec. 3.3.1.2). Since the image embedding layer and the following attention layers are agnostic to the absolute indices of their inputs, the order of the object is not specified. Lastly, in Equation 3.1, the layer normalization is applied to the projected features before summation so as to balance the energy of the two different types of features.

3.2.2 Encoders

We build our encoders, i.e., the language encoder, the object-relationship encoder, and the cross-modality encoder, mostly on the basis of two kinds of attention layers: self-attention layers and cross-attention layers. We first review the definition and notations of attention layers and then discuss how they form our encoders.

Background: Attention Layers Attention layers (Bahdanau et al., 2014; Xu et al., 2015) aim to retrieve information from a set of *context* vectors $\{y_j\}$ related to a *query* vector x. An attention layer first calculates the matching score a_j between the *query* vector x and each *context* vector y_j. Scores are then normalized by softmax:

$$\begin{aligned} a_j &= \text{score}(x, y_j) \\ \alpha_j &= \exp(a_j) / \sum\nolimits_k \exp(a_k) \end{aligned}$$

The output of an attention layer is the weighted sum of the *context* vectors w.r.t. the softmax-normalized score: $\text{Att}_{\text{X}\rightarrow\text{Y}}(x, \{y_j\}) = \sum_j \alpha_j y_j$. An attention layer is called *self-attention* when the *query* vector x is in the set of *context* vectors $\{y_j\}$. Specifically, we use the multi-head attention following Transformer (Vaswani et al., 2017).

Single-Modality Encoders After the embedding layers, we first apply two transformer encoders (Vaswani et al., 2017), i.e., a **language encoder** and an **object-relationship encoder**, and each of them only focuses on a single modality (i.e., language or vision). Different from BERT (Devlin et al., 2019b), which applies the transformer encoder only to language inputs, we apply it to vision inputs as well (and to cross-modality inputs as described later below). Each layer (left dashed blocks in Fig. 3.1) in a single-modality encoder contains a self-attention ('Self') sub-layer and a feed-forward ('FF') sub-layer, where the feed-forward sub-layer is further composed of two fully-connected sub-layers. We take $\mathbf{N_L}$ and $\mathbf{N_R}$ layers in the language encoder and the object-relationship encoder, respectively. We add a residual connection and layer normalization (annotated by the '+' sign in Fig. 3.1) after each sub-layer as in Vaswani et al. (2017).

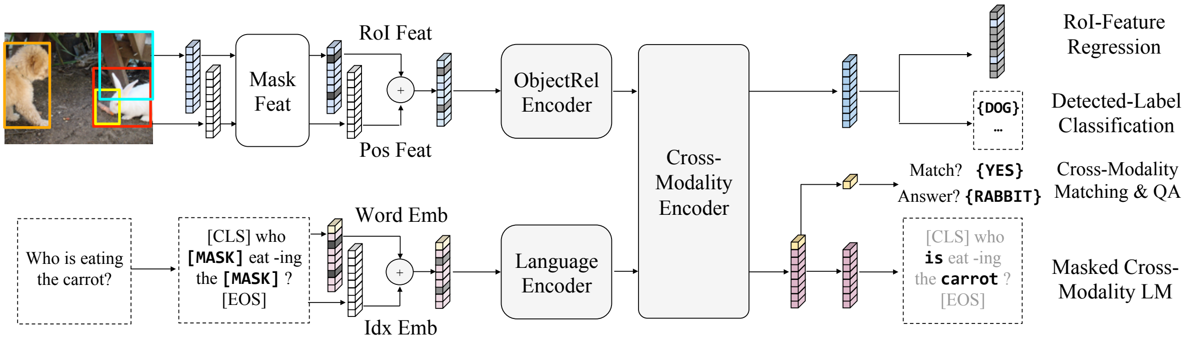


Figure 3.2: Pre-training in LXMERT. The object RoI features and word tokens are masked. Our five pre-training tasks learn the feature representations based on these masked inputs. Special tokens are in brackets and classification labels are in braces.

Cross-Modality Encoder Each cross-modality layer (the right dashed block in Fig. 3.1) in the cross-modality encoder consists of two self-attention sub-layers, one bi-directional cross-attention sub-layer, and two feed-forward sub-layers. We stack (i.e., using the output of k-th layer as the input of $(k{+}1)$-th layer) $\mathbf{N_X}$ these cross-modality layers in our encoder implementation. Inside the k-th layer, the bi-directional cross-attention sub-layer ('Cross') is first applied, which contains two uni-directional cross-attention sub-layers: one from language to vision and one from vision to language. The query and context vectors are the outputs of the $(k\text{-}1)$-th layer (i.e., language features $\{h_i^{k-1}\}$ and vision features $\{v_j^{k-1}\}$):

$$\begin{aligned}
\hat{h}_i^k &= \mathrm{CrossAtt}_{\mathrm{L}\rightarrow\mathrm{R}}\left(h_i^{k-1}, \{v_1^{k-1}, \dots, v_m^{k-1}\}\right) \\
\hat{v}_j^k &= \mathrm{CrossAtt}_{\mathrm{R}\rightarrow\mathrm{L}}\left(v_j^{k-1}, \{h_1^{k-1}, \dots, h_n^{k-1}\}\right)
\end{aligned}$$

The cross-attention sub-layer is used to exchange the information and align the entities between the two modalities in order to learn joint cross-modality representations. For further building internal connections, the self-attention sub-layers ('Self') are then applied to the output of the cross-attention sub-layer:

$$\begin{aligned}
\tilde{h}_i^k &= \mathrm{SelfAtt}_{\mathrm{L}\rightarrow\mathrm{L}}\left(\hat{h}_i^k, \{\hat{h}_1^k, \dots, \hat{h}_n^k\}\right) \\
\tilde{v}_j^k &= \mathrm{SelfAtt}_{\mathrm{R}\rightarrow\mathrm{R}}\left(\hat{v}_j^k, \{\hat{v}_1^k, \dots, \hat{v}_m^k\}\right)
\end{aligned}$$

Image Split	Images	Sentences (or Questions)					
		COCO-Cap	VG-Cap	VQA	GQA	VG-QA	All
MS COCO - VG	72K	361K	-	387K	-	-	0.75M
MS COCO $\cap$ VG	51K	256K	2.54M	271K	515K	724K	4.30M
VG - MS COCO	57K	-	2.85M	-	556K	718K	4.13M
All	180K	617K	5.39M	658K	1.07M	1.44M	9.18M

Table 3.1: Amount of data for pre-training. Each image has multiple sentences/questions. 'Cap' is caption. 'VG' is Visual Genome. Since MS COCO and VG share 51K images, we list it separately to ensure disjoint image splits.

Lastly, the k-th layer output $\{h_i^k\}$ and $\{v_j^k\}$ are produced by feed-forward sub-layers ('FF') on top of $\{\hat{h}_i^k\}$ and $\{\hat{v}_j^k\}$. We also add a residual connection and layer normalization after each sub-layer, similar to the single-modality encoders.

3.2.3 Output Representations

As shown in the right-most part of Fig. 3.1, our LXMERT cross-modality model has three outputs for language, vision, and cross-modality, respectively. The language and vision outputs are the feature sequences generated by the cross-modality encoder. For the cross-modality output, following the practice in Devlin et al. (2019b), we append a special token [CLS] (denoted as the top yellow block in the bottom branch of Fig. 3.1) before the sentence words, and the corresponding feature vector of this special token in language feature sequences is used as the cross-modality output.

3.3 Pre-Training Strategies

In order to learn a better initialization which understands connections between vision and language, we pre-train our model with different modality pre-training tasks on a large aggregated dataset.

3.3.1 Pre-Training Tasks

3.3.1.1 Language Task: Masked Cross-Modality LM

On the language side, we take the masked cross-modality language model (LM) task. As shown in the bottom branch of Fig. 3.2, the task setup is almost same to BERT (Devlin et al., 2019b): words are randomly masked with a probability of 0.15 and the model is asked to predict these masked words. In addition to BERT where masked words are predicted from the non-masked words in the language modality, LXMERT, with its cross-modality model architecture, could predict masked words from the vision modality as well, so as to resolve ambiguity. For example, as shown in Fig. 3.2, it is hard to determine the masked word 'carrot' from its language context but the word choice is clear if the visual information is considered. Hence, it helps building connections from the vision modality to the language modality, and we refer to this task as masked *cross-modality* LM to emphasize this difference. We also show that loading BERT parameters into LXMERT will do harm to the pre-training procedure in Sec. 3.5.1 since BERT can perform relatively well in the language modality without learning these cross-modality connections.

3.3.1.2 Vision Task: Masked Object Prediction

As shown in the top branch of Fig. 3.2, we pre-train the vision side by randomly masking objects (i.e., masking RoI features with zeros) with a probability of 0.15 and asking the model to predict proprieties of these masked objects. Similar to the language task (i.e., masked cross-modality LM), the model can infer the masked objects either from visible objects or from the language modality. Inferring the objects from the vision side helps learn the object relationships, and inferring from the language side helps learn the cross-modality alignments. Therefore, we perform two sub-tasks: **RoI-Feature Regression** regresses the object RoI feature f_j with L2 loss, and **Detected-Label Classification** learns the labels of masked objects with cross-entropy loss. In the 'Detected-Label Classification' sub-task, although most of our pre-training images have object-level annotations, the ground truth labels of the annotated objects are inconsistent in different datasets (e.g., different number of label classes). For these reasons, we take detected labels output by Faster R-CNN (Ren et al., 2015). Although detected labels are noisy, experimental results show that these labels contribute to pre-training in Sec. 3.5.3.

3.3.1.3 Cross-Modality Tasks

As shown in the middle-rightmost part of Fig. 3.2, to learn a strong cross-modality representation, we pre-train the LXMERT model with 2 tasks that explicitly need both language and vision modalities.

Cross-Modality Matching For each sentence, with a probability of 0.5, we replace it with a mismatched[1] sentence. Then, we train a classifier to predict whether an image and a sentence match each other. This task is similar to 'Next Sentence Prediction' in BERT (Devlin et al., 2019b).

Image Question Answering (QA) In order to enlarge the pre-training dataset (see details in Sec. 3.3.2), around $1/3$ sentences in the pre-training data are questions about the images. We ask the model to predict the answer to these image-related questions when the image and the question are matched (i.e., not randomly replaced in the cross-modality matching task). We show that pre-training with this image QA leads to a better cross-modality representation in Sec. 3.5.2.

3.3.2 Pre-Training Data

As shown in Table. 3.1, we aggregate pre-training data from five vision-and-language datasets whose images come from MS COCO (Lin et al., 2014) or Visual Genome (Krishna et al., 2017). Besides the two original captioning datasets, we also aggregate three large image question answering (image QA) datasets: VQA v2.0 (Antol et al., 2015), GQA balanced version (Hudson and Manning, 2019), and VG-QA (Zhu et al., 2016). We only collect **train and dev** splits in each dataset to avoid seeing any test data in pre-training. We conduct minimal pre-processing on the

[1] We take a sentence from another image as the mismatched sentence. Although the sentence and the image still have chance to match each other, this probability is very low.

Method	VQA				GQA			NLVR^2	
	Binary	Number	Other	**Accu**	Binary	Open	**Accu**	Cons	**Accu**
Human	-	-	-	-	91.2	87.4	89.3	-	96.3
Image Only	-	-	-	-	36.1	1.74	17.8	7.40	51.9
Language Only	66.8	31.8	27.6	44.3	61.9	22.7	41.1	4.20	51.1
State-of-the-Art	85.8	53.7	60.7	70.4	76.0	40.4	57.1	12.0	53.5
LXMERT	**88.2**	**54.2**	**63.1**	**72.5**	**77.8**	**45.0**	**60.3**	**42.1**	**76.2**

Table 3.2: Test-set results. VQA/GQA results are reported on the ‘test-standard’ splits and NLVR^2 results are reported on the unreleased test set (‘Test-U’). The highest method results are in bold. Our LXMERT framework outperforms previous (comparable) state-of-the-art methods on all three datasets w.r.t. all metrics.

five datasets to create aligned image-and-sentence pairs. For each image question answering dataset, we take questions as sentences from the image-and-sentence data pairs and take answers as labels in the image QA pre-training task (described in Sec. 3.3.1.3). This provides us with a large aligned vision-and-language dataset of 9.18M image-and-sentence pairs on 180K distinct images. In terms of tokens, the pre-training data contain around 100M words and 6.5M image objects.

3.3.3 Pre-Training Procedure

We pre-train our LXMERT model on the large aggregated dataset (discussed in Sec. 3.3.2) via the pre-training tasks (Sec. 3.3.1). We carefully split each dataset to ensure that all testing images are not involved in any pre-training or fine-tuning steps. Our data splits for each dataset and reproducible code are available at https://github.com/airsplay/lxmert.

LXMERT Pre-Traininig Since MS COCO has a relative large validation set, we sample a set of 5k images from the MS COCO validation set as the mini-validation (minival) set. The rest of the images in training and validation sets (i.e., COCO training images, COCO validation images besides minival, and all the other images in Visual Genome) are used in pre-training. Although the captions and questions of the MS COCO test sets are available, we exclude all of them to make sure that testing images are not seen in pre-training.

Fine-tuning For training and validating VQA v2.0, we take the same split convention as in our LXMERT pre-training. The data related to images in LXMERT mini-validation set is used to validate model performance and the rest of the data in train+val are used in fine-tuning. We test our model on the VQA v2.0 'test-dev' and 'test-standard' splits. For GQA fine-tuning, we follow the suggestions in official GQA guidelines[2] to take *testdev* as our validation set and fine-tune our model on the joint train + validation sets. We test our GQA model on GQA 'test-standard' split. The images in NLVR^2 are not from either MS COCO or Visual Genome, we thus keep using the original split: fine-tune on train split, validate the model choice on val split, and test on the public ('Test-P') and unreleased ('Test-U') test splits. The input sentences are split by the WordPiece tokenizer (Wu et al., 2016) provided in BERT (Devlin et al., 2019b). The objects are detected by Faster R-CNN (Ren et al., 2015) which is pre-trained on Visual Genome (provided by Anderson et al. (2018a)). We do not fine-tune the Faster R-CNN detector and freeze it as a feature extractor. Different from detecting variable numbers of objects in Anderson et al. (2018a), we consistently keep 36 objects for each image to maximize the pre-training compute utilization by avoiding padding. For the model architecture, we set the numbers of layers $\mathbf{N_L}$, $\mathbf{N_X}$, and $\mathbf{N_R}$ to 9, 5, and 5 respectively.[3] More layers are used in the language encoder to balance the visual features extracted from 101-layer Faster R-CNN. The hidden size 768 is the same as $\text{BERT}_{\text{BASE}}$. We pre-train all parameters in encoders and embedding layers from scratch (i.e., model parameters are randomly initialized or set to zero). We also show results of loading pre-trained BERT parameters in Sec. 3.5.1. LXMERT is pre-trained with multiple pre-training tasks and hence multiple losses are involved. We add these losses with equal weights. For the image QA pre-training tasks, we create a joint answer table with 9500 answer candidates which roughly cover 90% questions in all three image QA datasets.

We take Adam (Kingma and Ba, 2014) as the optimizer with a linear-decayed learning-rate schedule (Devlin et al., 2019b) and a peak learning rate at $1e - 4$. We train the model for 20

[2] https://cs.stanford.edu/people/dorarad/gqa/evaluate.html

[3] If we count a single modality layer as one half cross-modality layer, the equivalent number of cross-modality layers is $(9 + 5)/2 + 5 = 12$, which is same as the number of layers in $\text{BERT}_{\text{BASE}}$.

epochs (i.e., roughly 670K[4] optimization steps) with a batch size of 256. We only pre-train with image QA task (see Sec. 3.3.1.3) for the last 10 epochs, because this task converges faster and empirically needs a smaller learning rate. The whole pre-training process takes 10 days on 4 Titan Xp.

Fine-tuning Fine-tuning is fast and robust. We only perform necessary modification to our model with respect to different tasks (details in Sec. 3.4.2). We use a learning rate of $1e-5$ or $5e-5$, a batch size of 32, and fine-tune the model from our pre-trained parameters for 4 epochs.

3.4 Experimental Setup and Results

In this section, we first introduce the datasets that are used to evaluate our LXMERT framework and empirically compare our single-model results with previous best results.

3.4.1 Evaluated Datasets

We use three datasets for evaluating our LXMERT framework.

VQA The goal of visual question answering (VQA) (Antol et al., 2015) is to answer a natural language question related to an image. We take VQA v2.0 dataset (Goyal et al., 2017c) which reduces the answer bias compared to VQA v1.0. The dataset contains an average of 5.4 questions per image and the total amount of questions is 1.1M.

GQA The task of GQA (Hudson and Manning, 2019) is same as VQA (i.e., answer single-image related questions), but GQA requires more reasoning skills (e.g., spatial understanding and multi-step inference). 22M questions in the dataset are generated from ground truth image scene graph to explicitly control the question quality.

NLVR^2 Since the previous two datasets are used in pre-training for increasing the amount of pre-training data to a certain scale, we evaluate our LXMERT framework on another challenging vi-

[4] For comparison, ResNet on ImageNet classification takes 600K steps and BERT takes 1000K steps.

sual reasoning dataset NLVR^2 where all the sentences and images are not covered in pre-training. Each datum in NLVR^2 consists of a two-image pair (img_0, img_1), one statement s, and a ground truth label y^* indicating whether the statement correctly describe the two images. The task is to predict the label y given the images and the statement. To use our LXMERT model on NLVR^2, we concatenate the cross-modality representations of the two images and then build the classifier with GeLU activation(Hendrycks and Gimpel, 2016). Suppose that $\mathrm{LXMERT}(img, sent)$ is the single-vector cross-modality representation, the predicted probability is:

$$\begin{aligned}
x_0 &= \mathrm{LXMERT}(img_0, s) \\
x_1 &= \mathrm{LXMERT}(img_1, s) \\
z^0 &= W_0[x_0; x_1] + b_0 \\
z^1 &= \mathrm{LayerNorm}\left(\mathrm{GeLU}(z^0)\right) \\
prob &= \sigma(W_1 z^1 + b_1)
\end{aligned}$$

where σ is sigmoid function. The model is optimized by maximizing the log-likelihood, which is equivalent to minimize the binary cross entropy loss:

$$\mathcal{L} = \text{-}y^* \log prob - (1 - y^*) \log(1 - prob)$$

3.4.2 Implementation Details

On VQA and GQA, we fine-tune our model from the pre-trained snapshot without data augmentation (analysis in Sec. 3.5.2). When training GQA, we only take raw questions and raw images as inputs and do not use other supervisions (e.g., functional programs and scene graphs). Since each datum in NLVR^2 has two natural images img_0, img_1 and one language statement s, we use LXMERT to encode the two image-statement pairs (img_0, s) and (img_1, s), then train a classifier based on the concatenation of the two cross-modality outputs. More details in Appendix.

3.4.3 Empirical Comparison Results

We compare our single-model results with previous best published results on VQA/GQA test-standard sets and NLVR^2 public test set. Besides previous state-of-the-art (SotA) methods, we also show the human performance and image-only/language-only results when available.

VQA The SotA result is BAN+Counter in Kim et al. (2018), which achieves the best accuracy among other recent works: MFH (Yu et al., 2018), Pythia (Jiang et al., 2018), DFAF (Gao et al., 2019a), and Cycle-Consistency (Shah et al., 2019).[5] LXMERT improves the SotA overall *accuracy* ('Accu' in Table 3.2) by 2.1% and has 2.4% improvement on the 'Binary'/'Other' question sub-categories. Although LXMERT does not explicitly take a counting module as in BAN+Counter, our result on the counting-related questions ('Number') is still equal or better.[6]

GQA The GQA (Hudson and Manning, 2019) SotA result is taken from BAN (Kim et al., 2018) on the public leaderbaord. Our 3.2% *accuracy* gain over the SotA GQA method is higher than VQA, possibly because GQA requires more visual reasoning. Thus our framework, with novel encoders and cross-modality pre-training, is suitable and achieves a 4.6% improvement on open-domain questions ('Open' in Table 3.2).[7]

NLVR^2 NLVR^2 (Suhr et al., 2019) is a challenging visual reasoning dataset where some existing approaches (Hu et al., 2017; Perez et al., 2018) fail, and the SotA method is 'MaxEnt' in Suhr et al. (2019). The failure of existing methods (and our model w/o pre-training in Sec. 3.5.1) indicates that the connection between vision and language may not be end-to-end learned in a complex vision-and-language task without large-scale pre-training. However, with our novel pre-training strategies in building the cross-modality connections, we significantly improve the *accuracy* ('Accu' of 76.2% on unreleased test set 'Test-U', in Table 3.2) by 22%. Another evalua-

[5] These are state-of-the-art methods at the time of our EMNLP May 21, 2019 submission deadline. Since then, there have been some recently updated papers such as MCAN (Yu et al., 2019b), MUAN (Yu et al., 2019a), and MLI (Gao et al., 2019b). MCAN (VQA challenge version) uses stronger mixture of detection features and achieves 72.8% on VQA 2.0 test-standard. MUAN achieves 71.1% (compared to our 72.5%).

[6] Our result on VQA v2.0 'test-dev' is 72.4%.

[7] Our result on GQA 'test-dev' is 60.0%.

Method	VQA	GQA	NLVR^2
LSTM + BUTD	63.1	50.0	52.6
BERT + BUTD	62.8	52.1	51.9
BERT + 1 CrossAtt	64.6	55.5	52.4
BERT + 2 CrossAtt	65.8	56.1	50.9
BERT + 3 CrossAtt	66.4	56.6	50.9
BERT + 4 CrossAtt	66.4	56.0	50.9
BERT + 5 CrossAtt	66.5	56.3	50.9
Train + BERT	65.5	56.2	50.9
Train + scratch	65.1	50.0	50.9
Pre-train + BERT	68.8	58.3	70.1
Pre-train + scratch	**69.9**	**60.0**	**74.9**

Table 3.3: Dev-set accuracy of using BERT.

tion metric *consistency* measures the proportion of unique sentences for which all related image pairs[8] are correctly predicted. Our LXMERT model improves *consistency* ('Cons') to 42.1% (i.e., by 3.5 times).[9]

3.5 Analysis

In this section, we analyze our LXMERT framework by comparing it with some alternative choices or by excluding certain model components/pre-training strategies.

3.5.1 BERT versus LXMERT

BERT (Devlin et al., 2019b) is a pre-trained language encoder which improves several language tasks. As shown in Table 3.3, we discuss several ways to incorporate a BERT$_{\text{BASE}}$ pre-trained model for vision-language tasks and empirically compare it with our LXMERT approach. Although our full model achieves accuracy of 74.9% on NLVR^2, all results without LXMERT pre-training is around 22% absolute lower.

[8] Each statement in NLVR^2 is related to multiple image pairs in order to balance the dataset answer distribution.

[9] These are the unreleased test set ('Test-U') results. On the public test set ('Test-P'), LXMERT achieves 74.5% Accu and 39.7% Cons.

BERT+BUTD Bottom-Up and Top-Down (BUTD) attention (Anderson et al., 2018a) method encodes questions with GRU (Chung et al., 2015), then attends to object RoI features $\{f_j\}$ to predict the answer. We apply BERT to BUTD by replacing its GRU language encoder with BERT. As shown in the first block of Table. 3.3, results of BERT encoder is comparable to LSTM encoder.

BERT+CrossAtt Since BUTD only takes the raw RoI features $\{f_j\}$ without considering the object positions $\{p_j\}$ and object relationships, we enhance BERT+BUTD with our novel position-aware object embedding (in Sec. 3.2.1) and cross-modality layers (in Sec. 3.2.2). As shown in the second block of Table 3.3, the result of 1 cross-modality layer is better than BUTD, while stacking more cross-modality layers further improves it. However, without our cross-modality pre-training (BERT is language-only pre-trained), results become stationary after adding 3 cross-attention layers and have a 3.4% gap to our full LXMERT framework (the last bold row in Table 3.3).

BERT+LXMERT We also try loading BERT parameters[10] into LXMERT, and use it in model training (i.e., without LXMERT pre-training) or in pre-training. We show results in the last block of Table. 3.3. Compared to the 'from scratch' (i.e., model parameters are randomly initialized) approach, BERT improves the fine-tuning results but it shows weaker results than our full model. Empirically, pre-training LXMERT initialized with BERT parameters has lower (i.e., better) pre-training loss for the first 3 pre-training epochs but was then caught up by our 'from scratch' approach. A possible reason is that BERT is already pre-trained with single-modality masked language model, and thus could do well based only on the language modality without considering the connection to the vision modality (as discussed in Sec. 3.3.1.1).

[10] Since our language encoder is same as BERT$_{\text{BASE}}$, except the number of layers (i.e., LXMERT has 9 layers and BERT has 12 layers), we load the top 9 BERT-layer parameters into the LXMERT language encoder.

Method	VQA	GQA	NLVR^2
1. P20 + DA	68.0	58.1	-
2. P20 + FT	68.9	58.2	72.4
3. P10+QA10 + DA	69.1	59.2	-
4. P10+QA10 + FT	**69.9**	**60.0**	**74.9**

Table 3.4: Dev-set accuracy showing the importance of the image-QA pre-training task. P10 means pre-training without the image-QA loss for 10 epochs while QA10 means pre-training with the image-QA loss. DA and FT mean fine-tuning with and without Data Augmentation, resp.

Method	VQA	GQA	NLVR^2
1. No Vision Tasks	66.3	57.1	50.9
2. Feat	69.2	59.5	72.9
3. Label	69.5	59.3	73.5
4. Feat + Label	**69.9**	**60.0**	**74.9**

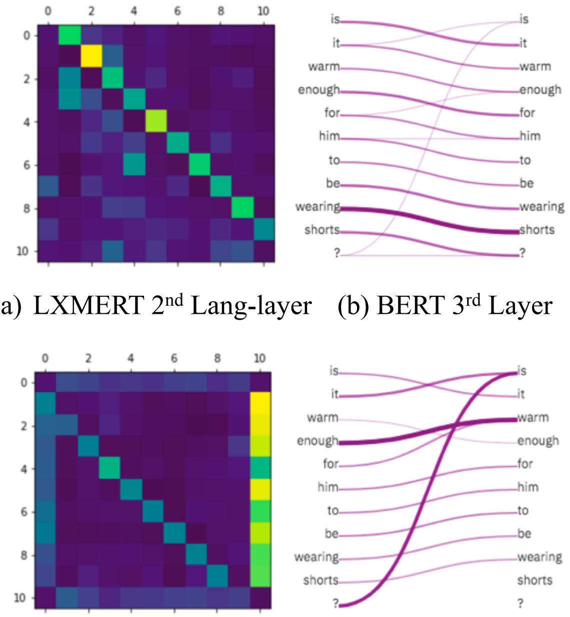
Table 3.5: Dev-set accuracy of different vision pre-training tasks. 'Feat' is RoI-feature regression; 'Label' is detected-label classification.

3.5.2 Effect of the Image QA Pre-training Task

We show the importance of image QA pre-training task (introduced in Sec. 3.3.1.3) by excluding it or comparing it with its alternative: data augmentation.

Pre-training w/ or w/o Image QA To fairly compare with our original pre-training procedure (10 epochs w/o QA + 10 epochs w/ QA, details in Sec. 3.3.3) , we pre-train LXMERT model without image QA task for 20 epochs. As shown in Table 3.4 rows 2 and 4, pre-training with QA loss improves the result on all three datasets. The 2.1% improvement on NLVR^2 shows the stronger representations learned with image-QA pre-training, since all data (images and statements) in NLVR^2 are not used in pre-training.

Pre-training versus Data Augmentation Data augmentation (DA) is a technique which is used in several VQA implementations (Anderson et al., 2018a; Kim et al., 2018; Jiang et al., 2018). It increases the amount of training data by adding questions from other image QA datasets. Our LXMERT framework instead uses multiple QA datasets in pre-training and is fine-tuned only on one specific dataset. Since the overall amounts of data used in pre-training and DA are similar, we thus can fairly compare these two strategies, and results show that our QA pre-training ap-



(a) LXMERT 2nd Lang-layer (b) BERT 3rd Layer

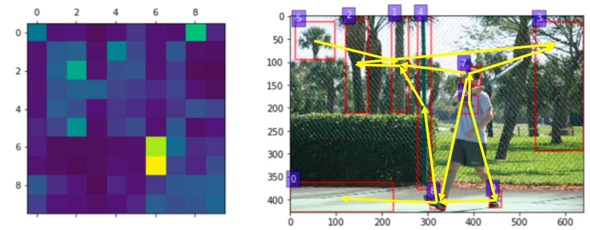
(c) LXMERT 4th Lang-layer (d) BERT 4th Layer

Figure 3.3: Attention graphs reveal similar behavior in the LXMERT language encoder (a, c) and in the original BERT encoder (b, d). Fig. a & b show the attention pointing to next words while Fig. c & d show the attention pointing to previous words.

proach outperforms DA. We first exclude the QA task in our pre-training and show the results of DA fine-tuning. As shown in Table. 3.4 row 1, DA fine-tuning decreases the results compared to non-DA fine-tuning in row 2. Next, we use DA after QA-pre-training (row 3) and DA also drops the results.

3.5.3 Effect of Vision Pre-training tasks

We analyze the effect of different vision pre-training tasks in Table 3.5. Without any vision tasks in pre-training (i.e., only using the language and cross-modality pre-training tasks), the results (row 1 of Table 3.5) are similar to BERT+3 CrossAtt in Table 3.3. The two visual pre-training tasks (i.e., RoI-feature regression and detected-label classification) could get reasonable results (row 2 and row 3) on their own, and jointly pre-training with these two tasks achieves the highest results (row 4).



(a) LXMERT 1st Visn-layer (b) Recovered graphs

Figure 3.4: The attention graph (a) and its recovered scene graph (b) in the first layer of LXMERT's object-relationship encoder.

3.6 Visualizing LXMERT Behavior

In this section, we show the behavior of LXMERT by visualizing its attention graphs in the language encoder, object-relationship encoder, and cross-modality encoder, respectively.

3.6.1 Language Encoder

In Fig. 3.3, we reveal that the LXMERT language encoder has similar behaviour as the original BERT encoder, by using the same sentence "Is it warm enough for him to be wearing shorts?" as the input to both models. LXMERT's attention graphs (in Fig. 3.3(a, c)) are extracted from the pre-trained LXMERT without fine-tuning on a specific task. BERT's attention graphs (in Fig. 3.3(b, d)) come from Hoover et al. (2019).[11] We find that both the second LXMERT layer (Fig. 3.3(a)) and third BERT layer (Fig. 3.3(b)) point to the next words while both the fourth LXMERT layer (Fig. 3.3(c)) and fourth BERT layer (Fig. 3.3(d)) point to the previous words, thus showing the similar behaviour of the two encoders.

3.6.2 Object-Relationship Encoder

In Fig. 3.4, we visualize the attention graph of the first layer in LXMERT's object-relationship encoder. We only highlight the objects with the highest attention scores while the other objects are mostly not attended to. We manually build the connections between objects (marked as yellow lines in Fig. 3.4(b)) according to the attention graph. These connections faithfully draw a

[11] exBERT demo (Hoover et al., 2019) is available at `http://exbert.net/`

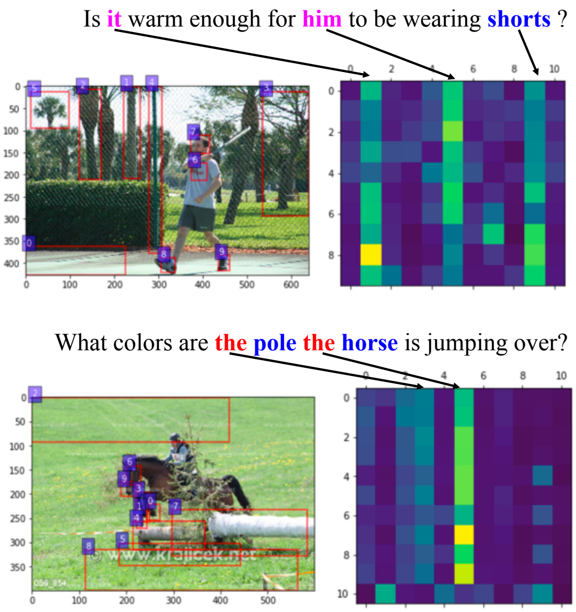


Figure 3.5: Attention graphs in LXMERT's cross-modality encoder showing that the attention focus on pronouns (marked in pink), nouns (marked in blue), and articles (marked in red).

scene graph of the figure, which indicates that the object-relationship encoder might be learning a reasonably good network of the relationships between objects.

3.6.3 Cross-Modality Encoder

In Fig. 3.5, we visualize the attention in LXMERT's cross-modality encoder to reveal the connections between objects and words. We find that the attention focuses on nouns and pronouns as shown in the top figure of Fig. 3.5 because they are the most informative words in current vision-and-language tasks. However, for non-plural nouns (as shown in the bottom example in Fig. 3.5), the attention will focus on the articles. Although we do not specifically design for this behavior, we think that articles are possibly serving as special tokens (e.g., [CLS], [SEP] in BERT), thus providing unified target entries for the attention layers. Next, we are also looking at how to utilize pre-training tasks which directly capture pairwise noun-noun and noun-verb relationships between the images and text sentences.

3.7 Conclusion

We presented a cross-modality framework, LXMERT, for learning the connections between vision and language. We built the model based on Transformer encoders and our novel cross-modality encoder. This model is then pre-trained with diverse pre-training tasks on a large-scale dataset of image-and-sentence pairs. Empirically, we showed state-of-the-art results on two image QA datasets (i.e., VQA and GQA) and show the model generalizability with a 22% improvement on the challenging visual reasoning dataset of NLVR^2. We also showed the effectiveness of several model components and training methods via detailed analysis and ablation studies.

CHAPTER 4: SINGLE-MODALITY PRE-TRAINING FOR MULTI-MODAL TASKS

4.1 Introduction

As vision-and-language tasks involve different modalities as input and output, the vision-and-language models are naturally designed as multiple components where each component focus on some modalities. For example, a typical model for visual questions answering containing three components: the visual encoder, the language encoder, and the fusion module. In general, the amount of aligned vision-and-language data (for both human-annotated and web-collected) is less than the single-modality data (e.g., image-only data or text-only data). To pursue a better vision-and-language model, single-modality pre-trained language modules or vision modules is widely used. The pre-training methods for these single-modality encoder keep evolving and largely contribute to the improvement in vision-and-language tasks. In this chapter, we discuss the history of these single-modality pre-trained models and their impact on vision-and-language tasks.

We first discuss the pre-trained visual encoders. When the encoder-decoder neural architecture was first introduced in Show-and-Tell (Vinyals et al., 2015) and Karpathy and Fei-Fei (2015), it takes the single feature output from the Convolution Neural Networks (CNN) since using single vector aligns well with the LSTM (Hochreiter and Schmidhuber, 1997) decoder structure. Show-Attend-and-Tell (Xu et al., 2015) then introduce the attention mechanism to the image captioning task and thus the 2D feature map from CNN is used. This grid feature approach have being a standard feature extractor setup until Bottom-Up Attention (Anderson et al., 2018a) proposes to use the detected objects as image representations. VinVL (Zhang et al., 2021b) push this detection-based exploration to extreme by using most of publicly available detection datasets. These detection-based approaches hypothesis that the improvement mainly come from the adaptive resolutions in the object-level (i.e., RoI-pooling) features (Ren et al., 2015). However, later works (Liu et al., 2019a; Jiang et al., 2020) show that the actual improvement is attributed to the fine-grained supervision inside the Visual Genome, where not only the classes but also the attributes (e.g., color, material, shape) of objects are annotated. Another useful insight in Bottom-Up Attention is to use higher resolution for the input image (e.g., from 224 to 448). Overall, the success of the visual feature extractors needs to have two characteristics 1) fine-grained visual supervision when training the backbone, 2) large input resolution when extracting the features. The recent work CLIP (Radford et al., 2021a) is trained by contrastive loss on large and diverse image-sentence pairs to enable zero-shot image classifier. It takes the free-form natural language as the visual supervision, and thus its visual encoder coincidentally satisfies these two criterions. In our recent work (Shen et al., 2021), we consider to use CLIP encoder as the next-generation feature extractors. By simply replacing the previous feature extractors to CLIP feature extractor, significant and consistent improvements are observed on diverse downstream tasks.

Like pre-trained vision encoder, pre-trained language encoder are explored as well. The development of pre-trained language encoder is slower than the pre-trained visual encoder. Previous works (Anderson et al., 2018a) use pre-trained word embedding like Word2Vec (Mikolov et al., 2013) and GLoVE (Pennington et al., 2014). ELMo (Peters et al., 2018) shows a success in pre-trained language encoder by using the language modeling task. We test the ability of the ELMo encoder in our work Tan et al. (2019) and find that ELMo helps stabilize the training but do not provide better results. BERT (Devlin et al., 2019a) take the transformer model and train on the clean Wikipedia dataset with masked language modeling. The surprisingly high results produced by BERT excite the community and also encourage the exploration of the usage in vision-and-language tasks. With the BERT integration, PRESS (Li et al., 2019b) shows success in vision-and-language navigation and B2T2 (Alberti et al., 2019) got state-of-the-art results on visual commonsense reasoning. Part of the improvement in using BERT possibly comes from a good initialization for a large language model. Thus, when sufficient large pre-training dataset is provided, LXMERT (Tan and Bansal, 2019) and CLIP (Radford et al., 2021a) train the language encoder from scratch and do not observe a significant performance drop. However, these pre-trained language encoder are still needed in some language-heavy datasets, e.g., VCR (Zellers et al., 2019) and Hateful Meme (Kiela et al., 2020).

In our experiments, we take the vision-and-language navigation as the main task to illustrate the impact of different features. Vision-and-language navigation data are collected from the Matterport-3D environments (Chang et al., 2017) thus it prohibits any potential data overlapping with the single-modality pre-training. For other dataset, it shows an inevitable data collision since most of vision-and-language datasets are collected on popular vision benchmarks, where the image encoder is pre-trained on. VQA (Antol et al., 2015; Goyal et al., 2017b) is collected on MS COCO that shares about 50% images with the Bottom-up Attention (Anderson et al., 2018a) and 100% of training images are seen in VinVL (Zhang et al., 2021b). GQA is auto-generated from Visual Genome annotations, thus visual encoders trained on VG detection benefits from such overlapping semantic annotations. In our paper Shen et al. (2021), we also provide the result for other datasets as well.

4.2 Pre-trained Vision Encoder in Vision-and-Language Navigation

Vision-and-language navigation tests the agent's ability to take action according to human instructions, which recently gains popularity in embodied AI (Anderson et al., 2018b; Chen et al., 2019a; Jain et al., 2019; Chen et al., 2019a; Qi et al., 2020b; Krantz et al., 2020; Nguyen and Daumé III, 2019; Ku et al., 2020). Specifically, the agent is put at a location in the environment (Chang et al., 2017) and asked to reach a target by following the language instructions. Here, we investigate the impact of the visual encoder on this task.

Model Architecture. We experiment with the basic attentive neural agent as in Fried et al. (2018) and Tan et al. (2019). At each time step, the agent attends to the panoramic views and the instruction to make an action. The panoramic view is processed with a pre-trained visual encoder (e.g., ResNet) and the instructions are processed by a language LSTM (Hochreiter and

Schmidhuber, 1997). The agent model (i.e., another LSTM) then attends to the visual features and the language representations to predict the actions. At each time step t, the agent attends to the panoramic views $\{v_{t,i}\}_i$ and the instruction $\{w_j\}$ to make the action. The panoramic view is processed with a pre-trained visual encoder (e.g., ResNet) and the instructions are processed by a language LSTM (Hochreiter and Schmidhuber, 1997), denoted $\mathrm{LSTM}_{\mathrm{L}}$. The agent model, $\mathrm{LSTM}_{\mathrm{A}}$, then attends to the visual features and the language representations to predict the actions.

$$g_{t,i} = \mathrm{ResNet}(v_{t,i}) \tag{4.1}$$

$$x_1, \ldots, x_l = \mathrm{LSTM}_{\mathrm{L}}(w_1, \ldots, w_l) \tag{4.2}$$

$$\mathit{input}_t = [\mathrm{Attn}(h_{t-1}, \{g_{t,i}\}), \mathrm{Attn}(h_{t-1}, \{x_j\})] \tag{4.3}$$

$$h_t, c_t = \mathrm{LSTM}_{\mathrm{A}}(\mathit{input}_t, h_{t-1}, c_{t-1}) \tag{4.4}$$

where h_t and c_t are the hiddens and states of the action LSTM at time step t, respectively.

4.2.1 Impact to the Results

We replace the pre-trained visual encoder from ImageNet pre-trained ResNet to the CLIP visual encoders. We use a single-vector output for the entire image following previous works (Fried et al., 2018). For CLIP-ViT models, we take the output of the [CLS] token. For CLIP-ResNet models, we take the attentive pooled feature (Radford et al., 2021a) of the feature map. These features are also linearly projected and L2-normalized as in the CLIP model.

Experimental Setup. We apply our model to two vision-and-language navigation datasets: Room-to-Room (R2R, Anderson et al. (2018b)) and Room-across-Room (RxR, Ku et al. (2020)). R2R is built on the indoor environments from the MatterPort3D dataset (Chang et al., 2017). The environments are split into training (61 environments), unseen validation (11 environments), and unseen test (18 environments). The agent is trained on the training environments (with 14,025 navigation instructions) and tested on separate sets of environments (2,349 in the unseen-

Method	Unseen Test	
	SR	SPL
No Pre-Training		
R2R (Anderson et al., 2018b)	20	18
RPA (Wang et al., 2018)	25	23
S-Follower (Fried et al., 2018)	35	28
RCM (Wang et al., 2019c)	43	38
SMNA (Ma et al., 2019a)	48	35
Regretful (Ma et al., 2019b)	48	40
FAST-Short (Ke et al., 2019)	54	41
EnvDrop (Tan et al., 2019)	51	47
PRESS (Li et al., 2019b)	49	45
ALTR (Huang et al., 2019)	48	45
CG (Anderson et al., 2019)	33	30
RelGraph (Hong et al., 2020)	55	52
EnvDrop + CLIP-ViL	**59**	**53**
Pre-Training		
AuxRN (Zhu et al., 2020)	55	50
PREVALENT (Hao et al., 2020)	54	51
VLN-BERT(Hong et al., 2021)+OSCAR	57	53
VLN-BERT(Hong et al., 2021)	63	57

Table 4.1: Unseen test results for Room-to-Room (R2R) dataset. ‘SR’ and ‘SPL’ are Success Rate and Success rate normalized by Path Length. ‘Pre-Training’ methods are mostly in-domain pre-trained on the Matterport3D (Chang et al., 2017) environments.

validation and 4,173 in the unseen-test). RxR extends the R2R dataset with multiple languages and follow the environment split. Besides the multilingual nature, RxR is also more diverse in the navigation paths and richer in the present language. For R2R dataset, we follow the hyperparameter (e.g., batch size, learning rate, optimizer) of the publicly available implementation [1] R2R-EnvDrop (Tan et al., 2019) and replace the input features [2] with the CLIP features. To reduce the computational cost, the features are pre-extracted and frozen during the training of the navigational agent. For RxR dataset, we take the processed multilingual data provided in Li et al. (2021a) with Stanza tokenizers (Qi et al., 2020a). Since RxR dataset contains instructions longer than R2R, we change the maximum input length to 160 (from 80) and increase the imitation learning ratio from 0.2 to 0.4 to stabilize the training. Other training hyperparameters of RxR are

[1] https://github.com/airsplay/R2R-EnvDrop

[2] https://github.com/peteanderson80/Matterport3DSimulator

Method	**Unseen Test**	
	SR	**nDTW**
Random-Baseline (Ku et al., 2020)	7.5	15.4
Mono-Baseline (Ku et al., 2020)	25.4	41.1
SAA (Li et al., 2021a)	35.4	46.8
EnvDrop + CLIP-ViL	**38.3**	**51.1**

Table 4.2: Unseen test results for Room-across-Room (RxR) dataset under mono-lingual setup. 'SR' and 'nDTW' are Success Rate and normalized Dynamic Time Warping.

Features	Room-to-Room				Room-across-Room							
	Agent		BT-Agent		English		Hindi		Telugu		Average	
	SR	*SPL*	SR	*SPL*	SR	*nDTW*	SR	*nDTW*	SR	*nDTW*	SR	*nDTW*
ImageNet-Res152	48.2	44.4	53.5	48.8	35.3	50.6	37.9	51.9	37.1	52.0	36.8	51.5
CLIP-Res50	52.6	47.4	56.2	49.7	38.8	53.3	44.1	55.7	43.5	55.5	42.1	54.8
CLIP-ViT-B	52.5	47.7	57.4	51.3	40.2	52.5	44.3	55.0	42.1	54.6	42.2	54.0
CLIP-Res101	53.6	47.5	56.7	49.5	**41.0**	54.6	**44.9**	**56.9**	42.2	55.3	**42.7**	55.6
CLIP-Res50x4	**54.7**	**48.7**	**59.2**	**52.9**	40.8	**54.7**	44.5	56.5	**42.4**	**56.0**	42.6	**55.7**

Table 4.3: Results of Room-to-Room (R2R) and Room-across-Room (RxR) datasets with original ResNet features and CLIP feature variants. 'BT-Agent' is the agent trained with back translation (BT). 'SR' is Success Rate. 'SPL' and 'nDTW' are the main metrics for R2R and RxR, respectively. The best results are bold. CLIP-ViL shows clear improvements over the previous ImageNet-trained ResNet model.

the same as R2R. The models are trained on one RTX 2080 Ti GPU. It takes 1 days to converge in R2R and about 1.5 days to converge in RxR. We report two significant digits for R2R unseen test results following the leaderboard convention.

Experimental Results. We show the test-unseen results of our best model (CLIP-Res50x4) and the comparison to the previous methods. On R2R dataset (in Table 4.1), CLIP-ViL reaches 8% higher in SR (success rate) and 6% higher in SPL (Success Rate normalized by Path Length) than our baseline, EnvDrop. CLIP-ViL outperforms previous non-pre-training agents and shows competitive results to VLN-specific pre-trained models. On RxR dataset (Table 4.2), CLIP-ViL achieves the best success rate and nDTW (normalized Dynamic Time Warping) under the mono-lingual setup (Ku et al., 2020) and is 4.3% better then the previous results for nDTW. In Table 4.3, we compare different CLIP variants with the previous standard ResNet-152 feature extractors. These extractors are pre-trained on ImageNet and use the mean-pooled features as the representation for the image. CLIP-Res50 shows a clear improvement over the IN alternative

Feature	Dimension	SR	**SPL**
ImageNet-Res152	2048	48.2	44.4
CLIP-Res50	1024	52.6	47.4
Grid-Res50	2048	47.6	44.7
Grid-ResX101	2048	46.5	43.2
Grid-ResX152	2048	47.8	44.6

Table 4.4: Comparison between grid features, CLIP features, and ImageNet-trained features on the R2R dataset. 'SR' and 'SPL' are success rate and success rate weighted by path length.

('ImageNet-Res152'). With larger models (i.e., 'CLIP-Res101' and 'CLIP-Res50x4'), the agent performance scales well on both R2R and RxR. Lastly, we find that the CLIP ViT model ('CLIP-ViT-B') has similar results as CLIP-Res50 model. ViT also shows a relatively better result when back translation (BT) is applied. The success of ViT model in VLN is possibly due to the use of [CLS] feature instead of the feature map.

Results Comparison to Grid Features We previously compare the results regarding the ImageNet-pre-trained ResNet-152. We also report the comparison to grid features Jiang et al. (2020) that is trained with detection dataset. Jiang et al. (2020) showed that the results with these features are comparable to the original bottom-up attention with a heavy detection module. We test the performance of these detection-trained grid features on VLN tasks. Specifically, we use the mean pooling of the feature map as the representation of each view following previous works (Anderson et al., 2018b). As shown in Table 4.4, under the same ResNet50 backbone [3], we find that the detection-trained grid features are on par with the classification-trained grid features, still showing a gap to the contrastive-trained grid features. We hypothesize that the grid features inject regional knowledge into the dense feature map thus showing good results with grid-based modules (as in VQA). However, pooling the feature map into a single feature vector (as in previous VLN works) leads to a loss of this dense information.

[3] The CLIP model uses an attention pooling module and makes modifications over the original ResNet (He et al., 2016) backbone.

Method	**Result**		
	Val Seen	**Val Unseen**	**Gap** $\|\Delta\|$
Room-to-Room (Anderson et al., 2018b)			
R2R	38.6	21.8	16.8
RPA	42.9	24.6	18.3
S-Follower	66.4	35.5	30.9
RCM	66.7	42.8	23.9
SMNA	67	45	22
Regretful	69	50	19
EnvDrop	62.1	52.2	9.9
ALTR	55.8	46.1	9.7
RN+Obj	59.2	39.5	19.7
CG	31	31	**0**
Our baseline	56.1	47.5	8.6
Our Learned-Seg	52.6	53.3	**0.7**
Room-for-Room (Jain et al., 2019)			
S-Follower	51.9	23.8	28.1
RCM	55.5	28.6	26.9
Our baseline	54.6	30.7	23.9
Our Learned-Seg	38.0	34.3	**3.7**
CVDN (Thomason et al., 2020)			
NDH	5.92	2.10	3.82
Our baseline	6.60	3.05	3.55
Our Learned-Seg	5.82	4.41	**1.41**

Table 4.5: Results showing the performance gaps between seen ('Val Seen') and unseen ('Val Unseen') environments in several VLN tasks. Room-to-Room and Room-for-Room are evaluated with 'Success Rate', CVDN is evaluated with 'Goal Progress', Touchdown is evaluated with 'Task Completion'.

4.2.2 Impact to the Environmental Bias

For vision-and-language tasks, most of the works (Anderson et al., 2018b; Wang et al., 2018; Fried et al., 2018; Wang et al., 2019c; Ma et al., 2019a,b; Tan et al., 2019; Huang et al., 2019; Hu et al., 2019) observe a significant performance drop from the environments used in training (seen environments) to the ones not used in training (unseen environments), which indicates the models are strongly biased to the seen environments. We here show that a more semantical feature could relieve this bias.

In order to evaluate the generalizability of agent models, indoor VLN datasets (e.g., those collected from Matterport3D (Chang et al., 2017)) use disjoint sets of environments in training and testing. Two validation splits are provided as well: validation seen (which takes the data from training environments) and validation unseen (whose data is taken from testing environments different from the training). In the first part of Table 4.5, we list most of the previous works (R2R (Anderson et al., 2018b), RPA (Wang et al., 2018), S-Follower (Fried et al., 2018), RCM (Wang et al., 2019c), SMNA (Ma et al., 2019a), Regretful (Ma et al., 2019b), EnvDrop (Tan et al., 2019), ALTR (Huang et al., 2019), RN+Obj (Hu et al., 2019), CG (Anderson et al., 2019)) on the Room-to-Room dataset (Anderson et al., 2018b) and their *success rate* under greedy decoding (i.e., without beam-search) on validation seen and validation unseen splits. The large absolute gaps (from 30.9% to 9.7%) between the results of seen and unseen environments show that current agent models on R2R suffer from environment bias.[4] This phenomenon is also revealed in two other newly-released indoor navigation datasets, Room-for-Room (R4R) (Jain et al., 2019) and Cooperative Vision-and-Dialog Navigation (CVDN) (Thomason et al., 2020). The significant result drops from seen to unseen environments can also be observed (i.e., 26.9% on R4R and 3.82 on CVDN), as shown in the second and third parts of Table 4.5. Lastly, we show the results ('Our Learned-Seg' in Table 4.5) when the environment bias is effectively reduced by our learned semantic-segmentation features, compared to our baselines (denoted as 'Our baseline') and previous works.

4.3 Conclusion

We analyzed the impact of different pre-trained vision modules to vision-and-language tasks. With the same model, the features not only largely affect the performance but also change the characteristics of model behavior. We conducted detailed experiments on diverse vision-and-language navigation, visual questions answering, and image captioning tasks. In general, the

[4] Our work's aim is to both close the seen-unseen gap while also achieving competitive unseen results. Note that (Anderson et al., 2019) also achieve 0% gap but at the trade-off of low unseen results.

features containing more semantic information is better for vision-and-language tasks, and we found that the recent CLIP (Radford et al., 2021a) features perform the best. These results call for a study to further improve the single-modality pre-training scheme.

CHAPTER 5: LANGUAGE PRE-TRAINING FROM VISUAL SUPERVISION

5.1 Introduction

Most humans learn language understanding from multiple modalities rather than only from the text and audio, especially using the visual modality. As claimed in Bloom (2002), visual pointing is an essential step for most children to learn meanings of words. However, existing language pre-training frameworks are driven by contextual learning which only takes the language context as self-supervision. For example, word2vec (Mikolov et al., 2013) takes surrounding bag-of-words; ELMo (Peters et al., 2018) and GPT (Radford et al., 2018) take succeeding contexts; and BERT (Devlin et al., 2019a) takes randomly masked tokens. Although these self-supervised frameworks have achieved strong progress towards understanding human language, they did not borrow grounding information from the external visual world.

In this paper, we introduce the visually-supervised language model that simulates human language learning with visual pointing (Bloom, 2002). As shown in Fig. 5.1, this model takes language tokens as input and uses token-related images as visual supervision. We name these images as *vokens* (i.e., visualized tokens), since they act as visualizations of the corresponding tokens. Assuming that a large aligned token-voken dataset exists, the model could learn from these vokens via voken-prediction tasks.

Unfortunately, such an aligned token-voken dataset is currently unavailable and hence there are two main challenges in creating it from visually-grounded language datasets. First, there is a large discrepancy between visually-grounded language (which provides innate visual grounding supervision) and other types of natural language. For example, about 120M tokens are available in visually-grounded language datasets (Tan and Bansal, 2019; Chen et al., 2019b), which is far less compared to the 3,300M tokens in BERT training data and 220B tokens in T5 (Raffel et al.,

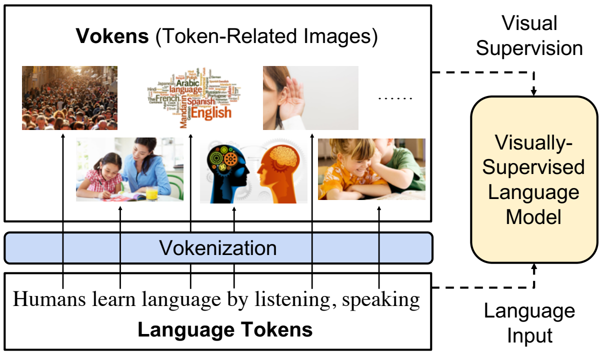

Vokens (Token-Related Images)
Visual Supervision
Visually-Supervised Language Model
Vokenization
Humans learn language by listening, speaking
Language Tokens
Language Input

Figure 5.1: We visually supervise the language model with token-related images. We call these images vokens (visualized tokens) and develop a vokenization process to contextually generate them.

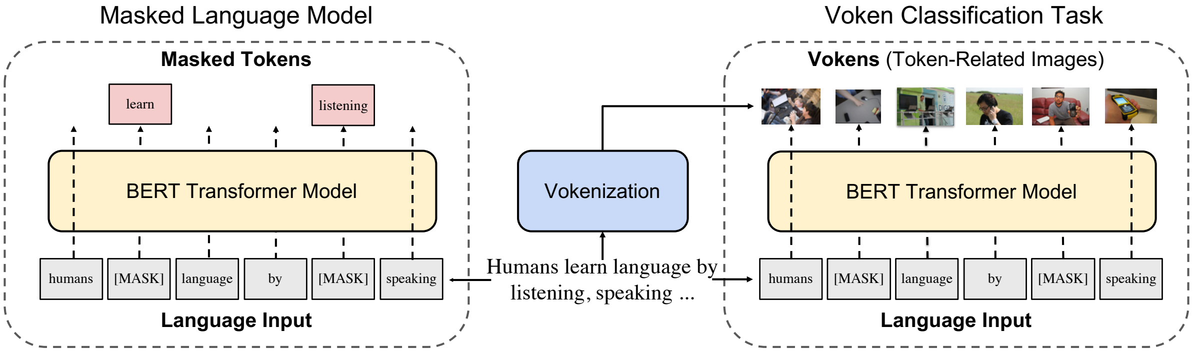


Figure 5.2: Illustration of the BERT transformer model trained with a visually-supervised language model with two objectives: masked language model (on the left) and voken classification (on the right). The first objective (used in original BERT pre-training) predicts the masked tokens as self-supervision while the second objective predicts the corresponding vokens (contextually generated by our vokenization process) as external visual supervision. Since the inputs are the same, we optimize the two objectives simultaneously and share the model weights.

2019). Grounded language also prefers short and instructive descriptions, and thus has different distributions of sentence lengths and active words to other language types. Second, most of the words in natural language are not visually grounded, hence this challenges the premise in creating visual supervision. With an approximate estimation, the ratio of grounded tokens is only about 28% in English Wikipedia. This low grounding ratio leads to low coverage of visual supervision in previous approaches (Frome et al., 2013; Kiela et al., 2018).

To resolve the above two challenges, we propose our *vokenization* method (as shown in Fig. 5.1) that contextually maps the tokens to the visualized tokens (i.e., vokens) by retrieval. Instead of directly supervising the language model with visually grounded language datasets

Dataset	# of Tokens	# of Sents	Vocab. Size	Tokens #/ Sent.	1-Gram JSD	2-Gram JSD	Grounding Ratio
MS COCO	7.0M	0.6M	9K	11.8	0.15	0.27	54.8%
VG	29.2M	5.3M	13K	5.5	0.16	0.28	57.6%
CC	29.9M	2.8M	17K	10.7	0.09	0.20	41.7%
Wiki103	111M	4.2M	29K	26.5	0.01	0.05	26.6%
Eng Wiki	2889M	120M	29K	24.1	0.00	0.00	27.7%
CNN/DM	294M	10.9M	28K	26.9	0.04	0.10	28.3%

Table 5.1: Statistics of image-captioning dataset and other natural language corpora. VG, CC, Eng Wiki, and CNN/DM denote Visual Genome, Conceptual Captions, English Wikipedia, and CNN/Daily Mail, respectively. JSD represents Jensen–Shannon divergence to the English Wikipedia corpus. A large discrepancy exists between the visually grounded captioning and general language corpora.

(e.g., MS COCO (Lin et al., 2014)), we use these relative small datasets to train the vokenization processor (i.e., the *vokenizer*). We then generate vokens for large language corpora (e.g., English Wikipedia), and our visually-supervised language model will take the input supervision from these large datasets, thus bridging the gap between different data sources, which solves the first challenge. The second challenge of low grounding ratio seems to be an inherent characteristic of language; however, we observe that some non-visually-grounded tokens can be effectively mapped to related images when considering its context, e.g., the abstract word "angry" in the sentence "an angry cat lies on my leg". This observation is realized by our *contextual* token-image matching model (defined in Sec. 5.3.2) inside our vokenization processor, where we map tokens to images by viewing the sentence as the context.

Using our proposed vokenizer with a contextualized token-image matching model, we generate vokens for English Wikipedia. Supervised by these generated vokens, we show consistent improvements upon a BERT model on several diverse NLP tasks such as GLUE (Wang et al., 2019a), SQuAD (Rajpurkar et al., 2016), and SWAG (Zellers et al., 2018). We also show the transferability of our vokens to other frameworks (i.e., RoBERTa).

5.2 Visually-Supervised Language Models

Contextual language representation learning is driven by self-supervision without considering explicit connections (grounding) to the external world. In this section, we illustrate the idea of a visually-supervised language model and discuss the challenges of creating its visual supervision.

5.2.1 Vokens: Visualized Tokens

To provide visual supervision to the language model, we assume a text corpus where each token is aligned with a related image (although these voken annotations currently do not exist, we will try to generate vokens next in Sec. 5.3 by the vokenization process). Hence, these images could be considered as visualizations of tokens and we name them as 'vokens'. Based on these vokens, we propose a new pre-training task for language: voken classification.

5.2.2 The Voken-Classification Task

Most language backbone models (e.g., ELMo (Peters et al., 2018), GPT (Radford et al., 2018), BERT (Devlin et al., 2019a)) output a localized feature representation $\{\boldsymbol{h}_i\}$ for each token in a sentence $s = \{w_i\}$. Thus it allows adding a token-level classification task without modifying the model architecture. Suppose the vokens come from a finite set $\mathbb{X}$, we convert the hidden output $\boldsymbol{h}_i$ to a probability distribution p_i with a linear layer and a softmax layer, then the voken classification loss is the negative log probability of all corresponding vokens:

$$\begin{aligned}
\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_l &= \mathrm{lm}(w_1, w_2, \ldots, w_l) \\
p_i(v \mid s) &= \mathrm{softmax}_v\{W\, \boldsymbol{h}_i + b\} \\
\mathcal{L}_{\textsc{voken-cls}}(s) &= -\sum_{i=1}^{l} \log p_i \left(v(w_i; s) \mid s \right)
\end{aligned}$$

This task could be easily integrated into current language pre-training frameworks, and we next show an example.

Example: Visually-Supervised BERT Fig. 5.2 shows an example realization of the voken-classification task that provides visual supervision to BERT (Devlin et al., 2019a). The original BERT pre-training mainly relies on the task of masked language model[1] (illustrated on the left side of Fig. 5.2): tokens are randomly masked and the model needs to predict these missing tokens from language context. For simplicity, we use s and $\hat{s}$ to denote the set of tokens and masked tokens, separately. The unmasked tokens are the set difference $s \backslash \hat{s}$. Suppose q_i is the conditional probability distribution of the i-th token, the Masked Language Model (MLM) loss is the negative log-likelihood of the masked tokens:

$$\mathcal{L}_{\text{MLM}}(s, \hat{s}) = -\sum_{w_i \in \hat{s}} \log q_i \left(w_i \mid s \backslash \hat{s} \right)$$

Without changing the model and model's inputs, we calculate the voken-classification loss for all tokens (illustrated on the right side of Fig. 5.2):

$$\mathcal{L}_{\text{VOKEN-CLS}}(s, \hat{s}) = -\sum_{w_i \in s} \log p_i \left(v(w_i; s) \mid s \backslash \hat{s} \right)$$

The visually-supervised masked language model takes the sum of these two losses with a ratio λ.

$$\mathcal{L}_{\text{VLM}}(s, \hat{s}) = \mathcal{L}_{\text{VOKEN-CLS}}(s, \hat{s}) + \lambda \mathcal{L}_{\text{MLM}}(s, \hat{s}) \tag{5.1}$$

5.2.3 Two Challenges in Creating Vokens

Previous sections illustrate the potential external supervision by assuming the existence of vokens. However, we are currently lacking the dense annotations from tokens to images. The most similar concept to vokens is phrase localization (e.g., in Flickr30K entities (Young et al., 2014; Plummer et al., 2017)). Because the process of collecting phrase localization is costly, the

[1] The next-sentence prediction task is removed in RoBERTa (Liu et al., 2019b) and XLM (Lample and Conneau, 2019) and the fine-tuning results are not largely affected.

coverage and the amount of annotations cannot meet our requirements.[2] Apart from phrase localization, the most promising data source is image captioning datasets with sentence-to-image mappings. Image captions belong to a specific type of language called *grounded language* (Roy and Pentland, 2002; Hermann et al., 2017), which has an explicit grounding to external existence or physical actions. However, grounded language has a large discrepancy to other types of natural language (e.g., News, Wiki, and Textbooks). To illustrate this, we list key statistics of three image-captioning dataset (i.e., MS COCO (Lin et al., 2014), Visual Genome (Krishna et al., 2017), and Conceptual Captions (Sharma et al., 2018)) and three language corpora of other language types (i.e., Wiki103 (Merity et al., 2017), English Wiki, and CNN/Daily Mail (See et al., 2017)) in Table 5.1. This discrepancy between grounded language and other types of natural language leads to two challenges:

A. Different Distributions between Grounded Language and Other Natural Language Corpora. Sentences belonging to grounded language are usually short and informative, e.g., the average sentence length in MS COCO is 11.8, which is much shorter than the average sentence length of 24.1 in English Wiki. The vocabulary[3] of MS COCO only covers around one-third of token types (Smith, 2019) in English Wiki. There is also a large divergence of the 1-Gram and 2-Gram distributions (measured by Jensen–Shannon divergence) between grounded language dataset and the English Wikipedia. Lastly, the amount of tokens in grounded language corpora are also orders of magnitude smaller than commonly-used Wikipedia.

B. Low Grounding Ratio in Natural Language. The grounding ratio is defined as the percentage of visually grounded tokens in the dataset. Visually grounded tokens (e.g., concrete nouns) are the token types that are naturally related to specific visual contents (e.g., 'cat', 'cake', 'clock'). Since a precise list of such token types is hard to define, we thus estimate the grounding ratio based on existing grounded language corpora. Specifically, we consider a token type with

[2] Recently, a concurrent work Pont-Tuset et al. (2019) releases localized narratives. The tokens are aligned with image pixels instead of images.

[3] The vocabulary is calculated following Karpathy and Fei-Fei (2015) where the words with $> \quad 5$ occurrence is counted.

more than 100 occurrences in MS COCO (after removing all stop words) as visually-grounded. As shown in the last column of Table 5.1, the grounding ratio of English Wiki is 27.7%, which is almost half of that in Visual Genome.

To address these two challenges, we propose a vokenizer with contextual token-image matching models next in Sec. 5.3.

5.3 Vokenization

In the previous section, we discuss the potential of using vokens (i.e., visualized tokens) as visual supervision to the language model, and also demonstrate the large gap between currently available resources (i.e., annotated dataset) and the desired requirements. Hence, in this section, we develop a framework that can generate vokens. As shown in Fig. 5.2, the general idea is that we learn a "vokenizer" from image-captioning dataset and use it to annotate large language corpora (i.e., English Wiki), thus bridging the gap between grounded language and other types of natural language. We start by illustrating the vokenization process and then describe how we implement it.

5.3.1 The Vokenization Process

As shown in Fig. 5.1 and Fig. 5.2, vokenization is the process to assign each token w_i in a sentence $s = (w_1, w_2, \dots, w_l)$ with a relevant image $v(w_i; s)$. We call this image $v(w_i; s)$ as a 'voken' (visualized token). Instead of creating this image with generative models, we retrieve an image from a set of images $\mathbb{X} = \{x_1, x_2, \dots, x_n\}$ regarding a token-image-relevance scoring function $r_\theta(w_i, x; s)$. This scoring function $r_\theta(w_i, x; s)$, parameterized by θ, measures the relevance between the token w_i in the sentence s and the image x. We here assume that the optimal parameter of this function is θ^* and will discuss the details of formulations later. The voken $v(w_i; s)$ related to a token w_i in the sentence s is realized as the image $x \in \mathbb{X}$ that maximizes their relevance score r_{θ^*}:

$$v(w_i; s) = \arg\max\nolimits_{x \in \mathbb{V}} \; r_{\theta^*}(w_i, x; s)$$

Since the image set $\mathbb{X}$ indeed builds a finite vocabulary for vokens, we could utilize the voken-classification task (formulated in Sec. 5.2.2) to visually supervise the language model training. We next talk about the detailed implementation of this vokenization process.

5.3.2 Contextual Token-Image Matching Model

Lying in the core of the vokenization process is a contextual token-image matching model. The model takes a sentence s and an image x as input, and the sentence s is composed of a sequence of tokens $\{w_1, w_2, \ldots, w_l\}$. The output $r_\theta(w_i, x; s)$ is the relevance score between the token $w_i \in s$ and the image x while considering the whole sentence s as a context.

Modeling To model the relevance score function $r_\theta(w_i, x; s)$, we factorize it as an inner product of the language feature representation $\boldsymbol{f}_\theta(w_i; s)$ and the visual feature representation $\boldsymbol{g}_\theta(x)$:

$$r_\theta(w_i, x; s) = \boldsymbol{f}_\theta(w_i; s)^\intercal \boldsymbol{g}_\theta(x)$$

These two feature representations are generated by language and visual encoders respectively. The language encoder first uses a pre-trained BERT$_{\text{BASE}}$ (Devlin et al., 2019a) model to contextually embed the discrete tokens $\{w_i\}$ into hidden-output vectors $\{\boldsymbol{h}_i\}$:

$$\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_l = \mathit{bert}(w_1, w_2, \ldots, w_l)$$

Then we apply a multi-layer perceptron (MLP) w_mlp_θ to down project the hidden output $\boldsymbol{h}_i$. In order to simplify the retrieval process in Sec. 5.3.1, the final language features are normalized to norm-1 vectors by dividing their Euclidean norms:

$$\boldsymbol{f}_\theta(w_i; s) = \frac{w\_mlp_\theta(\boldsymbol{h}_i)}{\|w\_mlp_\theta(\boldsymbol{h}_i)\|}$$

On the other side, the visual encoder first extracts the visual embedding e from a pre-trained ResNeXt (Xie et al., 2017). Similar to the language encoder, an MLP layer x_mlp_θ and an L2-normalization layer are applied subsequently:

$$\begin{aligned}\boldsymbol{e} &= \mathrm{ResNeXt}(x)\\ \boldsymbol{g}_\theta(x) &= \frac{x\_mlp_\theta(\boldsymbol{e})}{\|x\_mlp_\theta(\boldsymbol{e})\|}\end{aligned}$$

Training Since the dense annotations from tokens to images are lacking and hard to generate (illustrated in Sec. 5.2.3), we thus alternatively train the token-image matching model from weak supervision in image-captioning datasets (e.g., MS COCO (Lin et al., 2014)). These datasets are comprised of sentence-image pairs $\{(s_k, x_k)\}$ where the sentence s_k describes the visual content in image x_k. To build alignments between tokens and images, we pair all tokens in a sentence s_k with the image x_k. The model is then optimized by maximizing the relevance score of these aligned token-image pairs over unaligned pairs.

Without loss of generality, assuming (s, x) is an image-captioning data point, we randomly sample another image x' with the condition $x' \neq x$. We then use hinge loss to optimize the weight θ so that the score of the positive token-image pair $r_\theta(w_i, x; s)$ aims to be larger than the negative pair $r_\theta(w_i, x'; s)$ by at least a margin M.

$$\begin{aligned}\mathcal{L}_\theta(s, x, x') = \sum_{i=1}^{l} \max\{0, M - r_\theta(w_i, x; s)\\ + r_\theta(w_i, x'; s)\}\end{aligned}$$

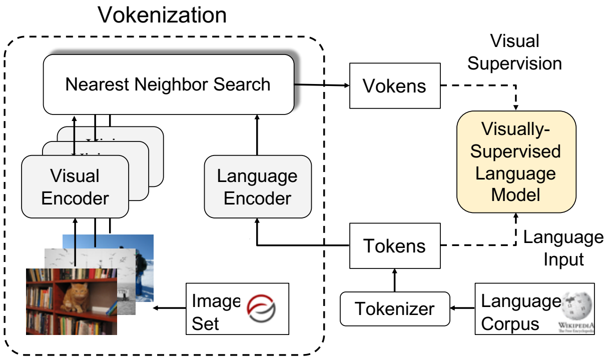


Figure 5.3: Implementation of our vokenization process. For the tokens in language corpora, we contextually retrieved images (with nearest neighbor search) from the image set as vokens. These generated vokens are then used as the visual supervision to the language model.

Intuitively, minimizing this hinge loss $\max\{0, M - pos + neg\}$ will try to increase the score of the positive pair and decrease the score of the negative pair when the score difference is smaller than the margin M. Otherwise (if the difference is $\geqslant$ margin M), the two scores remain unchanged.

Inference Given that the relevance score is factorized as the inner product of feature representations $\boldsymbol{f}_\theta(w_i; s)$ and $\boldsymbol{g}_\theta(v)$, the retrieval problem in Sec. 5.3.1 could be formulated as Maximum Inner Product Search (Mussmann and Ermon, 2016)). Moreover, since the vectors are norm-1, the vector with the maximum inner product is identical to the closest vector in the Euclidean space (i.e., Nearest Neighbor (Knuth, 1973)). We illustrate the detailed implementation in Fig. 5.3.

5.3.3 Revokenization

A constraint of the vokenization process in Sec. 5.3.1 is that the vokens depend on the actual tokenizer of the language encoder in Sec. 5.3.2. Since different frameworks utilize a various range of tokenizers, this constraint limits the transferability of vokens between different frameworks. Instead of binding our vokenizer to a specific pre-training framework (e.g., BERT), we want to enable its extensibility to other frameworks (e.g., RoBERTa). Thus, we introduce a "revokenization" technique to address this limitation.

Given two different tokenizers T_1 and T_2, they tokenize a sentence s into two different sequences of tokens: $T_1(s) = (w_1, w_2, \ldots, w_l)$ and $T_2(s) = (u_1, u_2, \ldots, u_m)$.

Method	SST-2	QNLI	QQP	MNLI	SQuAD v1.1	SQuAD v2.0	SWAG	Avg.
BERT$_{6L/512H}$	88.0	85.2	87.1	77.9	71.3/80.2	57.2/60.8	56.2	75.6
BERT$_{6L/512H}$ + Voken-cls	89.7	85.0	87.3	78.6	71.5/80.2	61.3/64.6	58.2	76.8
BERT$_{12L/768H}$	89.3	87.9	83.2	79.4	77.0/85.3	67.7/71.1	65.7	79.4
BERT$_{12L/768H}$ + Voken-cls	**92.2**	**88.6**	**88.6**	**82.6**	**78.8/86.7**	68.1/71.2	**70.6**	**82.1**
RoBERTa$_{6L/512H}$	87.8	82.4	85.2	73.1	50.9/61.9	49.6/52.7	55.1	70.2
RoBERTa$_{6L/512H}$ + Voken-cls	87.8	85.1	85.3	76.5	55.0/66.4	50.9/54.1	60.0	72.6
RoBERTa$_{12L/768H}$	89.2	87.5	86.2	79.0	70.2/79.9	59.2/63.1	65.2	77.6
RoBERTa$_{12L/768H}$ + Voken-cls	**90.5**	**89.2**	**87.8**	**81.0**	**73.0/82.5**	**65.9/69.3**	**70.4**	**80.6**

Table 5.2: Fine-tuning results of different pre-trained models w/ or w/o the voken classification task (denoted as "Voken-cls"). SQuAD results are "exact match"/"F1". The results which significantly outperform the second-best ones are marked in bold. The averages of metrics (denoted as "Avg.") show improvement from voken supervisions.

Without loss of generality, assuming the vokenizer is built based on the first tokenizer T_1, the standard vokenization process will generate a sequence of vokens $\{v(w_i; s)\}_{i=1}^{l}$ which are one-to-one aligned with the tokens $\{w_i\}_{i=1}^{l}$. Our goal is to transfer these w-related vokens to the u-related vokens generated by T_2. We adapt the idea of "nearest neighbor algorithm" (Altman, 1992) here. For a given token u_j, among all w's, we select the one that overlaps the most with u_j and record it as $w_{ind(j)}$. The voken for u_j is defined as the voken for its "nearest neighbor" $w_{ind(j)}$:

$$v(u_j; s) := v(w_{ind(j)}; s)$$

$$ind(j) = \arg\max\nolimits_{i=1}^{l} \text{overlap}(w_i, u_j)$$

The overlapping of two tokens are further quantified by the intersection-over-union (i.e., Jaccard index, defined as $\text{IoU}(A, B)$=$\frac{|A \cap B|}{|A \cup B|}$) of their ranges in the raw sentence s.

5.4 Experimental Setups and Results

5.4.1 Pre-training Data and Fine-tuning Tasks

We train our model on English Wikipedia [4] and its featured subset Wiki103 (Merity et al., 2017). We use our vokenizer to generate vokens for these two datasets as well. The pre-trained models are then fine-tuned on GLUE (Wang et al., 2019a), SQuAD (Rajpurkar et al., 2016, 2018), and SWAG (Zellers et al., 2018) to assess the pre-training performance. Since some smaller tasks in GLUE are reported as unstable (Dodge et al., 2020), recent papers (e.g., Li et al. (2020c)) only report on selected tasks. We follow this trend and evaluate on the four largest datasets (i.e., SST-2 (Socher et al., 2013), QNLI (Rajpurkar et al., 2016), QQP (Iyer et al., 2017), MNLI (Williams et al., 2018)).[5].

5.4.2 Implementation Details

We train our contextual token-image matching model (in Sec. 5.3.2) on MS COCO image captioning dataset for 20 epochs. The concatenation of the last 4 layers of BERT outputs and $\mathrm{ResNeXt\text{-}101\text{-}32x8d}$ features are used as language hidden states and visual embedding, respectively. Both multi-layer perceptrons w_mlp_θ and x_mlp_θ have two fully-connected layers with 256-dimensional intermediate outputs (followed by ReLU activation) and 64-dimensional final outputs. The two backbone models BERT (Devlin et al., 2019a) and ResNeXt (Xie et al., 2017) are not fine-tuned. We set the hinge loss margin M to 0.5. During the vokenization process of English Wikipedia and Wiki103, we use the faiss (Johnson et al., 2019) library to speed up the nearest neighbor search. The vokens are retrieved from the Visual Genome images that are not used in MS COCO. We fix a voken size of 50000.

When pre-training the model on pure language corpus, we unify the training protocols to avoid possible side effects. We follow previous works to conduct two simplifications: 1. Remov-

[4] BERT (Devlin et al., 2019a) also uses Toronto Books Corpus (Zhu et al., 2015). However, the dataset is not publicly released. We thus exclude it in our study to ensure reproducibility.

[5] The size of the used four dataset range from 60K to 400 while the omitted dataset range from 0.6K to 8.5K.

Model	Init. with BERT?	Diff. to BERT Weight	SST-2	QNLI	QQP	MNLI
ViLBERT (Lu et al., 2019)	Yes	0.0e-3	90.3	89.6	88.4	82.4
VL-BERT (Su et al., 2020)	Yes	6.4e-3	90.1	89.5	88.6	82.9
VisualBERT (Li et al., 2019a)	Yes	6.5e-3	90.3	88.9	88.4	82.4
Oscar (Li et al., 2020b)	Yes	41.6e-3	87.3	50.5	86.6	77.3
LXMERT (Tan and Bansal, 2019)	No	42.0e-3	82.4	50.5	79.8	31.8
$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019a)	-	0.0e-3	90.3	89.6	88.4	82.4
$\text{BERT}_{\text{BASE}}$ + Weight Noise	-	6.5e-3	89.9	89.9	88.4	82.3

Table 5.3: Results of vision-and-language pre-trained models on GLUE tasks. We also provide BERT models w/ and w/o weight noise as baselines.

Pre-trained on	SST-2	QNLI	QQP	MNLI
MS COCO	83.7	60.6	82.1	69.3
Wiki103*	85.8	77.9	84.8	73.9
No Pre-train	77.1	50.5	31.6	31.8

Table 5.4: Results of BERT models pre-trained on captions in MS COCO and a reduced version of Wiki103 dataset (denoted as Wiki103*). Models without pre-training are taken as a baseline.

ing the next-sentence-prediction task (Liu et al., 2019b) 2. Using fixed sequence length (Conneau et al., 2020) of 128. We take the 12-layer $\text{BERT}_{\text{BASE}}$ model of 768 hidden dimensions and train it on English Wikipedia for 200K steps from scratch. We also take a reduced 6-layer model and train it on Wiki103 for 40 epochs (160K steps) because this reduced model could not fit the full English Wikipedia dataset.

Since we only use the vokens in the supervision, the voken-classification task does not bring additional parameters to the language model but needs more computations. We thus adjust the training steps for pure masked-language-model (MLM) training accordingly for a fair comparison. The loss ratio λ=1.0 in Eqn. 5.1 is not tuned because of limited budget. All pre-training processes take batch sizes of 256 and learning rates of $2e$-4. For fine-tuning tasks, we report the results on the validation sets. We train 3 epochs with a learning rate of $1e$-4 and a batch-size of 32 for all tasks in GLUE. The hyper-parameters for SQuAD, SWAG are borrowed from BERT.

5.4.3 Results

As reported in Table 5.2, we fine-tune the pre-trained models on different natural-language tasks. The models are either pre-trained with masked language model (e.g., "$\text{BERT}_{\text{6L/512H}}$")

Method	Retrieval	Supervision	SST-2	QNLI	QQP	MNLI
SentLabel	Sent-level	Sent-level	88.3	86.1	86.9	78.0
Propagated	Sent-level	Token-level	88.9	87.9	88.1	80.2
Term Frequency	Token-level	Token-level	89.0	86.9	85.5	79.8
Vokens	Contextual Token-level	Token-level	92.2	88.6	88.6	82.6

Table 5.5: Comparisons of sentence-level (denoted as "Sent-level") and token-level approaches. Token-level approaches outperform the sentence-level approaches from both retrieval-method and supervision perspective.

or pre-trained with masked language model with an additional voken-classification task (e.g., "$\text{BERT}_{6L/512H}$+Voken-cls") following Eqn. 5.1. The default metric is accuracy. Following Wang et al. (2019a), we report the average of F1 and accuracy for QQP. For SQuAD, we report the exact matching and F1 score respectively. We also compute macro-averages for evaluated tasks (denoted as "Avg." in the last column) as a general indicator. Although the different architectures of models (i.e., 6L/512H and 12L/768H) affect the fine-tuning results, the voken-classification task consistently improves the downstream tasks' performance and achieves large average gains. We also show the transferability of our vokenizer to the RoBERTa model and observe the same phenomenon as that in BERT.

5.5 Analysis

5.5.1 Limit of Visually-Grounded Language

In Sec. 5.2.3, we illustrated the differences between (visually-)grounded-language datasets and other natural-language corpora by demonstrating their contrasting statistics. In this section, we study the models trained with grounded language and show their ineffectiveness on pure-language tasks. We first investigate vision-and-language pre-training frameworks, which succeed on multimodal tasks. As shown in Table 5.3, when fine-tuning them on pure-language tasks, the results are generally lower than the pre-trained BERT model.[6] Although these frameworks are dif-

[6] ViLBERT (Lu et al., 2019) freezes the BERT weight in its training thus their results are the same to BERT; Uniter (Chen et al., 2019b) shrinks its vocab thus is not shown.

ferent in multiple ways, the only remarkable factor to the fine-tuning results is the BERT-weight initialization. Moreover, we also show that these models are similar to a BERT model with a random weight noise of the same magnitude. We thus claim that vision-and-language pre-training on visually-grounded language dataset currently might not help the pure-language tasks. Note that the BERT results in Table 5.2 are not fairly comparable to the results in Table 5.3 because the original BERT model (Devlin et al., 2019a) also uses Toronto Books Corpus (Zhu et al., 2015). Unfortunately, this dataset is not publicly available and hence we exclude it. According to Raffel et al. (2019), the exclusion of Toronto Books Corpus downgrades the results and we observe the same tendency here (comparing $\text{BERT}_{\text{12L/768H}}$ in Table 5.2 and $\text{BERT}_{\text{BASE}}$ in Table 5.3).

Besides these existing models, we next investigate the BERT models trained with masked language model on grounded language data (i.e., MS COCO). A control experiment is built by shrinking the Wiki103 to the same token amount as MS COCO. We also provide the BERT model trained from scratch as a baseline. As shown in Table 5.4, the model trained with MS COCO is significantly worse than the model trained with Wiki103 on all downstream tasks. The reason might be the large discrepancy between visually-grounded language and other types of language as shown in Sec. 5.2.3.

5.5.2 Token-Level vs. Sentence-Level Approaches

In Sec. 5.1, we stated the drawbacks of the purely sentence-level and token-level approaches, then introduce the contextual token-level approach (i.e., the contextual token-image matching model in Sec. 5.3.2) which combines these two approaches. In this section, we demonstrate a careful comparison between our vokenization process and the other two approaches from two perspectives: the retrieval methods and the supervision types. Experiments are conducted with the same hyper-parameters and dataset as "$\text{BERT}_{\text{12L/768H}}$+Voken-cls" in Table 5.2.

Sentence-Level Retrieval To conduct sentence-level retrieval, we first adapt the contextual token-image matching model in Sec. 5.3.2 to a sentence-image matching model. We then retrieve a related image for each sentence. As shown in Table 5.5, these retrieved images are used as two kinds of supervisions by putting classifiers at different places: in the row "SentLabel", we provide sentence-level supervision by using the classifier to predict the label for the whole sentence (similar to the BERT's "next-sentence prediction" (NSP) task); and in the row "Propagated", we provide token-level supervision by propagating sentence-level labels to all tokens in the sentences, and apply the classifier at each token (similar to our voken-classification task). The results of both kinds of supervisions are lower than our proposed vokens (in the row "Vokens"). One possible reason for these lower results is that finding an image that conveys the meaning of the whole sentence is hard. We also find that dense token-level supervision also outperforms the sentence-level supervision.

Token-level Retrieval Our proposed vokenization process is viewed as contextual token-level retrieval, which grounds tokens with whole sentences as context. We here consider a purely token-level retrieval method regarding term frequencies. The term frequency $\mathit{tf}(tok, x_i)$ (Manning et al., 2008) is calculated based on the occurrence $\#(tok, x_i)$ of the token tok in the image x_i's captions.

$$\mathit{tf}(tok, x_i) = \frac{\#(tok, x_i)}{\sum_{tok'} \#(tok', x_i)}$$

We then convert this term frequency to the conditional distribution via Boltzmann distribution:

$$p(x_i \mid tok) = \frac{\exp\left(\mathit{tf}(tok, x_i)/\gamma\right)}{\sum_{x'} \exp\left(\mathit{tf}(tok, x')/\gamma\right)}$$

where γ is temperature. We stochastically map the tokens to images with this conditional distribution $p(x_i \mid tok)$. The results trained with these special vokens are shown in Table 5.5 as "Term Frequency". Overall, token-level supervision is still better than the sentence-level supervision (as in the row "SentLabel"). However, among the models trained with token-level supervision, this token-level retrieval method neglects the contextual information thus is worse compared with sentence-level (in the row "Propagated") and contextual token-level retrieval methods (in the row "Voken") .

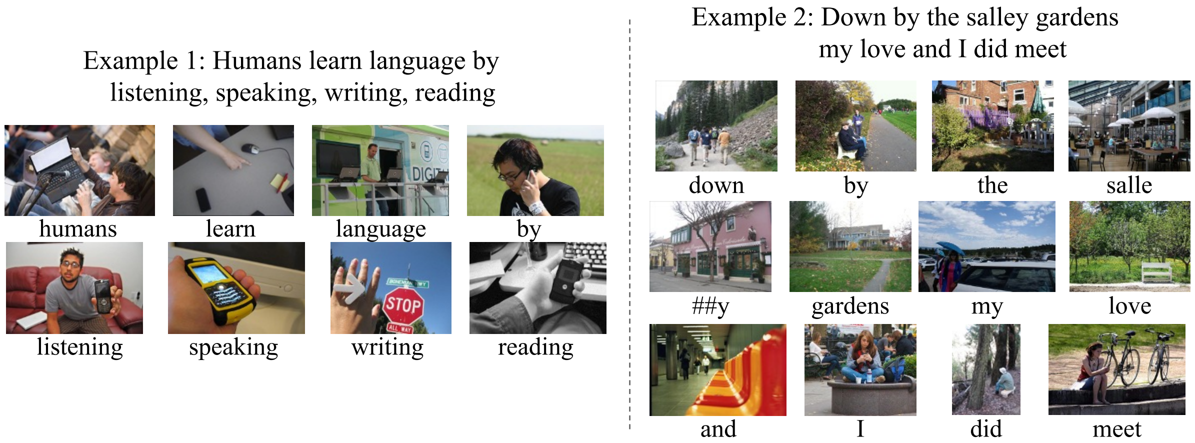

Example 1: Humans learn language by listening, speaking, writing, reading
humans
learn
language
by
listening
speaking
writing
reading
Example 2: Down by the salley gardens my love and I did meet
down
by
the
salle
##y
gardens
my
love
and
I
did
meet

Figure 5.4: Visualization of model-generated vokens. Example 1 takes the leading sentence of this paper while Examples 2 takes Yeats's poet.

5.5.3 Visualization of Vokens

In Fig. 5.4, we visualize our generated vokens. The first example takes the leading sentence in our paper (without commas), which is also used in the imaginary example in Fig. 5.1. We also vokenize another sentence from William Yeats's poet "Down by the Salley Gardens" in Fig. 5.4. Although the vokenizer is trained on image-captioning datasets without localizing token-to-image annotations, the vokenizer shows a strong selectivity: different images are selected w.r.t the tokens. The contextual token-level retrieval could also disambiguate certain tokens (e.g., "down" in Example 2) with the help of its context. When the *unique* related image is hard to define, our vokenizer aims to ground the non-concrete tokens (e.g., "by"/"and"/"the") to relevant images: the voken for the token "by" in Example 2 (of Fig. 5.4) is better aligned with the [centering token, context] pair than the voken for the same token "by" in Example 1. This related visual information helps understand the language and leads to the improvement in Table 5.2. On the other hand, some tokens are not faithfully grounded (e.g., "writing" in Example 1) and we also observe a shift in alignment (e.g., the relevant image for the phrase "my love" in Example 2 is aligned to "my" instead of "love"). These misalignments are possibly caused by the limitations of sentence-image weak supervision in our training data since the strong token-image annotations are not available.

5.6 Conclusion

We explored the possibility of utilizing visual supervision to language encoders. In order to overcome the challenges in grounded language, we developed the vokenizer with contextual token-image matching models and used it to vokenize the language corpus. Supervised by these generated vokens, we observed a significant improvement over the purely self-supervised language model on multiple language tasks.

CHAPTER 6: COMBINING VISION AND LANGUAGE PRE-TRAINING METHODS FOR VIDEO UNDERSTANDING

6.1 Introduction

In recent years, state-of-the-art self-supervised methods have been exploring different directions for pre-training images and text representations, with Contrastive Learning (CL) providing strong results for vision representation learning (Oord et al., 2018; Chen et al., 2020b; He et al., 2020a; Chen et al., 2020c; Tian et al., 2020), and Language Modeling (LM) becoming the defacto standard in natural language processing (NLP) pre-training (Devlin et al., 2019a; Liu et al., 2019b; Yang et al., 2019; Lan et al., 2019). Both approaches are quite different from each other. A contrastive objective compares positive/negative examples at a coarse/sample level, focusing on global-content (e.g., for object detection) while a token modeling objective predict missing tokens from context at a much finer/sub-sample level to model sequential and short range interactions between tokens (e.g. in text generation tasks). Interestingly, video understanding naturally combines both types of requirements. 2D processing along the spatial dimensions of the video bears similarity to image processing, while 1D processing along the temporal dimension often involves modeling sequential events and short range coherence.

Hence, in this work, we propose to combine both text and image representation learning approaches for improved video pre-training, taking advantage of recent advances in self-supervised methods of both fields. We name our method as VIMPAC: VIdeo pre-training via Masked token Prediction And Contrastive learning. From language research, we adopt a ‘masked language model’ pre-training objective (Devlin et al., 2019a) where a model is trained to reconstruct local masked regions in images or videos. From the computer vision world, we borrow a contrastive learning objective, specifically the InfoNCE (Oord et al., 2018) objective applied on positive/negative video samples. While the masked language model objective encourages models to learn low-level semantics and sequential interaction, the contrastive loss provide a supervision for the models to learn more global and separable representations that are useful for many downstream tasks (e.g., action classification (Soomro et al., 2012b; Kuehne et al., 2011a; Carreira and Zisserman, 2017)). Combining both objectives allow to provide training signal covering complementary aspects of a video signal: while short range correlations can be predominantly modeled from the training signal of the mask-and-predict task, the contrastive learning objective can provide signal on a more coarse-grained global-context and semantic level.

However, unlike language and its compact vocabulary of discrete tokens, videos are typically represented as RGB pixels in an almost continuous, high dimensional vector space. Naively masking pixels in videos induces a prohibitive computation cost while also tending to over-emphasize local details. To overcome these issues, we first tokenize input videos using the latent codes of a pretrained Vector Quantized-Variational Auto-Encoder (VQ-VAE) (van den Oord et al., 2017; Ramesh et al., 2021) to encode them in smaller quantized representations on which a reconstruction model can then be trained with a masked token modeling objective. In practice, we also discovered that models trained with a uniform random token masking strategy can fail to learn meaningful and useful visual representations as neighboring pixels may contain very similar and correlated content (in particular along the temporal frame axis), making the task of predicting a randomly masked token from its visible neighbors easy. We therefore also introduce a block-masking scheme for videos by simultaneously masking video tokens in a contiguous 3D spatio-temporal block. Reconstructing such an extended spatio-temporal cube requires performing long-range predictions, forcing the models to learn a more complex set of relations between the video tokens, resulting in better visual representations.

Our contrastive learning approaches also departs from previous work in several aspects. First, since we apply the contrastive objective on token-discretized video samples and in combination with the token modeling loss, we observe strong performance without requiring the usual extensive set of data augmentations (Chen et al., 2020b,c; Qian et al., 2021; Feichtenhofer et al., 2021).

Second, we are able to leverage positive clip pairs that are temporally distant from each other (can be as far as 400 seconds away), while previous work favors using positives within a shorter range (maximum 36 seconds for uncurated videos in (Feichtenhofer et al., 2021) or 10 seconds in (Qian et al., 2021)).

We evaluate the performances of our method VIMPAC on several video understanding datasets including two temporal-heavy tasks, SSV2 and Diving48 on which it achieves state-of-the-art results with regard to both self-supervised and supervised pre-training works and a set of more spatial-heavy datasets (UCF101, HMDB51, and Kinetics-400) on which it achieve competitive results with regards to the literature. Overall, taking advantage of VQ-VAE discretized video tokens, we present a method for self-supervised learning of video representations that combines two general streams of research in self-supervision: masked language modeling and contrastive learning. Our contribution is 3-folds: (i) We apply the mask-then-predict task to video understanding and introduce the use of block masking. (ii) We propose a contrastive learning method which is able to achieve strong performance without spatial data augmentation. (iii) We empirically show that this method can achieve state-of-the-art results on several video classification datasets. We also present comprehensive ablation studies to analyze the various aspects of our proposed approach.

6.2 Methods

In this section, we present our proposed video pre-training method VIMPAC (illustrated in Fig. 6.1) as well its detailed components. We first introduce the mask-then-predict task in Sec. 6.2.1, and then the contrastive learning task in Sec. 6.2.2. Lastly, we discuss how these two tasks are combined in Sec. 6.2.3.

6.2.1 Mask-then-Predict Task

Suppose that a video clip input comprises T frames $\{f_1, f_2, \ldots, f_{\mathrm{T}}\}$, the mask-then-predict task learns video representations by predicting the masked contents from their spatio-temporal

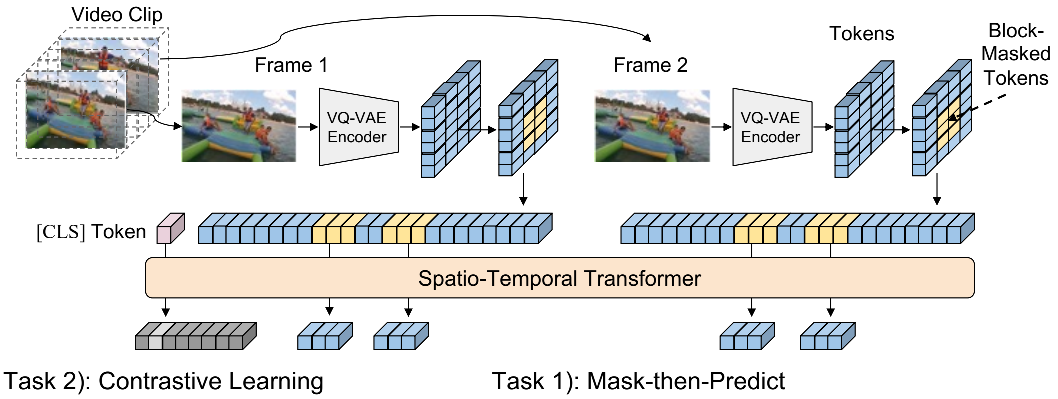


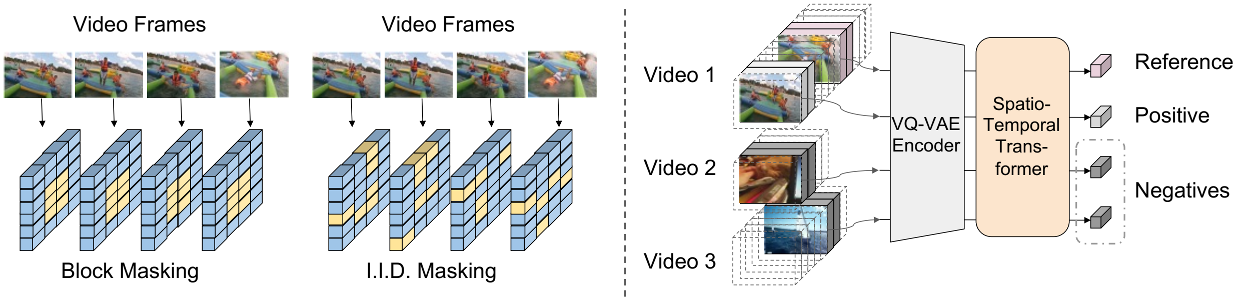
Figure 6.1: Illustration of our VIMPAC framework. Frames are sampled from the video clip and discretized by VQ-VAE encoder. The tokens from VQ-VAE are then block-masked (in light yellow blocks). The model is self-supervised by two tasks: 1) *mask-then-predict* task predicts the masked tokens from visible context; 2) *contrastive learning* task classifies the positive examples (details in Fig. 6.2) with the feature of the additional [CLS] token. For space limit, we only show 2 frames and a smaller token map.

context. Denote the set of mask-token locations as M, we learn to predict the original tokens $\{x_{t,i,j}\}$ (see details below) by optimizing the negative log-likelihood:

$$\mathcal{L}_{\text{mask}} = -\frac{1}{|M|} \sum_{t,i,j \in M} \log p_{t,i,j} \left(x_{t,i,j} \mid \{x_{t',i',j'}\}_{t',i',j' \in M^{\mathrm{C}}} \right), \tag{6.1}$$

where M^{C} is the complement of M and thus indicates the unmasked context.

Video Quantization with VQ-VAE. Since directly applying mask-then-predict over raw pixels and masking/predicting pixels leads to prohibitive computational costs and also tends to make the model overfit on detailed low-level visual information, we quantize the input videos with Vector Quantized-Variational Auto Encoder (VQ-VAE) (van den Oord et al., 2017; Ramesh et al., 2021). The VQ-VAE encoder takes an image as input and produces a token map, where the tokens belong to a predefined vocabulary V of cardinal 'vocabulary size'. The VQ-VAE decoder then tries to reconstruct the original image from these latent codes. In our method, we use a frozen and pretrained generic VQ-VAE encoder as a compressor that converts an input from an original input space $\mathbb{R}^{H \times W \times 3}$ into a discretized space $[V]^{\frac{H}{8} \times \frac{W}{8}}$. We independently apply the



(a) Illustration of Block and I.I.D. Masking. (b) Illustration of Contrastive Learning.

Figure 6.2: Illustration of pre-training task details. In (a), block masking constructs the 3D-contiguous masking cube while i.i.d masking independently samples masked tokens. In (b), given the reference video clip, the positive clip is uniformly sampled from the same video (video 1) while negative clips are sampled from other videos (video 2 and video 3). No spatial augmentations are applied to the raw video clips.

VQ-VAE encoder to each frame f_t inside a clip.[1] We keep the VQ-VAE weights frozen and do not finetune or adapt this model on our corpus.

Block Masking For sampling tokens to mask, the original BERT methods proposes the i.i.d. (independent and identically distributed) random mask M_{iid} that constitutes of masked tokens:

$$M_{\text{iid}} = \{(t, i, j) \mid \mathcal{U}_{t,i,j}[0, 1] < \xi\}, \tag{6.2}$$

where $\mathcal{U}_{t,i,j}[0, 1]$ is the uniform distribution from 0 to 1. Intuitively, ξ is the expectation of masked-token ratio and hence controls the difficulty of our mask-then-predict task. In our early experiments, we found it easy to infer a masked token from its direct spatio-temporal neighbours (e.g., neighboring frames in a video tend to look similar thus contain similar tokens). To overcome this issue, we propose to use block masking (see Fig. 6.2 (a)), which masks continuous tokens inside spatio-temporal blocks. For each mask block B, we randomly sample lower ($B_{*,0}$) and upper boundaries ($B_{*,1}$) for each of the temporal (T), height (H), and width (W) dimensions. The direct product of the intervals delimited by these boundaries constructs the block mask. The final

[1] We do not use the Video-VQVAE (Walker et al., 2021) method since the image-trained VQVAE (Ramesh et al., 2021) has been pretrained on a very large image corpus and as a consequence cover a much more diverse set of visual scenes and elements.

mask M_{block} is the union of them:

$$M_{\text{block}} = \bigcup_B [B_{T,0}, B_{T,1}] \times [B_{H,0}, B_{H,1}] \times [B_{W,0}, B_{W,1}]. \tag{6.3}$$

6.2.2 Contrastive Learning

Contrastive learning aims to distinguishing positive pairs from negative pairs (see Fig. 6.2 (b)). For each video $video_i$, we uniformly and independently sample two clips c_i, c'_i as a positive pair, while the clips in a batch belonging to other videos are used to construct negative pairs. A model (described in Sec. 6.2.4) processes clips c_i, c'_i to build respective vector representations f_i, f'_i and an InfoNCE (Oord et al., 2018) loss is used to distinguishes the positive feature pair (f_i, f'_i) from the negative pairs $\bigcup \{\{(f_i, f_k), (f_i, f'_k)\} \mid k \neq i\}$ for each clip c_i:

$$\mathcal{L}_{\text{InfoNCE}}(i) = -\log \frac{\exp\left(f_i^\top f'_i/\gamma\right)}{\sum_{k\neq i} \exp\left(f_i^\top f_k/\gamma\right) + \sum_k \exp\left(f_i^\top f'_k/\gamma\right)}, \tag{6.4}$$

which we combine with the symmetric loss $\mathcal{L}'_{\text{InfoNCE}}(i)$ for paired clip sample c'_i.

The final loss for a mini batch $\mathcal{L}_{\text{cl}}$ is the average loss for all n clips in the mini-batch:

$$\mathcal{L}_{\text{cl}} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{\text{InfoNCE}}(i) + \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}'_{\text{InfoNCE}}. \tag{6.5}$$

6.2.3 Pre-Training Objective

We combine the two pre-training methods discussed above to define the overall objective as:

$$\mathcal{L} = \mathcal{L}_{\text{mask}} + \alpha\gamma\mathcal{L}_{\text{cl}}, \tag{6.6}$$

where α is a hyperparameter controlling the weight of the contrastive loss and multiplying the temperature γ will smooth training (Grill et al., 2020; Chen et al., 2021). The inputs for both tasks are shared in mini-batches with the contrastive learning loss using the same block-masked inputs necessary for the mask-then-predict task. We highlight that the masked tokens for the denoising task are the only noise introduced in the contrastive learning, and that no other data augmentation is applied to raw pixels, in contrast to previous vision contrastive learning methods in which data-augmentation was paramount to the final performances of the model. This phenomenon is empirically studied in Sec. 6.4.2.3.

6.2.4 Modeling

The model architecture follows the standard transformer architecture in its post-layer-norm variant (Vaswani et al., 2017; Devlin et al., 2019a) with two more recent additions: divided temporal-spatial attention (Bertasius et al., 2021), and sparse spatial attention (Child et al., 2019). The model embedding layer maps the discrete tokens $\{x_{t,i,j}\}$ of a quantized input video (see Sec. 6.2.1) into dense vectors and sum them with positional embeddings. The backbone transformer model then outputs corresponding features $\{h_{t,i,j}\}$. We append an additional `[CLS]` token to each input sequence following (Devlin et al., 2019a) and use its output feature h_{cls} as a representation for the whole video. For pre-training, we use two heads: a 2-layer MLP after each token outputs $\{h_{t,i,j}\}$ for the mask-then-predict task following BERT (Devlin et al., 2019a), and a 3-layer MLP after the CLS output h_{cls} for the contrastive learning task following SimCLR (Chen et al., 2020b). For fine-tuning on classification tasks, we remove the pre-training heads and add a fully-connected layer to the `[CLS]` output h_{cls}.

6.3 Experiments and Results

6.3.1 Datasets

For **pre-training**, we use the HowTo100M dataset (Miech et al., 2019). This dataset is constructed by searching YouTube videos with a list of text queries, it is significantly larger and more diverse than human-annotated datasets such as Kinetics 400 (Carreira et al., 2019). HowTo100M has 1.2M uncurated videos, with an average duration of 6.5 minutes. We only use videos and do not use other signals such as ASR captions in this dataset. For **downstream** evaluation, we experiment with several action classification datasets: UCF101 (Soomro et al., 2012a), HMDB51 (Kuehne et al., 2011b), Kinetics-400 (Carreira and Zisserman, 2017), SSV2 (Goyal et al., 2017a), and Diving48 (Li et al., 2018). It is important to note that in many cases, actions in UCF101, HMDB51, and Kinetics-400 can be recognized from a single frame of the video, thus these datasets are '*spatially-heavy*'. As a consequence, image-level methods (Bertasius et al., 2021; Radford et al., 2021b) show competitive results without modeling the temporal interactions inside the videos. To test the video model's ability beyond recognizing static images, we lay our focus on '*temporally-heavy*' datasets (SSV2 and Diving48), in which action recognition from a single frame is more difficult. For example, it is almost impossible to distinguish two SSV2 classes *moving something up* and *moving something down* without reasoning across frames, and the same for different diving classes in Diving48.

6.3.2 Experimental Setup

Our model shapes follow $\text{BERT}_{\text{LARGE}}$ with 24 layers and hidden size 1024, but with halved attention head size and MLP intermediate size as in (Child et al., 2019). For **pre-training**, we train the model for 100 epochs on the HowTo100M dataset with frames sampled at 2 FPS. We create the training inputs by sampling two clips from each video as described in Sec. 6.2.2. To reduce computation cost, the first 90 epochs are trained with a smaller input resolution (#frames T=5 and frame size S=128) and we increase the spatial resolution (T=5, S=256) for the last 10 epochs following (Devlin et al., 2019a). Positional embeddings are interpolated as in (Dosovitskiy et al., 2021) when input resolution changes. Importantly, our pre-training scheme does not involve spatial augmentations: all frames are resized and centered cropped without random flipping, color distortion, etc. We use a batch size of 1024 in pre-training. The number of negative clips used for contrastive learning is 255 for the first 90 epochs and 127 for the last 10 epochs. The num-

Table 6.1: Comparison with state-of-the-art. Our model outperforms previous works on SSV2 and Diving48 dataset while showing competitive results on other datasets. UCF101 and HMDB51 are average over three train-val splits.

Method	Temporally-Heavy		Spatially-Heavy		
	SSV2 (Goyal et al., 2017a)	Diving48 (Li et al., 2018)	UCF101 (Soomro et al., 2012b)	HMDB51 (Kuehne et al., 2011a)	K400
Previous SotA	65.4 (Arnab et al., 2021)	81.0 (Bertasius et al., 2021)	98.7 (Kalfaoglu et al., 2020)	85.1 (Kalfaoglu et al., 2020)	84.8 (Arnab et al., 2021)
w/o Temporal Modeling	36.6 (Bertasius et al., 2021)	-	92.0 (Radford et al., 2021b)	-	77.6 (Bertasius et al., 2021)
Self-supervised Pre-Training					
K400 Self-Sup.	55.8 (Feichtenhofer et al., 2021)	-	96.3 (Feichtenhofer et al., 2021)	75.0 (Feichtenhofer et al., 2021)	-
MIL-NCE	-	-	91.3	61.0	-
MMV	-	-	95.2	75.0	-
MoCo	53.2	-	92.9	-	-
VIMPAC	68.1	85.5	92.7	65.9	75.3
Supervised Pre-Training					
K400 Sup.	63.1 (Feichtenhofer et al., 2019)	-	96.8 (Tran et al., 2018)	82.5 (Wang et al., 2019b)	81.5 (Kondratyuk et al., 2021)
TimeSformer	62.3	81.0	-	-	80.7
ViViT	65.4	-	-	-	80.6

ber of negative pairs used in our ablation analyses is kept constant at 127.[2] For **fine-tuning**, we use more input frames (T=10 and S=256), and batch size 128. We sample frames at 2 FPS for datasets with longer videos (i.e., UCF101 and Kinetics-400), and sample 4 FPS for datasets with shorter videos (i.e., HMDB51, SSV2, Diving48). During inference, we follow (Feichtenhofer et al., 2019, 2021) to use 3 spatial crops and 10 temporal crops (in total 30 crops), and average their prediction scores as the final score.[3] All models are trained with AdamW (Loshchilov and Hutter, 2018) optimizer with linear warm-up and linear learning rate decay. We observe similar pre-training instability as reported in (Chen et al., 2020a, 2021) and follow their practice to sequentially choose learning rate at 1e-3, 5e-4, 3e-4, ..., until convergence.

6.3.3 Results

We compare our primary results with previous work in Table 6.1. We expand the results that are most related to our work: self-supervised training on uncurated videos and supervised pre-training with transformers. For other results, we select the best-performing models to our knowledge and denote their reference in the table.

[2] During pre-training, we always accumulate the gradient to a batch size of 1024 before updating the weights but use different numbers of negative examples. We analyze this effect in Sec. 6.4.2.3.

[3] As in (Bertasius et al., 2021; Arnab et al., 2021), we observe that the performance is saturated at 4∼5 temporal crops for our model.

Our model VIMPAC sets the new state of the art on the two temporally-heavy datasets SSV2 and Diving48, where we achieve 2.7% and 4.5% absolute improvement, respectively, over previous best models among all self-supervised and supervised pre-trained methods. This is especially surprising considering the two previous SotA models ViViT (Arnab et al., 2021) and TimeSformer (Bertasius et al., 2021) both use large-scale supervised pre-training, and ViViT also uses various regularization techniques (e.g., stochastic depth (Huang et al., 2016), random augment (Cubuk et al., 2020) and mixup (Zhang et al., 2018)). VIMPAC also achieves competitive results on other three spatially-heavy datasets: UCF101, HMDB51, and Kinetics-400. As discussed in Sec. 6.3.1, recognizing actions in SSV2 and Diving48 require a strong temporal reasoning ability, while in the other datasets, spatial understanding is dominant. Some relatively low results of our VIMPAC (e.g., K400) are thus possibly due to the VQ-VAE spatial information loss. To illustrate this, we show a comparison between the SotA models [4] with temporal modeling in the first row and the ones without in the second row of Table 6.1. Note the gaps between these two types of models are significantly larger for temporally-heavy datasets (SSV2) than spatially-heavy datasets (UCF101, Kinetics-400), demonstrating the importance of temporal modeling for temporally-heavy datasets. We also show the methods pre-trained on HowTo100M that take other modalities to help video learning thus beyond the scope of visual self-supervised learning.

Previous self-supervised pre-training such as MoCo (Feichtenhofer et al., 2021) are good at global understanding, but the pre-training schema does not consider the internal interactions inside videos (especially for the temporal dimensions). As a result, it could reach or even outperform the supervised alternatives on UCF101. However, it shows lower results on SSV2 compared to the transformer models (Bertasius et al., 2021; Arnab et al., 2021) (although with different backbone models) that warm up from image-pre-trained models and learn the temporal interactions directly from the downstream tasks.

[4] Some SotA models are pre-trained with extremely large (weakly-)supervised datasets, e.g., IG65M (Ghadiyaram et al., 2019) in (Kalfaoglu et al., 2020) and JFT-300M (Sun et al., 2017) in (Arnab et al., 2021).

Table 6.2: Impact of model size. ‘Params’ is the number of parameters. ‘Speed’ is the normalized pre-training speed measured by #videos/second on one V100 GPU. ‘Mask-Accu.’ and ‘CL-Loss’ are mask-then-predict accuracy and contrastive learning loss to indicate the pre-training performance. ‘UCF101’ is the fine-tuning accuracy on UCF101 dataset. First line is defautly used in analysis and the configuration producing final results are underlined.

Layers	Dim	Params	Speed	Mask-Accu.↑	CL-Loss ↓	UCF101↑
6	512	29.4M	32.0	17.2	1.06	69.4
6	768	63.0M	21.0	17.7	1.03	75.0
12	512	54.7M	18.1	17.9	1.02	76.6
12	768	119.7M	11.2	18.4	1.00	78.1
24	1024	210.1M	5.0	18.7	0.98	78.5

6.4 Analysis

We also analyze the model’s scalability and the effectiveness of our pre-training methods. To save computation, for all analyses, we use a smaller model (6-layer transformer with hidden dimension 512) and smaller input resolution (5 input frames with spatial size 128, i.e., T=5, S=128) throughout this section, unless otherwise stated. We also perform pre-training with fewer epochs (i.e., 10). For downstream tasks, we use the same input resolution as pre-training (i.e., T=5, S=128), and we use 2 temporal crops for inference. All results are reported on the train-val split 1 if applicable.

6.4.1 Scalability

Model. In Table 6.2, we illustrate the scalability of our method with different model sizes (i.e., number of layers and hidden dimensions). Larger models have more parameters (‘Params’) and higher computational cost (measured by the normalized pre-training ‘Speed’). To evaluate the pre-training tasks performance, we provide both pre-training metrics (mask-then-predict accuracy denoted by ‘Mask-Accu.’, and contrastive learning loss denoted by ‘CL-Loss’) and UCF101 downstream fine-tuning results. As the size of the model grows, the fine-tuning results show consistent improvement with the pre-training metrics. Note that for the last row in Table 6.2, we halve the attention head and MLP intermediate dimensions.

Table 6.3: Impact of input resolutions T and S. 'Mask-Accu.' and 'CL-Loss' are the pre-training metrics. 'UCF101' indicates the UCF101 fine-tuning results with the pre-training resolution. 'UCF101-Full-Reso.' indicates the full-resolution fine-tuning with T=10 and S=256.

#frames T	Frame Size S	Params	Pre-train Speed	Mask-Accu.↑	CL-Loss↓	UCF101↑	UCF101-Full-Reso.↑
5	128	29.4M	32.0	17.2	1.06	69.4	73.8
10	128	29.4M	16.5	17.2	0.96	74.2	74.6
5	256	29.4M	8.4	10.8	0.93	72.9	75.7
10	256	29.4M	4.4	10.6	0.85	78.1	78.1

Input Resolution. In Table 6.3, we show model scalability over input resolution (i.e., #frames T and frame size S). With the same frame size S, longer clips perform better than shorter clips (e.g., T=10, S=128 is better than T=5, S=128). With the same number of input frames T, larger frame size improves the performance (e.g., T=10, S=256 is better than T=10, S=128). For each pre-training resolution, we also try to fine-tune under a full-resolution with T=10, S=256 (denoted as 'UCF101-Full-Reso.'). As in pre-training, fine-tuning with larger resolution generally improves the results. Although longer and smaller clips (T=10, S=128) show better results than shorter and larger clips (T=5, S=256) when using the same pre-training and fine-tuning resolutions, they show different trends with the full-resolution fine-tuning. Increasing frame size during fine-tuning (the second block in Table 6.3) only improves the UCF101 result by 0.4, while increasing the clip length (the third block) improves the UCF101 result by 3.8. These results call for a need of pre-training with large spatial size, and we follow this practice in our large-scale experiments as in Sec. 6.3.3.

6.4.2 Pre-Training Methods

We analyze the key designs of our two pre-training tasks. When analyzing the mask-then-predict task in Sec. 6.4.2.2, we exclude the contrastive learning loss (by setting loss ratio α=0) to preclude potential side effects. However, we still use masked prediction loss when assessing the contrastive learning task in Sec. 6.4.2.3 as we observe very low performance with only contrastive learning objective.

Table 6.5: Impact of masking strategies. Models are pre-trained with only mask-then-predict.

Strategy	Frame Size S	Mask-Accu.↑	UCF101 ↑
block	128	17.6	68.3
i.i.d.	128	24.3	63.5 (-4.8)
block	256	11.2	69.5
i.i.d.	256	19.5	61.4 (-8.1)

Table 6.6: Impact of masking ratios. Models are pre-trained with only mask-then-predict. Default setup is underlined.

Strategy	#Blocks	Ratio	Mask-Accu.↑	UCF101 ↑
block	4	11.9%	17.9	66.8
block	5	14.5%	17.6	68.3
block	6	17.0%	17.3	67.3

6.4.2.1 The Impact of Pre-Training

We first compare different pre-training tasks and the non-pre-training results. As shown in Table 6.4, mask-then-predict is good at temporally-heavy datasets (SSV2, Diving48) while contrastive learning improves the spatially-heavy datasets. We also compare with the non-pre-training results (the first row of Table 6.4) and observe that both tasks significantly improve the results. We notice that these non-pre-training results are lower than previous from-scratch models, which might be caused by the difficulty in training video transformers (Bertasius et al., 2021; Arnab et al., 2021) and the information loss in our input quantization process (Ramesh et al., 2021).

Table 6.4: Impact of pre-training tasks. ‘MP’=Mask-then-Predict, ‘CL’=Contrastive Learning task.

MP	CL	Temporally-Heavy		Spatially-Heavy		
		SSV2	Diving48	UCF101	HMDB51	K400
✗	✗	1.2	10.0	41.3	19.0	41.0
✗	✓	32.5	26.3	57.1	30.7	47.0
✓	✗	41.4	37.2	68.3	35.3	53.7
✓	✓	41.1	37.5	69.4	37.8	54.5

6.4.2.2 Mask-then-Predict

Block Masking versus I.I.D. Masking. We first compare our proposed block masking strategy and the uniform i.i.d. masking strategy (discussed in Sec. 6.2.1 and illustrated in Fig. 6.2). As shown in Table 6.5, although the i.i.d. masking achieves higher pre-training mask-token-prediction accuracy (‘Mask-Accu.’), it shows lower downstream results (‘UCF101’) than block masking. The higher mask accuracy is possibly due to the easier i.i.d. mask-then-predict task. The existence of such a trivial solution potentially prevents the model from learning useful video

Table 6.7: Impact of maximum sampling distance d_{max} (seconds) between two positive clips.

d_{max}	Mask-Accu.↑	CL-Loss↓	UCF101↑
$\underline{\infty}$	17.2	1.06	69.4
30	17.3	0.77	69.0 (-0.4)
10	17.4	0.61	68.3 (-1.1)
0	17.5	0.41	66.7 (-2.7)

Table 6.8: Impact of number of negative samples.

#samples	Mask-Accu.↑	CL-Loss↓	UCF101↑
$\underline{128}$ -1	17.2	1.06	69.4
256 - 1	17.1	1.30	69.2
512 - 1	17.2	1.56	70.4
1024 - 1	17.0	1.86	69.8

representations for downstream tasks. Meanwhile, we also find that the model with larger input frame size 256 benefits more from the block masking strategy, because the adjacent tokens are closer in the original 2D image for these larger frames. Hence, the spatial locality is amplified.

Masking Ratio. In Table 6.6, we study the impact of masking ratio, by varying the number of masked blocks for block masking. Empirically, the result differences among different masking ratios are marginal and the original BERT's 15% masking ratio (with roughly 5 masking blocks) works slightly better. Thus we always select the number of mask blocks whose induced masking ratio is closest to 15%.

6.4.2.3 Contrastive Learning

Positive Sampling Distance. As illustrated in Sec. 6.2.2 and Fig. 6.2.(b), we uniformly sample positive clip pairs across the whole video without any distance restriction. To analyze the effect of such a sampling strategy, We perform a set of experiments by varying the maximum sampling distance d_{max} (in seconds) between two positive clips. The results are shown in Table 6.7. d_{max}=∞ denotes our default setup without any distance restriction. d_{max}=0 samples two same clips, and d_{max}=10 samples two positive clips with a maximum distance of 10 seconds. Although previous contrastive learning methods (Qian et al., 2021; Feichtenhofer et al., 2021) favor the sampling of temporal positives within a shorter range (e.g., maximum 36 seconds for uncurated videos in (Feichtenhofer et al., 2021)), we observe a performance gain when using larger distance. We also want to emphasize that the results with d_{max}=10 and d_{max}=0 are not better than the model pre-trained with only mask-then-predict (UCF101 accuracy 68.3), which suggests that short-range contrastive learning does not improve upon our mask-then-predict task. This is potentially because our mask-then-predict already gives the model the ability to model local interactions, thus contrastive learning objective can only be useful when it focuses on longer-range interactions.

Number of Negative Samples. Previous constrastive learning methods (Chen et al., 2020b, 2021; Feichtenhofer et al., 2021) benefit from more negative samples. In this section, we show that the number of negative samples has less impact on our method when mask-then-predict task is added. As shown in Table 6.8, we experiment with different contrastive learning sample sizes (i.e., n in Sec. 6.2.2 which is 1 + number of negative samples) and always accumulate the gradients to 1024 samples before updating the parameters. Although increasing sample size makes the contrastive learning task harder (reflected by 'CL-Loss'), it does not show clear evidence of improving UCF101 downstream performance.

Input Masking as Augmentation. Most self-supervised visual representation learning methods (Chen et al., 2020c,b; Grill et al., 2020; Feichtenhofer et al., 2021; Qian et al., 2021) based on contrastive learning suffer from a large drop when removing strong spatial augmentations. In contrast, our pre-training does not use any spatial augmentations on raw frames, such as flipping and random cropping. However, as we tie the input between mask-then-predict and contrastive learning to reduce computation cost, the random masking noise is naturally introduced. We here investigate its impact in Table 6.9. When pre-trained jointly with mask-then-predict, adding mask noise improves UCF101 accuracy by +2.0; however, when pre-trained without it, adding mask noise hurts the performance (-1.6). We hypothesize that this is due to the large input mismatches between pre-training and fine-tuning when mask-then-

Table 6.9: Impact of mask augmentation in contrastive learning. 'MP'=Mask-then-Predict. 'CL-Mask'=Use input mask in CL. Default setup is underlined.

MP	CL-Mask	Mask-Accu.↑	CL-Loss↓	UCF101↑
✗	✗	-	1.07	57.1
✗	✓	-	1.08	55.5
✓	✗	17.2	1.04	67.4
✓	✓	17.2	1.06	69.4

predict objective is not applied. Noisy masking creates ‘holes’ to the input token maps during pre-training, while for fine-tuning the input token maps are intact. When mask-then-predict task is applied, it guides the model to fill these holes, thus reducing this mismatch and allowing the contrastive learning task to benefit from noisy masking as a type of regularization. In constrast, this input mismatch becomes dominant when only using the contrastive learning objective.

6.5 Conclusion

We presented the video pre-training framework VIMPAC that introduces mask-then-predict task to video self-supervised learning. mask-then-predict task helps model spatio-temporal interactions that are important for video understanding. We used the VQ-VAE quantizer and propose the block masking method that is essential to overcome the strong locality in video. The contrastive learning task is also added to learn separable global features. Different from previous methods, our contrastive learning does not use data augmentation over raw frames and is less sensitive to the temporal sampling distribution for positive pairs. We showed that our frameworks could achieve state-of-the-art performance on two temporally-heavy dataset (SSV2 and Diving48) and reach competitive results on other datasets. Detailed analyses are provided regarding the model scalability and task design.

CHAPTER 7: SUMMARY, LIMITATIONS, AND FUTURE WORK

7.1 Summary of Contributions

We presented multiple pre-training frameworks for advancing the vision-and-language research. We developed the first vision-and-language pre-training frameworks (LXMERT) in handling image-text interactions and show a significant improvement over non-pre-trained models on multiple benchmark dataset. We study the impact of single-modality pre-training to vision-and-language tasks, where visual features containing more semantic information generally performs better. We next explore whether we could build pre-training frameworks to improve single-modality tasks with the help from multimodal data. In Vokenization, we present a visually-supervised language model that learn the language meaning from corresponding images. Lastly, in VIMPAC, we combine the language modeling and contrastive learning pre-training methods and show that they are complementary to each other. All these projects have publicly available code that attracted substantial attention from the community. We hope that our works could inspire and help future research in this area as well.

7.2 Limitations and Future Work

7.2.1 Vision-and-Language Pre-training

After the development of vision-and-language pre-training models (Sun et al., 2019; Tan and Bansal, 2019; Lu et al., 2019), a lot of works in building image-and-text pre-training have been presented (Li et al., 2019a; Su et al., 2020; Chen et al., 2020d; Zhou et al., 2020; Huang et al., 2020; Li et al., 2020b; Zhang et al., 2021b; Li et al., 2021b). Researchers also extend the pre-training method to other problem setups, e.g., video-and-language (Sun et al., 2019; Zhu and

Yang, 2020; Li et al., 2020a; Miech et al., 2020), instruction-guided navigation (Zhu et al., 2020; Hao et al., 2020; Hong et al., 2021), documents (Xu et al., 2020), and audio (Akbari et al., 2021). Although current vision-and-language pre-training frameworks has shown success on diverse vision-and-language tasks, there are some directions that current models have less explored. First, the pre-training frameworks are built on deeply-interacted models that do not support efficient retrieval. ViLBERT (Lu et al., 2019) allows retrieving images from about a set of 5K images, which is far away from industry requirement of billions (even trillions) images. CLIP (Radford et al., 2021a) tries to solve this problem by building additive models (Hessel and Lee, 2020), but it still shows a gap to the deeply-interacted models. Given the industry interest in retrieval-based problems (e.g., searching, recommendation), we expect substantial progress in building retrieval-specific pre-trained models in the next few years. Second, the embodied systems (e.g., navigation, assistant) have a strong need in our real life and usually lack specific data. Thus, it would be an ideal scenario where pre-training could help. Besides a few works in in-door navigation pre-training (e.g., Hong et al. (2021)) that is bounded by the small number of environments (i.e., 90), the power of pre-training frameworks in embodied tasks has not shown yet. Third, balanced data such as image-caption and video-subscription only contribute a part of our daily life. For most cases, the multimodal data is imbalanced (in Sec. 2.2.3) that one modality dominates, and other modality helps with complementary information. For examples, news has long text and a few related figures, while movies have millions of image frames with only a few-line descriptions. We think that the understanding of these imbalanced data would be important. It will also require developing new models, finding new pre-training data, and designing new pretext tasks.

7.2.2 Building Multi-modal Learner

Most of us learn from the multi-modal worlds and apply the knowledge to all kinds of tasks: pure-language tasks, pure-vision tasks, and vision-and-language tasks. Thus, building a multi-modal learner that leverages all these supervisions is appealing. Our Vokenization (Tan and Bansal, 2020) (in Chapter 5) is an initial step to pursue this goal by improving language understanding from multimodal data. It shows improvement on pure-language tasks, but has two main constraints: 1. the pre-trained BERT (Devlin et al., 2019a) and ResNet (He et al., 2016) model are needed in training the retriever, which possibly causes information leak from the original BERT model. 2. The method has not shown the ability to scale up along the amount of vision-and-language data. We partially resolve these limitations in our recent work (Tang et al., 2021) by including the video dataset and using a pure knowledge-distillation method. The requirements of the pre-trained language models are thus removed, and the video data can be easily scaled up. Meanwhile, concurrent works (Desai and Johnson, 2020; Zhang et al., 2020; Radford et al., 2021a) explore the possibility to improve pure-vision tasks from the multimodal data. Unsupervised VisualBERT (Li et al., 2021b) and UniT (Hu and Singh, 2021) starts to study a multitasking model by applying both language tasks, vision tasks, and vision-and-language tasks. Given these pieces, we are now at the time to build a real multi-modal learner that could benefit from the redundant and complementary information inside the multimodal signals.

7.2.3 The Universal Model

Current vision-and-language models are designed to contain multiple contents. In LXMERT (Tan and Bansal, 2019), we have the vision ResNet (He et al., 2016) backbone, Regional Proposal Network (RPN) (Ren et al., 2015), vision transformer encoder, language transformer (Vaswani et al., 2017) encoder, and the cross-modal transformer encoder. This five-encoder structure is reduced in recent works. UNITER (Chen et al., 2020d) removes the vision transformer and language transformer modules but only keeps the cross-modal transformer encoder. PixelBERT (Huang et al., 2020) removes the Regional Proposal Network (RPN) and only uses the visual backbone. ViLT (Kim et al., 2021) goes one step further by removing the separate visual backbone. This provides a unified model to process the vision and language data. However, the input embeddings are still treated differently. The vision input is converted to vectors by the fully-connected layer, and the language input is mapped to embeddings by looking up a vector dictionary. We thus seek for a truly unified model that could treat image and text with the same input module.

Current progress focuses on unifying the image and text processing. The unified models for video, audio, kinetics, and other modalities are also important to study. Besides the detailed modeling choices, the pre-training data and tasks to warm up this universal model is another critical problem. The model is ideally supervised by all the data we had, but practically there would be a primary task to lead the learning process and build the first curriculum. Lu et al. (2021) takes language modeling as the universal pre-training tasks. Papadimitriou and Jurafsky (2020) also tries music, which is another interesting attempt. In the future, we might need to find suitable tasks and data for building this universal model.

REFERENCES

Akbari, H., Yuan, L., Qian, R., Chuang, W.-H., Chang, S.-F., Cui, Y., and Gong, B. (2021). Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *arXiv preprint arXiv:2104.11178*.

Aker, A. and Gaizauskas, R. (2010). Generating image descriptions using dependency relational patterns. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1250–1258.

Alberti, C., Ling, J., Collins, M., and Reitter, D. (2019). Fusion of detected objects in text for visual question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2131–2140.

Altman, N. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American statistician*, 46(3):175–184.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018a). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086.

Anderson, P., Shrivastava, A., Parikh, D., Batra, D., and Lee, S. (2019). Chasing ghosts: Instruction following as bayesian state tracking. In *NeurIPS*.

Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., and Van Den Hengel, A. (2018b). Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., and Parikh, D. (2015). Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bender, E. M. and Koller, A. (2020). Climbing towards nlu: On meaning, form, and understanding in the age of data. In *ACL*.

Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML*.

Bisk, Y., Holtzman, A., Thomason, J., Andreas, J., Bengio, Y., Chai, J., Lapata, M., Lazaridou, A., May, J., Nisnevich, A., et al. (2020). Experience grounds language. *arXiv preprint arXiv:2004.10151*.

Bloom, P. (2002). *How children learn the meanings of words*. MIT press.

Carreira, J., Noland, E., Hillier, C., and Zisserman, A. (2019). A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*.

Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y. (2017). Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE.

Changpinyo, S., Sharma, P., Ding, N., and Soricut, R. (2021). Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*.

Chen, H., Suhr, A., Misra, D., Snavely, N., and Artzi, Y. (2019a). Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. (2020a). Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020b). A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Chen, X., Fan, H., Girshick, R., and He, K. (2020c). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Chen, X., Xie, S., and He, K. (2021). An empirical study of training self-supervised visual transformers. *arXiv e-prints*, pages arXiv–2104.

Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. (2020d). UNITER: Universal image-text representation learning. In *ECCV*. ECCV.

Chen, Y.-C., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. (2019b). Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*.

Child, R., Gray, S., Radford, A., and Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2015). Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, pages 2067–2075.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *ACL*.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

Desai, K. and Johnson, J. (2020). Virtex: Learning visual representations from textual annotations. *arXiv preprint arXiv:2006.06666*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019a). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019b). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., and Smith, N. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Elliott, D., Frank, S., Sima'an, K., and Specia, L. (2016). Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74.

Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., and Forsyth, D. (2010). Every picture tells a story: Generating sentences from images. In *European conference on computer vision*, pages 15–29. Springer.

Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211.

Feichtenhofer, C., Fan, H., Xiong, B., Girshick, R., and He, K. (2021). A large-scale study on unsupervised spatiotemporal representation learning. *arXiv preprint arXiv:2104.14558*.

Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.-P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., and Darrell, T. (2018). Speaker-follower models for vision-and-language navigation. In *NeurIPS*.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129.

Gao, P., Jiang, Z., You, H., Lu, P., Hoi, S. C. H., Wang, X., and Li, H. (2019a). Dynamic fusion with intra- and inter-modality attention flow for visual question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gao, P., You, H., Zhang, Z., Wang, X., and Li, H. (2019b). Multi-modality latent interaction network for visual question answering. *arXiv preprint arXiv:1908.04289*.

Gao, Y., Beijbom, O., Zhang, N., and Darrell, T. (2016). Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–326.

Gerber, R. and Nagel, N.-H. (1996). Knowledge representation for the generation of quantified natural language descriptions of vehicle traffic in image sequences. In *Proceedings of 3rd IEEE international conference on image processing*, volume 2, pages 805–808. IEEE.

Ghadiyaram, D., Tran, D., and Mahajan, D. (2019). Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12046–12055.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Gordo, A., Almazán, J., Revaud, J., and Larlus, D. (2016). Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, pages 241–257. Springer.

Goyal, R., Kahou, S. E., Michalski, V., Materzyńska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. (2017a). The ”something something” video database for learning and evaluating visual common sense.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (2017b). Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (2017c). Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. (2020). Bootstrap your own latent a new approach to self-supervised learning. In *NeurIPS*.

Hao, W., Li, C., Li, X., Carin, L., and Gao, J. (2020). Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13137–13146.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020a). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020b). Momentum contrast for unsupervised visual representation learning. In *CVPR*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hendrycks, D. and Gimpel, K. (2016). Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *https://openreview.net/forum?id=Bk0MRI5lg*.

Hermann, K. M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari, D., Czarnecki, W. M., Jaderberg, M., Teplyashin, D., et al. (2017). Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*.

Hessel, J. and Lee, L. (2020). Does my multimodal model learn cross-modal interactions? it's harder to tell than you might think! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 861–877.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hong, Y., Rodriguez-Opazo, C., Qi, Y., Wu, Q., and Gould, S. (2020). Language and visual entity relationship graph for agent navigation. In *NeurIPS*.

Hong, Y., Wu, Q., Qi, Y., Rodriguez-Opazo, C., and Gould, S. (2021). Vlnœ bert: A recurrent vision-and-language bert for navigation. In *CVPR*.

Hoover, B., Strobelt, H., and Gehrmann, S. (2019). exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276*.

Hu, R., Andreas, J., Rohrbach, M., Darrell, T., and Saenko, K. (2017). Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813.

Hu, R., Fried, D., Rohrbach, A., Klein, D., Darrell, T., and Saenko, K. (2019). Are you looking? grounding to multiple modalities in vision-and-language navigation. In *ACL*.

Hu, R. and Singh, A. (2021). Unit: Multimodal multitask learning with a unified transformer. *arXiv preprint arXiv:2102.10772*.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. (2016). Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer.

Huang, H., Jain, V., Mehta, H., Ku, A., Magalhaes, G., Baldridge, J., and Ie, E. (2019). Transferable representation learning in vision-and-language navigation. In *ICCV*.

Huang, Z., Zeng, Z., Liu, B., Fu, D., and Fu, J. (2020). Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*.

Hudson, D. A. and Manning, C. D. (2019). Gqa: a new dataset for compositional question answering over real-world images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Iyer, S., Dandekar, N., and Csernai, K. (2017). First quora dataset release: Question pairs. *data. quora. com*.

Jain, V., Magalhaes, G., Ku, A., Vaswani, A., Ie, E., and Baldridge, J. (2019). Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1862–1872.

Jiang, H., Misra, I., Rohrbach, M., Learned-Miller, E., and Chen, X. (2020). In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276.

Jiang, Y., Natarajan, V., Chen, X., Rohrbach, M., Batra, D., and Parikh, D. (2018). Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*.

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Kalfaoglu, M. E., Kalkan, S., and Alatan, A. A. (2020). Late temporal modeling in 3d cnn architectures with bert for action recognition. In *European Conference on Computer Vision*, pages 731–747. Springer.

Kamath, A., Singh, M., LeCun, Y., Misra, I., Synnaeve, G., and Carion, N. (2021). Mdetr–modulated detection for end-to-end multi-modal understanding. *arXiv preprint arXiv:2104.12763*.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Kazemzadeh, S., Ordonez, V., Matten, M., and Berg, T. (2014). Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

Ke, L., Li, X., Bisk, Y., Holtzman, A., Gan, Z., Liu, J., Gao, J., Choi, Y., and Srinivasa, S. (2019). Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6741–6749.

Kiela, D., Conneau, A., Jabri, A., and Nickel, M. (2018). Learning visually grounded sentence representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 408–418.

Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Ringshia, P., and Testuggine, D. (2020). The hateful memes challenge: Detecting hate speech in multimodal memes. *arXiv preprint arXiv:2005.04790*.

Kim, J.-H., Jun, J., and Zhang, B.-T. (2018). Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1564–1574.

Kim, W., Son, B., and Kim, I. (2021). Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv preprint arXiv:2102.03334*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Knuth, D. E. (1973). The art of computer programming, volume 3: Searching and sorting. *Addison-Westley Publishing Company: Reading, MA*.

Koh, J. Y., Baldridge, J., Lee, H., and Yang, Y. (2021). Text-to-image generation grounded by fine-grained user attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 237–246.

Kondratyuk, D., Yuan, L., Li, Y., Zhang, L., Tan, M., Brown, M., and Gong, B. (2021). Movinets: Mobile video networks for efficient video recognition. *arXiv preprint arXiv:2103.11511*.

Krantz, J., Wijmans, E., Majumdar, A., Batra, D., and Lee, S. (2020). Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Ku, A., Anderson, P., Patel, R., Ie, E., and Baldridge, J. (2020). Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011a). Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011b). Hmdb: A large video database for human motion recognition. *2011 International Conference on Computer Vision*, pages 2556–2563.

Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27:2177–2185.

Li, J., Tan, H., and Bansal, M. (2021a). Improving cross-modal alignment in vision language navigation via syntactic information. *arXiv preprint arXiv:2104.09580*.

Li, L., Chen, Y.-C., Cheng, Y., Gan, Z., Yu, L., and Liu, J. (2020a). Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2046–2065.

Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., and Chang, K.-W. (2019a). Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Li, L. H., You, H., Wang, Z., Zareian, A., Chang, S.-F., and Chang, K.-W. (2021b). Unsupervised vision-and-language pre-training without parallel images and captions. In *NAACL*.

Li, X., Li, C., Xia, Q., Bisk, Y., Celikyilmaz, A., Gao, J., Smith, N. A., and Choi, Y. (2019b). Robust navigation with language pretraining and stochastic sampling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1494–1499.

Li, X., Yin, X., Li, C., Hu, X., Zhang, P., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. (2020b). Oscar: Object-semantics aligned pre-training for vision-language tasks. *arXiv preprint arXiv:2004.06165*.

Li, Y., Li, Y., and Vasconcelos, N. (2018). Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528.

Li, Z., Wallace, E., Shen, S., Lin, K., Keutzer, K., Klein, D., and Gonzalez, J. E. (2020c). Train large, then compress: Rethinking model size for efficient training and inference of transformers. In *ICML*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Liu, B., Huang, Z., Zeng, Z., Chen, Z., and Fu, J. (2019a). Learning rich image region representation for visual question answering. *arXiv preprint arXiv:1910.13077*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I. and Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Lu, J., Yang, J., Batra, D., and Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297.

Lu, K., Grover, A., Abbeel, P., and Mordatch, I. (2021). Pretrained transformers as universal computation engines. *arXiv preprint arXiv:2103.05247*.

Ma, C.-Y., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., and Xiong, C. (2019a). Self-monitoring navigation agent via auxiliary progress estimation. In *ICLR*.

Ma, C.-Y., Wu, Z., AlRegib, G., Xiong, C., and Kira, Z. (2019b). The regretful agent: Heuristic-aided navigation through progress estimation. In *CVPR*.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge university press.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2017). Pointer sentinel mixture models. In *ICLR*.

Miech, A., Alayrac, J.-B., Smaira, L., Laptev, I., Sivic, J., and Zisserman, A. (2020). End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889.

Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., and Sivic, J. (2019). HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212.

Mussmann, S. and Ermon, S. (2016). Learning and inference via maximum inner product search. In *International Conference on Machine Learning*, pages 2587–2596.

Nguyen, K. and Daumé III, H. (2019). Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 684–695.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Papadimitriou, I. and Jurafsky, D. (2020). Learning music helps you read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6829–6839.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. (2018). Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2015). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2017). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 123(1):74–93.

Pont-Tuset, J., Uijlings, J., Changpinyo, S., Soricut, R., and Ferrari, V. (2019). Connecting vision and language with localized narratives. *arXiv preprint arXiv:1912.03098*.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020a). Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Qi, Y., Wu, Q., Anderson, P., Wang, X., Wang, W. Y., Shen, C., and Hengel, A. v. d. (2020b). Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991.

Qian, R., Meng, T., Gong, B., Yang, M.-H., Wang, H., Belongie, S., and Cui, Y. (2021). Spatiotemporal contrastive video representation learning. In *CVPR*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021a). Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021b). Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2015). Sequence level training with recurrent neural networks.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Roy, D. K. and Pentland, A. P. (2002). Learning words from sights and sounds: A computational model. *Cognitive science*, 26(1):113–146.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *ACL*.

Shah, M., Chen, X., Rohrbach, M., and Parikh, D. (2019). Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. (2018). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*.

Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.-W., Yao, Z., and Keutzer, K. (2021). How much can clip benefit vision-and-language tasks?

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, N. A. (2019). Contextual word representations: A contextual introduction. *arXiv preprint arXiv:1902.06006*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Soomro, K., Zamir, A., and Shah, M. (2012a). Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402.

Soomro, K., Zamir, A. R., and Shah, M. (2012b). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., and Dai, J. (2020). Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*.

Suhr, A., Zhou, S., Zhang, I., Bai, H., and Artzi, Y. (2019). A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. (2019). Videobert: A joint model for video and language representation learning. *arXiv preprint arXiv:1904.01766*.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Tan, H. and Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5103–5114.

Tan, H. and Bansal, M. (2020). Vokenization: Improving language understanding with contextualized, visual-grounded supervision. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Tan, H., Lei, J., Wolf, T., and Bansal, M. (2021). Vimpac: Video pre-training via masked token prediction and contrastive learning. *In Submission*.

Tan, H., Yu, L., and Bansal, M. (2019). Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2610–2621.

Tang, Z., Cho, J., Tan, H., and Bansal, M. (2021). Vidlankd: Improving language understanding viavideo-distilled knowledge transfer. *In Submission*.

Thomason, J., Murray, M., Cakmak, M., and Zettlemoyer, L. (2020). Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. (2016). Yfcc100m: The new data in multimedia research. *Communications of the ACM*.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. (2020). What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2017). Neural discrete representation learning. In *NeurIPS*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Walker, J., Razavi, A., and Oord, A. v. d. (2021). Predicting video with vqvae. *arXiv preprint arXiv:2103.01950*.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2019a). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Wang, L., Koniusz, P., and Huynh, D. Q. (2019b). Hallucinating idt descriptors and i3d optical flow features for action recognition with cnns. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8698–8708.

Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.-F., Wang, W. Y., and Zhang, L. (2019c). Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *CVPR*.

Wang, X., Wu, J., Chen, J., Li, L., Wang, Y.-F., and Wang, W. Y. (2019d). Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4581–4591.

Wang, X., Xiong, W., Wang, H., and Yang Wang, W. (2018). Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *ECCV*.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wurtz, R. H., Kandel, E. R., et al. (2000). Central visual pathways. volume 4, pages 523–545.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Xu, J., Mei, T., Yao, T., and Rui, Y. (2016). Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. (2018). Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324.

Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., and Zhou, M. (2020). Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Yang, Z., He, X., Gao, J., Deng, L., and Smola, A. (2016). Stacked attention networks for image question answering. In *CVPR*.

Yao, B. Z., Yang, X., Lin, L., Lee, M. W., and Zhu, S.-C. (2010). I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78.

Yu, Z., Cui, Y., Yu, J., Tao, D., and Tian, Q. (2019a). Multimodal unified attention networks for vision-and-language interactions. *arXiv preprint arXiv:1908.04107*.

Yu, Z., Yu, J., Cui, Y., Tao, D., and Tian, Q. (2019b). Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6281–6290.

Yu, Z., Yu, J., Xiang, C., Fan, J., and Tao, D. (2018). Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12):5947–5959.

Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. (2019). From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. (2018). Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Zhang, H., Koh, J. Y., Baldridge, J., Lee, H., and Yang, Y. (2021a). Cross-modal contrastive learning for text-to-image generation. *arXiv preprint arXiv:2101.04702*.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915.

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., and Gao, J. (2021b). Vinvl: Revisiting visual representations in vision-language models. *arXiv preprint arXiv:2101.00529*.

Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., and Langlotz, C. P. (2020). Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*.

Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J., and Gao, J. (2020). Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049.

Zhu, F., Zhu, Y., Chang, X., and Liang, X. (2020). Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10012–10022.

Zhu, L. and Yang, Y. (2020). Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8746–8755.

Zhu, Y., Groth, O., Bernstein, M., and Fei-Fei, L. (2016). Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.