Unifying Vision-and-Language Tasks via Text Generation

Jaemin Cho **Jie Lei** **Hao Tan** **Mohit Bansal**
UNC Chapel Hill
{jmincho,jielei,haotan,mbansal}@cs.unc.edu

Abstract

Existing methods for vision-and-language learning typically require designing task-specific architectures and objectives for each task. For example, a multi-label answer classifier for visual question answering, a region scorer for referring expression comprehension, and a language decoder for image captioning, etc. To alleviate these hassles, in this work, we propose a unified framework that learns different tasks in a single architecture with the same language modeling objective, i.e., multimodal conditional text generation, where our models learn to *generate labels in text* based on the visual and textual inputs. On 7 popular vision-and-language benchmarks, including visual question answering, referring expression comprehension, visual commonsense reasoning, most of which have been previously modeled as discriminative tasks, our generative approach (with a single unified architecture) reaches comparable performance to recent task-specific state-of-the-art vision-and-language models. Moreover, our generative approach shows better generalization ability on answering questions that have rare answers. In addition, we show that our framework allows multi-task learning in a single architecture with a single set of parameters, which achieves similar performance to separately optimized single-task models.[1]

1. Introduction

Mirroring the success of the pretraining-finetuning paradigm with transformer language models (Devlin et al., 2019), recent vision-and-language transformers (Tan & Bansal (2019); Lu et al. (2019); Chen et al. (2020); Li et al. (2020b), *inter alia*) have also been adopted in a wide range of vision-and-language tasks. These models are first pretrained on the large image-text corpus (e.g., COCO Caption (Chen et al., 2015)), then finetuned on downstream tasks (e.g., visual question answering (Goyal et al., 2019) and referring expression comprehension (Mao et al., 2016)), which outperformed many previous non-pretraining-finetuning methods.

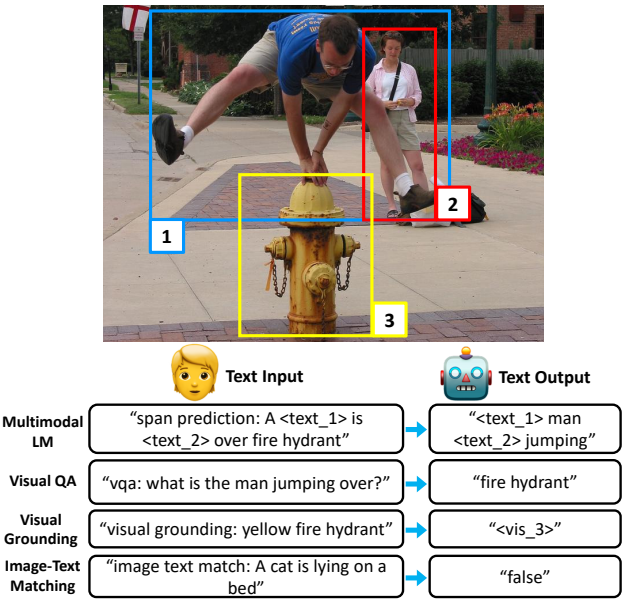


Figure 1. Our unified framework for learning vision-and-language tasks. While existing methods require designing task-specific architectures for different tasks, our framework unifies them together as generating text labels conditioned on multimodal inputs.

For each pretraining or downstream task, existing vision-and-language transformers typically require designing task-specific, separately-parameterized architectures on top of the transformer encoder (e.g., multi-label sigmoid classifier for visual question answering, and softmax classifier for referring expression comprehension). However, the reasoning skills required by these tasks overlap significantly. Consider the example in Fig. 1, answering the question "What is the man jumping over?" and grounding an image region corresponding to the referring phrase "yellow fire hydrant". Both require models to recognize the object "fire hydrant".

In addition, the labels for these tasks can be easily expressed in text. For instance, we can assign a region id (e.g., "<vis_3>", a special text token) to a specific region

[1]Our code will be publicly available at: https://github.com/j-min/VL-T5

in the image, and then the referring expression comprehension task can be expressed as generating the correct region id. For visual question answering, the labels are already in text, although existing approaches (Anderson et al., 2018; Tan & Bansal, 2019; Chen et al., 2020) tackle the task as learning a multi-label classifier over a fixed set of frequent answers (See Fig. 3).

Hence, in order to alleviate these hassles of designing task-specific architectures, we propose a unified framework for vision-and-language learning via *generating labels in text*. Specifically, we extend off-the-shelf pretrained language models T5 (Raffel et al., 2019) and BART (Lewis et al., 2020) with visual understanding ability, named 'VL-T5' and 'VL-BART'. In contrast to existing vision-and-language transformers which train different architectures for different pretraining and downstream tasks, our models tackle all the tasks with the same language modeling head. *To learn a new task, we can simply rewrite its input and output in text*, without the need of adding extra parameters or designing new architectures and objectives. This enables our models to adapt to different tasks easily. In addition, *we can leverage the text generation ability of pretrained language models when making predictions*. This is especially helpful when we answer open-ended questions that require non-trivial answers, where discriminative methods can only answer from a predefined set of frequent candidates, while our models can generate open-ended natural language answers.

To evaluate the effectiveness of our generative modeling approach, we compare our models against recent vision-and-language transformers on a diverse set of 7 downstream benchmarks, including visual question answering on VQA (Goyal et al., 2019) and GQA (Hudson & Manning, 2019), referring expression comprehension on RefCOCOg (Mao et al., 2016), natural language visual reasoning on NLVR^2(Suhr et al., 2019), visual commonsense reasoning on VCR (Zellers et al., 2019), image captioning on COCO Caption (Chen et al., 2015), and multimodal machine translation on Multi30K (Elliott et al., 2016). Our unified generative method reaches comparable performance to recent state-of-the-art vision-and-language pretraining methods. This is especially interesting because we use the same unified language modeling architecture with the same maximum likelihood estimation (MLE) objective for all the tasks, while existing approaches use heavily-engineered task-specific architectures and objective functions. In addition, we found that our generative models have better generalization ability compared to the discriminative versions on the rare-answer scenario in visual question answering, when ground truth answers for given questions are rarely seen during training. Finally, we also experiment with our unified framework which allows for multi-task learning via a single set of parameters. Our multi-task model is jointly finetuned with visual question answering and referring comprehension expression tasks and achieves similar performance to single-task models finetuned with separate parameters.

2. Related Works

Vision-and-Language pretraining Large-scale language pretraining with transformers (Vaswani et al., 2017; Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020; Clark et al., 2020; Yang et al., 2019; Raffel et al., 2019) have achieved remarkable success for a spectrum of natural language understanding tasks (Rajpurkar et al., 2016; Zellers et al., 2018; Wang et al., 2018; Williams et al., 2017). Following this success, in the vision-and-language domain, image+text pretraining models (Lu et al., 2019; Tan & Bansal, 2019; Chen et al., 2020; Huang et al., 2020; Li et al., 2020b; Cho et al., 2020; Radford et al., 2021) and video+text pretraining models (Sun et al., 2019b;a; Li et al., 2020a; Zhu & Yang, 2020; Miech et al., 2020) have also shown to perform better than previous approaches (Yu et al., 2018a; Anderson et al., 2018; Kim et al., 2018; Yu et al., 2018b) without such pretraining, in a wide range of discriminative tasks (Goyal et al., 2019; Hudson & Manning, 2019; Lei et al., 2018; Mao et al., 2016; Xu et al., 2016; Zhou et al., 2018) and generative tasks (Chen et al., 2015; Xu et al., 2016; Zhou et al., 2018). In this work, we focus on image+text tasks.

Existing image+text models encode an image as a set of bounding box region features (Lu et al., 2019; Tan & Bansal, 2019; Chen et al., 2020) or grid features (Huang et al., 2020; Cho et al., 2020), analogous to text embeddings. Some works study using better image encoder (Zhang et al., 2021), or using objects tags as additional text input (Li et al., 2020b). These improvements on stronger visual and text input representations are orthogonal to ours. We expect that our models can benefit from using these stronger input representations.

Common pretraining objectives include (i) multimodal masked language modeling (Lu et al., 2019; Tan & Bansal, 2019; Chen et al., 2020; Huang et al., 2020; Li et al., 2020b): predict masked words conditioned on the input image and neighboring text context; and (ii) image text matching (Lu et al., 2019; Tan & Bansal, 2019; Chen et al., 2020; Huang et al., 2020): predict whether an input sentence matches with the input image. Besides these two objectives, we also use visual question answering, visual grounding, and grounded captioning as additional tasks for pretraining.

For each pretraining and downstream task, existing approaches typically train separately-parameterized task-specific architecture along with the transformer backbone. Though these pretrained models use shared encoders across multiple tasks, the output layers for the downstream tasks, e.g., visual question answering (Goyal et al., 2019; Hudson

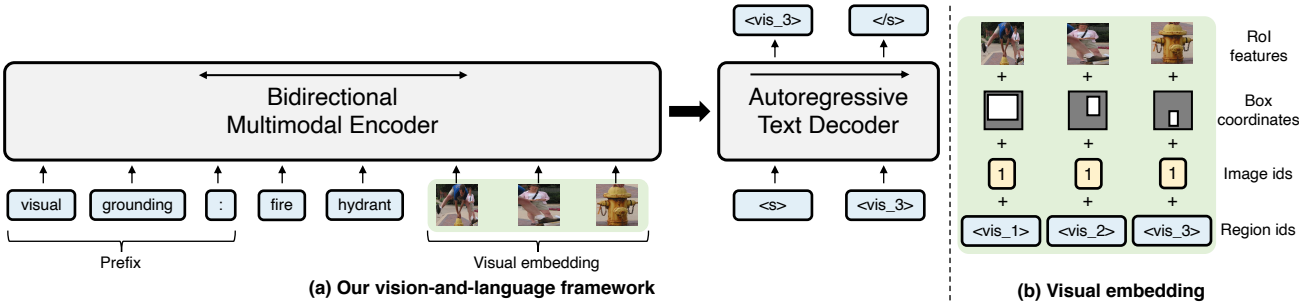


Figure 2. An illustration of our VL-T5 and VL-BART architectures for visual grounding task. Instead of task-specific architectures, our models use text prefixes to adapt to different tasks. The green block in (a) refers to visual embeddings. (b) shows the components of visual embedding. Note that we reuse the text embeddings of visual sentinel tokens (ex. `<vis_3>`) as region id embeddings, which allows our models to tackle many discriminative vision-language tasks as text generation, including visual grounding.

& Manning, 2019; Lei et al., 2018), referring expression comprehension (Kazemzadeh et al., 2014; Yu et al., 2016; Mao et al., 2016), and image captioning (Chen et al., 2015), are significantly different. For example, UNITER (Chen et al., 2020) uses a multi-label sigmoid classifier to regress soft answer scores for visual question answering (Goyal et al., 2019), while using softmax classifier head on object representations for referring expression comprehension (Mao et al., 2016). In contrast, our method casts all pretraining and downstream tasks as text generation and always uses the same unified architecture, i.e., language modeling head, alleviating the hassle of manually designing task-specific architectures.

Unified frameworks One line of work focus on solving natural language processing tasks in a unified format, as question answering (Mccann et al., 2018), span prediction (Keskar et al., 2019), or text generation (Raffel et al., 2019; Brown et al., 2020; Khashabi et al., 2020). These unified frameworks provide efficient knowledge sharing among different tasks and make it easy to leverage pretrained language models. In relation to these works, we propose to unify previously separately modeled vision-and-language tasks in a single unified format, via text generation, conditioned on multimodal inputs from the image and the textual context.

3. Model

We propose a new learning method that unifies vision-and-language problems as multimodal conditional text generation. We introduce VL-T5 and VL-BART based on two pretrained sequence-to-sequence transformer language models: T5$_{Base}$ (Raffel et al., 2019) and BART$_{Base}$ (Lewis et al., 2020). Specifically, we extend their text encoders to multimodal encoders by incorporating image region embeddings as additional input. The overall architecture of our framework is shown in Fig. 2. Since the architecture differences between VL-T5 and VL-BART are minor, we will use VL-T5 as an example to illustrate our framework in details in the rest of this section.

3.1. Visual Embeddings

We represent an input image v with n object regions from object detector. Following previous works, we use the Faster R-CNN (Ren et al., 2015) trained on Visual Genome (Krishna et al., 2016) for object and attribute classification, provided by Anderson et al. (2018). Following Tan & Bansal (2019), we use n=36 object regions per image.

As shown in Fig. 2 (*b*), each image region is encoded as a sum of four types of features: (i) RoI (Region of Interest) object features; (ii) RoI bounding box coordinates; (iii) image ids $\in \{1, 2\}$; and (iv) region ids $\in \{1, \ldots, n\}$. RoI features and bounding box coordinates are encoded with a linear layer, while image ids and region ids are encoded with learned embeddings (Devlin et al., 2019). Image ids are used to discriminate regions from different images, and take effect only when multiple images are given to the model (e.g., in NLVR^2 (Suhr et al., 2019), models take two input images). The final visual embeddings are denoted as e^v=$\{e^v_1, \ldots, e^v_n\}$. These embeddings have the same dimension as the text embeddings that we will discuss next.

3.2. Text Embeddings

Instead of designing task-specific architectures, we add different prefixes to the original input text to adapt to different tasks, as shown in Table. 1 (*top*). We show the prefixes for different tasks in Table 1.[2]

Input text x is tokenized as $\{x_1, \ldots, x_{|x|}\}$ and encoded as learned embedding $e^x = \{e^x_1, \ldots, e^x_{|x|}\}$. The embedding

[2]Note that since we use simple prefixes (e.g., "`vqa :`" for VQA task), it is likely that engineering in text prompts (Gao et al., 2020) would improve the accuracy of our methods. As this is not the focus of this paper, we leave it as future works.

parameters are shared by the encoder, decoder, and language modeling head (Press & Wolf, 2017). Since the attention layers are permutation-invariant, BART learns positional embedding (Vaswani et al., 2017; Devlin et al., 2019) for each absolute text position and adds them to the text token embeddings. In contrast, T5 adds relative position bias to each self-attention layer (Shaw et al., 2018). Our models follow the positional embedding configurations of their text backbone models. At the same time, we use bounding box coordinates to provide position information for visual embeddings, similar to absolute position embeddings for text.

In addition to the original vocabulary of T5 and BART, we introduce visual sentinel tokens {<vis_1>, ..., <vis_n>}, which corresponds to image regions. As illustrated in Fig. 2, we use the text embeddings of visual sentinel tokens as region id embeddings in Sec. 3.1. The embedding sharing enables our model to build the correspondence among query text, label text, and objects, which are useful in the grounding tasks (e.g., visual grounding and grounded captioning pretraining tasks in Sec. 4, referring expression comprehension in Sec. 5.3).

3.3. Encoder-Decoder Architecture

We use transformer encoder-decoder architecture (Vaswani et al., 2017) to encode visual and text inputs and generate label text. Our bidirectional multimodal encoder is a stack of m transformer blocks. Each transformer block consists of a self-attention layer and a fully-conencted layer with additional residual connections. As shown in Fig. 2 (*a*), the encoder takes the concatenation of text embeddings and visual embeddings as input and outputs their contextualized joint representations $h = \{h_1^x, \ldots, h_{|x|}^x, h_1^v, \ldots, h_n^v\} = \text{Enc}(e^x, e^v)$. As discussed in Sec. 3.2, T5 model takes relative positional embeddings by adding relative position bias. Since there is no natural order between image regions, we set relative position bias over visual inputs as zeros for VL-T5. At the same time, the features of RoI bounding box coordinates in the visual embeddings serve as positional embedding for the visual object regions.

Our decoder is another stack of m transformer blocks similar to the multimodal encoder. However, each block has an additional cross attention layer. For autoregressive generation, the decoder's inputs are shifted right, and a *start-of-sequence* token <s> is added in the beginning and used as the decoder's initial input token y_0. Likewise, an *end-of-sequence* token </s> is appended to the end of decoder outputs to indicate the completion of generation. The decoder iteratively attends to previously generated tokens $y_{<j}$ (via self-attention) and the encoder outputs h (via cross-attention), then predicts the probability of future text tokens $P_\theta(y_j|y_{<j}, x, v) = \text{Dec}(y_{<j}, h)$. For both pretraining

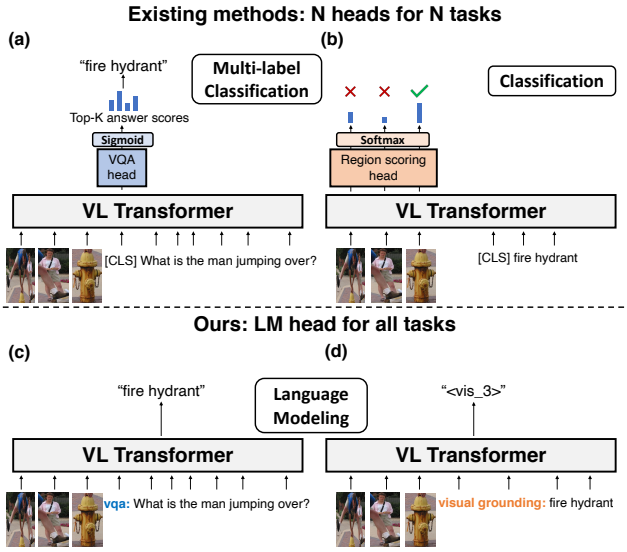


Figure 3. Comparison between existing vision-language transformers and our framework on visual question answering and referring expression comprehension (visual grounding) tasks. While existing methods use task-specific architectures and objectives, our models use language modeling head and maximum likelihood estimation on label text for all tasks.

(Sec. 4) and downstream tasks (Sec. 5), we train our model parameters θ by minimizing the negative log-likelihood of label text y tokens given input text x and image v (Eq. 1).

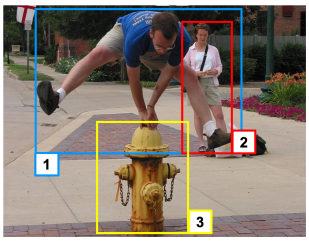
$$\mathcal{L}_\theta^{\text{GEN}} = -\sum_{j=1}^{|y|} \log P_\theta(y_j|y_{<j}, x, v) \tag{1}$$

3.4. Task-Specific Methods *vs.* Our Unified Framework

In this subsection, we compare our unified framework with existing vision-and-language transformers on two popular tasks: visual question answering (Goyal et al., 2019) and referring expression comprehension (Mao et al., 2016). We illustrate this comparison in Fig. 3.

Visual question answering task requires a model to answer a question to a given context image. As shown in Fig.3 (*a*), existing methods (Tan & Bansal, 2019; Lu et al., 2019; Chen et al., 2020) typically formulate this task as a discriminative task, i.e., multi-label classification over a predefined set of K frequent answer candidates $\{a^1, \ldots, a^K\}$. Specifically, they introduce a multi-layer perceptron (MLP) sigmoid scorer head on top of $h^x_{\texttt{[CLS]}}$ to learn the likelihood of each answer candidate being correct: $P_\theta^{\text{VQA}}(\text{correct}|a, x, v) = \text{sigmoid}(\text{MLP}^{\text{VQA}}(h^x_{\texttt{[CLS]}}))$. This VQA scorer head is trained end-to-end with the transformer encoder through a binary cross-entropy loss, by using VQA

Table 1. Input-output formats for pretraining (Sec. 4) and downstream tasks (Sec. 5). [a]We use different prefixes ("vqa:", "gqa:", "visual7w:") for questions from different datasets. [b]NLVR^2 takes two images as visual input, for brevity, we only show one here.

Tasks	Input image	Input text	Target text
Pretraning tasks (Sec. 4)			
Multimodal LM (VL-T5)		span prediction: A <text_1> is <text_2> over a fire hydrant.	<text_1> man <text_2> jumping
Multimodal LM (VL-BART)		denoise: A <mask> is <mask> over a fire hydrant.	A man is jumping over a fire hydrant
[a]Visual question answering		vqa: what is the color of the man's shirt?	blue
Image-text matching		image text match: A man with blue shirt is jumping over fire hydrant.	true
Visual grounding		visual grounding: yellow fire hydrant	<vis_3>
Grounded captioning		caption region: <vis_3>	yellow fire hydrant
Downstream tasks (Sec. 5)			
VQA		vqa: [Q]	[A]
GQA		gqa: [Q]	[A]
[b]NLVR^2		nlvr: [text]	true/false
VCR Q→A		vcr qa: question [Q] answer: [A]	true/false
VCR Q→AR		vcr qar: question [Q] answer: [A] rationale: [R]	true/false
RefCOCOg		visual grounding: [referring expression]	[region id]
COCO captioning		caption:	[caption]
COCO captioning (w/ object tags)		caption with tags: [Tag1 Tag2 ..]	[caption]
Multi30K En-De translation		translate English to German: [English text]	[German text]

score (Goyal et al., 2019)[3] as soft target distribution (Eq. 2).

$$\mathcal{L}_\theta^{\text{VQA}} = -\sum_{k=1}^{K} \text{score}(a^k, x, v) \log P_\theta^{\text{VQA}}(\text{correct}|a^k, x, v) \tag{2}$$

For referring expression comprehension, it requires models to localize a target region in an image that is described by a given referring expression. Previous methods tackle this task as multi-class (Chen et al., 2020) or binary (Lu et al., 2019) classification over image regions. For example, UNITER (Chen et al., 2020) introduces a region scoring head (an MLP layer) on top of the output representations of regions, as shown in Fig. 3(*b*). This region scoring head is jointly trained with the encoder by minimizing the negative log-likelihood of the target region r^*:

$$\mathcal{L}_\theta^{\text{REF}} = -\log P_\theta^{\text{REF}}(r^*|x, v) \tag{3}$$

In contrast to existing methods that develop task-specific architectures and objectives (e.g., Eq. 2, 3), our unified framework is free from extra model designs for new tasks. As shown in Fig. 3 (*c*,*d*) and Table 1, we formulate the task labels to corresponding text, and we learn these different tasks by predicting label text with the same language modeling objective (Eq. 1).

4. Pretraining

In this section, we describe how we pretrain our VL-T5 and VL-BART models (Sec. 3). We start with the details of the pretraining data and illustrate how we formulate diverse vision-and-language pretraining tasks as multimodal conditional text generation.

[3]$\text{score}(a, x, v) = \min((\#\text{humans that provided } a \text{ as the answer}) * 0.3, 1)$

4.1. Pretraining Data

We aggregate pretraining data from MS COCO (Lin et al., 2014; Chen et al., 2015) and Visual Genome (VG; Krishna et al. (2016)) images [4]. The captioning data from these two datasets are used in the multimodal language modeling task. The COCO captions are also used in the image-text matching task to learn cross-modal alignment. Besides the captions, we also use three visual question answering datasets (VQA v2.0 Goyal et al. (2019), GQA balanced version (Hudson & Manning, 2019), and Visual7W (Zhu et al., 2016)) as in Tan & Bansal (2019), but only used them in the visual question answering task. Details of these pretraining tasks are in Sec. 4.2.

Overall, our pretraining dataset contains 9.18M image-text pair on 180K distinct images. We carefully split our pretraining data to avoid any intersection between our training data and the evaluation set of downstream tasks (e.g., COCO Captioning, RefCOCOg). In this process, around 10K images are excluded from the training sets of COCO and VG. We then take the COCO *Karpathy val split* (Karpathy & Fei-Fei, 2015) with 5,000 images as our validation set to monitor pretraining performance.

4.2. Pretraining Tasks

We pretrain our models under a multi-task setup with diverse pretraining tasks, including multimodal language modeling, visual question answering, image-text matching, visual grounding, and grounded captioning. Table 1 shows input and output examples of our pretraining tasks. The training data for each of these tasks are summarized in Table 11. In the rest of this section, we explain these tasks in detail.

[4]Existing vision-and-language transformers are trained with different datasets and computational budget, thus their results may not be directly comparable to each other. We show the number of their pretraining images in Table 2.

Multimodal language modeling We follow Raffel et al. (2019) and Lewis et al. (2020) to construct the language modeling pretraining task. The basic idea is to recover the masked input text based on both visual and textual context (while original methods are only based on textual context). For VL-T5, we mask 15% of input text tokens and replace contiguous text span with sentinel tokens (e.g., <text_1>). Then we let the model predict the masked text spans. For VL-BART, we mask 30% of input text tokens with <mask> tokens, and let the model reconstruct the entire original text. See Table 1 for examples.

Visual question answering Similar to Tan & Bansal (2019), we include visual question answering in our pretraining tasks. The task requires models to answer a question to a given context image. While previous methods (Tan & Bansal, 2019; Lu et al., 2019; Chen et al., 2020) tackle the task as classification over predefined answer candidates (illustrate in Fig. 3), we directly generate answers in their original text format.

Image-text matching In this task, the model needs to verify whether an input text corresponds to the given input image. We consider the image and its captions[5] as positive pairs. With a probability of 50%, we create a negative pair by randomly sampling another image from training set and taking its caption. The model then predicts the correspondence between the input image and text with "true" or "false" as shown in Table 1.

Visual grounding Besides the above image-text matching task, we also develop an object-text matching task to endow the model with grounding ability, which is required in several tasks (e.g., referring expression comprehension and VCR). Previous vision-and-language transformers (Tan & Bansal, 2019; Lu et al., 2019; Chen et al., 2020) predict the property of masked objects to indirectly learn object-text alignment. To explicitly learn this important grounding ability, we give the model a region description and let it predict the id of the related object region. With the help of the visual sentinel token (e.g., <vis_3> in Table 1), this task fits naturally into our text generation objective. We make the region descriptions from the predictions of the object detector that we use for visual embeddings (see Sec. 3.1). Concretely, we sample an object region out of n region predictions. Then we concatenate its object name and attribute in their original text format (e.g., attribute: "yellow" + object: "fire hydrant" $\rightarrow$ "yellow fire hydrant").[6] This approach does not need extra annotation and could be extended to images without dense annotations (e.g., COCO images).

Grounded captioning To teach the model with object-level information, we also use an inverse task of the aforementioned visual grounding, called grounded captioning. As shown in Table 1, given a visual sentinel token (which indicates a region in the image) as text input, the model is asked to generate a corresponding textual description of this input region.[7]

4.3. Pretraining Implementation Details

For both VL-T5 and VL-BART, it takes 4 days for 30-epoch pretraining with mixed precision training (Narang et al., 2018) on 4 RTX 2080 Ti GPUs (4 x 11GB). We use batch size 320 and 600 for VL-T5 and VL-BART, respectively. We use AdamW optimizer (Loshchilov & Hutter, 2019) with $(\beta^1, \beta^2) = (0.9, 0.999)$ and learning rate 1e-4 with 5% linear warmup schedule. We use the VQA validation score to track the progress of pretraining. Our code is based on PyTorch (Paszke et al., 2017) and Huggingface Transformers (Wolf et al., 2019).

5. Downstream Tasks and Results

In this section, we evaluate our generative architectures VL-T5 and VL-BART on a diverse set of 7 downstream tasks, including two image question answering tasks (Goyal et al., 2019; Hudson & Manning, 2019), referring expression comprehension (Mao et al., 2016), natural language visual reasoning (Suhr et al., 2019), visual commonsense reasoning (Zellers et al., 2019), image captioning (Chen et al., 2015), and multimodal machine translation (Elliott et al., 2016). We summarize the statistics of the datasets used in downstream tasks in Table 12. We compare our models with strong vision-and-language pretrained transformers: LXMERT (Tan & Bansal, 2019), ViLBERT (Lu et al., 2019), UNITER (Chen et al., 2020), Unified VLP (Zhou et al., 2020), Oscar (Li et al., 2020b), and XGPT (Xia et al., 2020).

As summarized in Table 2, our models achieve similar results to most of the baselines. We highlight that our unified generative modeling approach (with the input-output format shown in Table 1) is close to the performance of the heavily developed task-specific discriminative models. Note that different vision-and-language transformers are trained with different setups (e.g., pretraining data, objectives, feature extractor, hyperparameters, computational budget), thus the results might not be directly comparable. For exam-

[5] We only use captions from COCO for this task, since many short captions from VG and visual questions are nondistinctive description of an image (e.g., 'what is in the image?').

[6] https://github.com/peteanderson80/bottom-up-attention/blob/master/data/genome/1600-400-20

[7] Our grounded captioning task can be seen as a simplified dense captioning (Johnson et al., 2016) task, where only one object is asked to describe at a time.

Table 2. Single model performance on downstream tasks. Note that the baseline models adopt task-specific objectives and architectures, whereas our models tackle all tasks, including discriminative tasks (e.g., RefCOCOg), as text generation with a single architecture and objective. ⋆ See our discussion in Sec.5.3. †Submitted to the leaderboard (the result will be updated).

Method	# Pretrain Images	Discriminative tasks					Generative tasks	
		VQA test-std Acc	GQA test-std Acc	NLVR^2 test-P Acc	RefCOCOg test^d Acc	VCR Q→ AR test Acc	COCO Cap Karpathy test CIDEr	Multi30K En-De test 2018 BLEU
LXMERT	180K	72.5	60.3	74.5	-	-	-	-
ViLBERT	3M	70.9	-	-	-	54.8	-	-
UNITER$_{Base}$	4M	72.9	-	77.9	74.5	58.2	-	
Unified VLP	3M	70.7	-	-	-	-	117.7	-
Oscar$_{Base}$	4M	73.4	61.6	78.4	-	-	123.7	-
XGPT	3M	-	-	-	-	-	120.1	-
MeMAD	-	-	-	-	-	-	-	38.5
VL-T5	180K	70.3	60.8	73.6	71.3	58.9	116.5	38.6
VL-BART	180K	71.3	60.5	70.3	22.4⋆	-†	116.6	28.1

ple, UNITER and Oscar use around 4M extra images from SBU captions (Ordonez et al., 2011) and Conceptual Captions (Sharma et al., 2018) for pretraining. The closest baseline to our models is LXMERT as both are pretrained on the same datasets and use the same visual features. See Table 10 in the appendix for a detailed comparison between baselines and our models. We tune the hyperparameters based on the validation set of each downstream task. See Table 13 for details. In the rest of this section, we'll provide a detailed comparison w.r.t. our models and the baselines, as well as elaborating the details of the evaluated tasks.

5.1. Visual Question Answering: VQA and GQA

The visual question answering task requires models to answer a question to a given context image. In this work, we evaluate our models on VQA (Goyal et al., 2019) and GQA (Hudson & Manning, 2019) datasets. Each question in VQA and GQA typically have multiple answers, at each training step, we randomly sample one answer from the ground-truth answer set and use it as the text generation target.

Table 2 compares our models VL-T5 and VL-BART with existing methods on visual question answering tasks VQA and GQA. For both tasks, our models achieve comparable performance to existing approaches. Note that in addition to the Visual Genome and COCO Captions data that we use, UNITER and Oscar also use around 4M extra images from SBU captions (Ordonez et al., 2011), Conceptual Captions (Sharma et al., 2018), and Flicker30k (Young et al., 2014) (Oscar only) for pretraining. Chen et al. (2020) have shown that adding these extra data during pretraining improves model performance across various downstream tasks.

Table 3. VQA Karpathy-test split accuracy using generative and discriminative methods. We break down the questions into two subsets in terms of whether the best-scoring answer a^* for each question is included in the top-K answer candidates A^{topk}. *In-domain*: $a^* \in A^{topk}$, *Out-of-domain*: $a^* \notin A^{topk}$.

Method	VQA Karpathy-test Acc.		
	In-domain	Out-of-domain	Overall
Discriminative			
UNITER$_{\text{Base}}$	**74.4**	10.0	**70.5**
VL-T5	70.2	7.1	66.4
VL-BART	69.4	7.0	65.7
Generative			
VL-T5	71.4	13.1	67.9
VL-BART	72.1	**13.2**	68.6

Generative *vs.* Discriminative model Modern approaches (Tan & Bansal, 2019; Lu et al., 2019; Chen et al., 2020; Zhou et al., 2020; Li et al., 2020b) are discriminative models, where they tackle visual question answering tasks as multi-label classification over a predefined set of answer candidates. For example, LXMERT and UNITER train a two-layer MLP classifier with sigmoid activation and use soft target scores on 3,129 answers that appear 9 or more times in the VQA train2014 split (See Sec. 3.4 and Fig. 3). While this strategy has achieved strong performance, it may not generalize to a real-world scenario where answers may not exist in this fixed answer set. In contrast, our models VL-T5 and VL-BART directly generate answers as free-form text, allowing a truly open-ended setup.

To quantitatively compare existing discriminative ap-

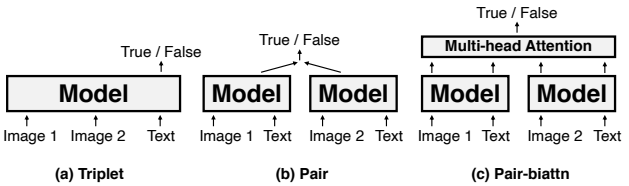


Figure 4. Different encoding settings for NLVR^2. *Pair* and *Pair-biattn* approximately double the computational cost over *Triplet* which our models are based on.

proaches and our generative approaches, we evaluate their performance on questions with rare answers, i.e., out-of-domain answers (for discriminative approaches). We break down VQA questions in Karpathy-test split, in terms of whether the best-scoring [8] answer a^* for each question is included in the top-K ($K = 3,129$) answer candidates A^{topk}. The questions with $a^* \notin A^{topk}$ can be treated as out-of-domain questions since their best-scoring answers are rare answers and have been excluded in standard discriminative approaches. After this split, the in-domain subset contains 24,722 questions, and the out-of-domain subset contains 1,558 questions. For discriminative baselines, we introduce a two-layer MLP classifier with sigmoid on top of the decoder representation of *start-of-sequence* token <s>, following LXMERT and UNITER.

In Table 3, we show the performance of the models on these two subsets. Comparing models with the same backbone using different modeling approaches, we notice the generative models improve upon the discriminative baselines across all the subsets. This improvement is more significant when looking at the out-of-domain subset, where the generative VL-T5 and VL-BART achieve 6 and 6.2 points improvement over their discriminative counterparts, showing the effectiveness of using generative modeling. When compared to the strong discriminative baseline UNITER$_{Base}$ (pretrained with 4M extra images), our generative models still show comparable overall performance while significantly outperform it on the out-of-domain subset.

5.2. Natural Language Visual Reasoning: NLVR^2

The task of NLVR^2 (Suhr et al., 2019) is to determine whether a natural language statement is true about two given images. To apply our model to this task, we concatenate region features from two images and use different image id embeddings to disambiguate which image the features are from. Then our model learns to generate "true" or "false". This is similar to *Triplet* (Fig. 4(a)) setting described in UNITER (Chen et al., 2020)

[8] The *best-scoring* answer is the ground-truth answer that has the best score according to the VQA scoring system.

Table 4. NLVR^2 performance comparison under different encoding settings. Note that *Triplet* takes lower computational cost than *Pair* and *Pair-biattn* (See also Fig. 4).

Method	Setting	dev	test-P
UNITER$_{Base}$	Triplet	73.0	73.9
UNITER$_{Base}$	Pair	75.9	75.8
UNITER$_{Base}$	Pair-biattn	**77.2**	**77.9**
LXMERT	Pair	74.9	74.5
Oscar$_{Base}$	Pair	78.1	78.4
VL-T5	Triplet	74.6	73.6
VL-BART	Triplet	71.7	70.3

Table 5. Referring expression comprehension performance comparison on RefCOCOg.

Method	V&L PT	val^d	test^d
MattNet		66.9	67.3
UNITER$_{Base}$	✓	**74.3**	**74.5**
VL-T5		63.4	62.9
VL-T5	✓	71.2	71.3
VL-BART		21.8	23.0
VL-BART	✓	23.6	22.4

Table 4 shows the performance of baselines and ours on NLVR^2 under different encoding settings (See Fig. 4): (a) *Triplet*: joint encoding of image pairs and text; (b) *Pair*: the concatenation of individual embedding of each image-text pair; (c) *Pair-biattn*: bidirectional attention added to *Pair*. UNITER shows that one can improve performance with a more complex encoding setting, i.e., *Pair-biattn* achieves better performance than *Pair*, which is again better than the simplest *Triplet*. Note that both the *Pair* and the *Pair-biattn* settings approximately double the computational cost compared to that of the *Triplet* setting. While there's the gap between our models and baselines in *Pair* and *Pair-biattn* setting, VL-T5 shows comparable performance to UNITER in *Triplet* setting.

5.3. Referring Expression Comprehension: RefCOCOg

Referring expression comprehension is a visual grounding task, where given a natural language referring expression (e.g., 'the car on the left') describing an object in an image, a model needs to correctly localize the object in this image (when object candidates are given, the task is reduced to choose an object from a set of candidates). In this work, we evaluate models on the RefCOCOg (Mao et al., 2016) dataset. Similar to the visual grounding pretraining task in Sec. 4, we give our model a referring phrase and candidate

Table 6. VCR accuracy. *Stage 1* refers to the original vision-and-language generic-domain pretraining and *Stage 2* refers to the in-domain pretraining on VCR.

Method	V&L PT		VCR val			VCR test		
	Stage 1	Stage 2	Q → A	QA → R	Q → AR	Q → A	QA → R	Q → AR
ViLBERT			69.3	71.0	49.5	-	-	-
ViLBERT	✓		72.4	74.4	54.0	73.3	74.6	54.8
UNITER$_{Base}$			72.4	73.7	53.5	-	-	-
UNITER$_{Base}$	✓		72.8	75.3	54.9	-	-	-
UNITER$_{Base}$	✓	✓	**74.6**	**77.0**	**57.8**	75.0	77.2	58.2
VL-T5			71.1	73.6	52.5	-	-	-
VL-T5	✓		72.9	75.0	54.7	-	-	-
VL-T5	✓	✓	**74.6**	**77.0**	**57.5**	**75.3**	**77.8**	**58.9**
VL-BART			65.4	68.1	44.6	-	-	-
VL-BART	✓		67.0	67.4	45.4	-	-	-
VL-BART	✓	✓	69.2	69.9	48.6	-	-	-

region features from the image, the model then generates the visual sentinel token (e.g., `<vis_1>`) of the region corresponding to the phrase. Following previous works UNITER and MAttNet (Yu et al., 2018a), we use detected regions from Mask R-CNN (He et al., 2017)[9] as candidates and mark a selected region to be correct if its intersection over union (IoU) with the ground truth region is greater than 0.5.

Table 5 compares our generative models with discriminative baselines. With pretraining, VL-T5 significantly outperforms the strong modular model MAttNet, and achieves a reasonable performance compared to the UNITER model that has been pretrained on a much larger corpus. While our method did not achieve state-of-the-art performance, these results suggest that referring expression comprehension/visual grounding can be effectively formulated as a text-generation task, rather than previously (Yu et al., 2018a; Chen et al., 2020) formulated classification task over a set of visual regions, allowing more flexible architecture design. We hope our work would inspire future works in this direction. We also observe that our experiments with VL-BART on RefCOCOg diverges. One reason might be the difference in positional encoding methods of T5 and BART. During training, BART adds learned absolute positional embedding to text token embedding, whereas T5 uses relative position biases in self-attention layers instead. We hypothesize that VL-BART found strong correspondence by memorizing the positions of each training object (we observe high training accuracy, but low validation accuracy). We are actively investigating this interesting phenomenon and looking for potential solutions.

[9] https://github.com/lichengunc/MAttNet#pre-computed-detectionsmasks

5.4. Visual Commonsense Reasoning: VCR

Visual Commonsense Reasoning (VCR) (Zellers et al., 2019) is a multiple-choice question answering task that requires commonsense reasoning beyond object or action recognition. Each VCR question (Q) has 4 answers (A) and 4 rationales (R), it can be decomposed into two multiple choices sub-tasks: question answering (Q→A), and answer justification (QA→R). The overall task (Q→AR) requires a model to not only select the correct answer to the question, but also the correct rationale for choosing the answer. Similar to Nogueira et al. (2020) that leverages language model for document ranking, we concatenate context (image+question) with each candidate choice, and let our models generate "true" for the correct choice and generate "false" otherwise. As shown in Table 1, for Q→A, we use "vcr qa: question: `[Q]` answer: `[A]`" as text input. For QA→R, we use "vcr qar: question: `[Q]` answer: `[A]` rationale: `[R]`" as text input. During inference, we use $\frac{P(true)}{P(true)+P(false)}$ to rank the choices and select the one with the highest score as prediction. UNITER (Chen et al., 2020) has shown that a second-stage in-domain pretraining (with the same pretraining objectives as generic-domain pretraining) on the VCR dataset would significant improve VCR task performance. This is likely due to the domain difference between VCR and the generic-domain pretraining corpus (e.g., COCO Captions), e.g., the input text (concatenation of multiple sentences: `[Q]+[A]+[R]`) in VCR is much longer than in generic-domain pretraining. We thus also experimented with a second stage pretraining on VCR.

The results are shown in Table 6. On the VCR test split, We notice that our best model VL-T5 achieves a comparable (slightly better) performance to UNITER, while significantly higher performance when compared to ViLBERT.

Table 7. COCO captioning scores on Karparthy-test split. All models are trained with cross-entropy loss. PT and FT refer to the use of object tags during pretraining and finetuning, respectively.

Method	V&L PT	Object tags	COCO Captioning			
			B	C	M	S
Oscar	✓	PT+FT	**36.5**	**123.7**	**30.3**	**23.1**
VL-T5	✓	FT	34.5	116.5	28.7	21.9
VL-BART	✓	FT	35.1	116.6	28.7	21.5
Oscar	✓		34.5	115.6	**29.1**	**21.9**
Unified VLP	✓		36.5	117.7	28.4	21.3
XGPT	✓		**37.2**	**120.1**	28.6	21.8
VL-T5	✓		34.6	116.1	28.8	**21.9**
VL-BART	✓		34.2	114.1	28.4	21.3
Unified VLP			35.5	**114.3**	28.2	21.0
XGPT			34.4	113.0	27.8	20.8
BUTD			**36.2**	113.5	27.0	20.3
VL-T5			32.6	109.4	28.2	21.0
VL-BART			33.8	112.4	**28.5**	**21.4**

Table 8. Multi30K En-De multimodal translation BLEU scores. † and * refer to data augmentation and ensemble, respectively. We use gray color for the ensemble model it is not fairly comparable.

Method	V&L PT	test2016	test2017	test2018
MSA		38.7	-	-
MeMAD		38.9	32.0	-
MSA†		39.5	-	-
MeMAD†		45.1	40.8	-
MeMAD†*		45.5	41.8	38.5
T5 (text only)		44.6	41.6	39.0
VL-T5		45.3	**42.4**	**39.5**
VL-T5	✓	**45.5**	40.9	38.6
BART (text only)		41.2	35.4	33.3
VL-BART		41.3	35.9	33.2
VL-BART	✓	37.7	29.7	28.1

On the VCR val split, comparing to the model variants that adapt different pretraining strategies, we find that both Stage 1 generic-domain pretraining and Stage 2 in-domain pretraining help improve the VCR task performance, which is consistent with the findings in UNITER.

5.5. Image Captioning: COCO Caption

We evaluate automatic caption generation performance on MS COCO Caption dataset (Chen et al., 2015). We use *Karparthy split* (Karpathy & Fei-Fei, 2015), which re-splits train2014 and val2014 COCO images (Lin et al., 2014) into 113,287 / 5000 / 5000 for train / validation / test. While some methods use reinforcement learning-based optimization on CIDEr, we only compare with methods using cross-entropy loss. Note that image captioning is the only task in our experiments that do not have meaningful textual contexts, which results in a notable difference in pretraining and finetuning w.r.t. the input format. Inspired by Oscar (Li et al., 2020a), we also experimented with using object tags as additional text inputs during finetuning. We use BLEU (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), METEOR (Banerjee & Lavie, 2005), SPICE (Anderson et al., 2016) as evaluation metrics using COCOEvalCap implementation.[10]

In Table 7, we compare our models with baselines in different settings: use of vision-and-language pretraining and use of object tag as additional text inputs. With and without vision-and-language pretraining, our models show comparable performance to baselines. Since the use of object tags requires significant extra computation, we only use it for finetuning. Using tags gives a comparable or slightly improved performance for both models, and the improvement is significant (2.5) in CIDEr for VL-BART. We expect tag-augmented pretraining like Oscar would further boost the performance of our models.

5.6. Multimodal Machine Translation: Multi30K

We evaluate English to German multimodal machine translation performance on Multi30K dataset (Elliott et al., 2016), which have been used in WMT multimodal machine translation shared tasks (Barrault et al., 2018). Multi30K dataset is collected by translating the Flickr30K (Young et al., 2014) dataset (in English) with paired German sentences. We report BLEU score using SacreBLEU (Post, 2018) implementation[11], which produces official WMT BLEU scores. Since no pretrained vision-and-language transformers have been evaluated on the multimodal machine translation task yet, we compare our models with state-of-the-art transformer models: Multimodal self-attention (MSA) (Yao & Wan, 2020), MeMAD (Grönroos et al., 2018).

Table 8 shows that our T5-based models outperformed all single-model baselines on all three test splits of Multi30K, without strong data augmentation (e.g., back-translation, captions from external image captioning model). Our vision-and-language models outperformed their original text-only backbones, but we did not observe notable improvement with vision-and-language pretraining. Vision-and-language pretraining degraded performance of VL-BART. We conjecture the reasons as (i) the source text in Multi30K contains sufficient information for machine translation without visual inputs as discussed in Caglayan et al. (2019). (ii) the visual grounding ability which VL-BART failed to learn (Sec.5.3) is important for multimodal machine translation task.

[10] https://github.com/tylin/coco-caption

[11] https://github.com/mjpost/sacrebleu

Table 9. Multi-task finetuning results on VQA and RefCOCOg. With a single set of parmeters, our multi-task model achieves similar performance to separately optimized single-task models.

Method	Finetuning tasks	VQA Karpathy test Acc	RefCOCOg test Acc
VL-T5	VQA	67.9	-
VL-T5	RefCOCOg	-	71.3
VL-T5	VQA + RefCOCOg	67.0	70.1

5.7. Multi-Task Finetuning

While our framework has unified the architecture for different downstream tasks, the parameters are separately optimized. To see whether we can go one step further, we train a single model that tackles different kinds of tasks at once with the same set of weights. Specifically, we finetune VL-T5 on two different tasks, VQA (Goyal et al., 2019) and RefCOCOg (Mao et al., 2016), in a multi-task learning setup. At each finetuning step, we sample a mini-batch of examples from one of the two tasks. The existing vision-and-language multi-task learning method (Lu et al., 2020) trains multiple task-specific heads and only shares the backbone encoder, as illustrated in Fig. 3. With the help of our unified encoder-decoder architecture and generative pretraining, we build a unified multi-task model, where only a single shared language modeling head is learned for both tasks.

Table 9 shows the multi-task and single-task finetuning results of VL-T5 on VQA and RefCOCOg. On both tasks, our multi-task model achieves similar performance compared to the single-task models, while using a single set of weights shared by both tasks. Since we did not use advanced multi-task learning strategies such as oversampling or dynamic stop-and-go (Lu et al., 2020), we expect the multi-task performance of our model to be further improved with these orthogonal techniques.

6. Conclusion

In this work, we proposed VL-T5 and VL-BART which tackle vision-and-language tasks with a unified text generation objective. Experiments show VL-T5 and VL-BART can achieve comparable performance with state-of-the-art vision-and-language transformers on diverse vision-and-language tasks without hand-crafted architectures and objectives. Especially, we demonstrate our generative approach is better suited for open-ended visual question answering. In addition, we also showed it is possible to train two different tasks simultaneously using the same architecture with the same weight while not losing much performance, it would be an interesting future work to further explore this direction by adding even more tasks.

Acknowledgments

We thank Hyounghun Kim, Zineng Tang, Swarnadeep Saha, Xiang Zhou for their comments and suggestions. This work was supported by NSF-CAREER Award 1846185, ARO-YIP Award W911NF-18-1-0336, DARPA MCS Grant N66001-19-2-4031, Google Focused Research Award, and Bloomberg Data Science Ph.D. Fellowship. The views, opinions, and/or findings contained in this article are those of the authors and not of the funding agency.

References

Anderson, P., Fernando, B., Johnson, M., and Gould, S. SPICE: Semantic Propositional Image Caption Evaluation. In *ECCV*, 2016.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *CVPR*, 2018. URL http://arxiv.org/abs/1707.07998.

Banerjee, S. and Lavie, A. METEOR : An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL Workshop*, 2005.

Barrault, L., Bougares, F., Specia, L., Lala, C., Elliott, D., and Frank, S. Findings of the Third Shared Task on Multimodal Machine Translation. In *WMT*, pp. 304–323, 2018. doi: 10.18653/v1/w18-6402.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. In *NeurIPS*, 2020. URL http://arxiv.org/abs/2005.14165.

Caglayan, O., Madhyastha, P., Specia, L., and Barrault, L. Probing the Need for Visual Context in Multimodal Machine Translation. In *NAACL*, 2019. ISBN 9781950737130. doi: 10.18653/v1/n19-1422.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollar, P., and Zitnick, C. L. Microsoft COCO Captions: Data Collection and Evaluation Server. apr 2015. URL http://arxiv.org/abs/1504.00325.

Chen, Y.-c., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*, 2020. URL https://arxiv.org/abs/1909.11740.

Cho, J., Lu, J., Schwenk, D., Hajishirzi, H., and Kembhavi, A. X-LXMERT: Paint, Caption and Answer Questions with Multi-Modal Transformers. In *EMNLP*, 2020. doi: 10.18653/v1/2020.emnlp-main.707.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, oct 2019. URL http://arxiv.org/abs/1810.04805.

Elliott, D., Frank, S., Sima'an, K., and Specia, L. Multi30K : Multilingual English-German Image Descriptions. In *ACL Workshop*, pp. 70–74, 2016.

Gao, T., Fisch, A., and Chen, D. Making Pre-trained Language Models Better Few-shot Learners. 2020. URL http://arxiv.org/abs/2012.15723.

Goyal, Y., Khot, T., Agrawal, A., Summers-Stay, D., Batra, D., and Parikh, D. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *International Journal of Computer Vision*, 2019. ISSN 15731405. doi: 10.1007/s11263-018-1116-0.

Grönroos, S.-A., Huet, B., Kurimo, M., Laaksonen, J., Merialdo, B., Pham, P., Sjöberg, M., Sulubacak, U., Tiedemann, J., Troncy, R., and Vázquez, R. The MeMAD Submission to the WMT18 Multimodal Translation Task. In *WMT*, volume 2, pp. 609–617, 2018.

He, K., Gkioxari, G., Dollar, P., and Girshick, R. Mask R-CNN. *ICCV*, 2017.

Huang, L., Wang, W., Chen, J., and Wei, X. Y. Attention on attention for image captioning. In *ICCV*, pp. 4633–4642, 2019. ISBN 9781728148038. doi: 10.1109/ICCV.2019.00473.

Huang, Z., Zeng, Z., Liu, B., Fu, D., and Fu, J. Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers. 2020. URL http://arxiv.org/abs/2004.00849.

Hudson, D. A. and Manning, C. D. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. ISBN 9781728132938. doi: 10.1109/CVPR.2019.00686.

Johnson, J., Karpathy, A., and Fei-Fei, L. DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In *CVPR*, 2016.

Karpathy, A. and Fei-Fei, L. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *CVPR*, 2015. ISBN 9781467369640. doi: 10.1109/TPAMI.2016.2598339.

Kazemzadeh, S., Ordonez, V., Matten, M., and Berg, T. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

Keskar, N. S., McCann, B., Xiong, C., and Socher, R. Unifying Question Answering and Text Classification via Span Extraction. 2019. URL http://arxiv.org/abs/1904.09286.

Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., and Hajishirzi, H. Unified QA : Crossing Format Boundaries with a Single QA System. In *Findings of EMNLP*, 2020.

Kim, J.-h., Jun, J., and Zhang, B.-t. Bilinear Attention Networks. In *NeurIPS*, pp. 1–12, 2018.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Jia-Li, L., Shamma, D. A., Michael Bernstein, and Fei-Fei, L. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 2016. ISSN 15731405. doi: 10.1007/s11263-016-0981-7.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*, 2020.

Lei, J., Yu, L., Bansal, M., and Berg, T. L. Tvqa: Localized, compositional video question answering. In *EMNLP*, 2018.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., and Bart, P.-t. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*, 2020.

Li, L., Chen, Y.-C., Yu Cheng, Z. G., Yu, L., and Liu, J. HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training. In *EMNLP*, 2020a.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., and Gao, J. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *ECCV*, 2020b. URL http://arxiv.org/abs/2004.06165.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. ISBN 978-3-319-10601-4. doi: 10.1007/978-3-319-10602-1_48.

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

Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. In *ICLR*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lu, J., Batra, D., Parikh, D., and Lee, S. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*, 2019. URL http://arxiv.org/abs/1908.02265.

Lu, J., Goswami, V., Rohrbach, M., Parikh, D., and Lee, S. 12-in-1: Multi-Task Vision and Language Representation Learning. In *CVPR*, 2020. URL http://arxiv.org/abs/1912.02315.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A., and Murphy, K. Generation and Comprehension of Unambiguous Object Descriptions. In *CVPR*, 2016.

Mccann, B., Keskar, N. S., Xiong, C., and Socher, R. The Natural Language Decathlon : Multitask Learning as Question Answering. 2018.

Miech, A., Alayrac, J.-B., Smaira, L., Laptev, I., Sivic, J., and Zisserman, A. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020.

Narang, S., Diamos, G., Elsen, E., Micikevicius, P., Alben, J., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. Mixed Precision Training. In *ICLR*, 2018. URL https://openreview.net/forum?id=r1gs9JgRZ.

Nogueira, R., Jiang, Z., Lin, J., Mar, I. R., Pradeep, R., and Lin, J. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of EMNLP*, pp. 1–8, 2020.

Ordonez, V., Kulkarni, G., and Berg, T. L. Im2Text : Describing Images Using 1 Million Captioned Photographs. In *NIPS*, 2011.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. W.-j. BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL*, 2002. ISBN 1-55860-883-4. doi: 10.3115/1073083.1073135. URL http://portal.acm.org/citation.cfm?doid=1073083.1073135http://dl.acm.org/citation.cfm?id=1073135.

Paszke, A., Gross, S., Chintala, S., Chana, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in PyTorch. In *NIPS Workshop*, 2017. URL https://openreview.net/pdf?id=BJJsrmfCZ.

Post, M. A Call for Clarity in Reporting BLEU Scores. In *WMT*, volume 1, pp. 186–191, 2018.

Press, O. and Wolf, L. Using the Output Embedding to Improve Language Models. In *EACL*, 2017.

Radford, A., Wook, J., Chris, K., Aditya, H., Gabriel, R., Sandhini, G., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 21:1–67, 2019. URL http://arxiv.org/abs/1910.10683.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

Ren, S., He, K., Girshick, R., and Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015. URL https://arxiv.org/abs/1506.01497.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. ISBN 9781948087322. URL https://www.aclweb.org/anthology/P18-1238/.

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-Attention with Relative Position Representations. In *NAACL*, 2018.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *ICML*, 2019. URL http://arxiv.org/abs/1905.02450.

Suhr, A., Zhou, S., Zhang, A., Zhang, I., Bai, H., and Artzi, Y. A Corpus for Reasoning About Natural Language Grounded in Photographs. In *ACL*, 2019. URL http://arxiv.org/abs/1811.00491.

Sun, C., Baradel, F., Murphy, K., and Schmid, C. Contrastive Bidirectional Transformer for Temporal Representation Learning. 2019a. URL http://arxiv.org/abs/1906.05743.

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. VideoBERT: A Joint Model for Video and Language Representation Learning. In *ICCV*, 2019b. URL http://arxiv.org/abs/1904.01766.

Tan, H. and Bansal, M. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *EMNLP*, 2019. URL http://arxiv.org/abs/1908.07490.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention Is All You Need. In *NIPS*, 2017. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Vedantam, R., Zitnick, C. L., and Parikh, D. CIDEr: Consensus-based Image Description Evaluation. In *CVPR*, nov 2015. URL http://arxiv.org/abs/1411.5726.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2018.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*, 2017.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. HuggingFace's Transformers: State-of-the-art Natural Language Processing. 2019. URL http://arxiv.org/abs/1910.03771.

Xia, Q., Huang, H., Duan, N., Zhang, D., and Ji, L. XGPT : Cross-modal Generative Pre-Training for Image Captioning. 2020. URL https://arxiv.org/abs/2003.01473.

Xu, J., Mei, T., Yao, T., and Rui, Y. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.

Yao, S. and Wan, X. Multimodal Transformer for Multimodal Machine Translation. In *ACL*, pp. 4346–4350, 2020. doi: 10.18653/v1/2020.acl-main.400.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions. *TACL*, 2(April):67–78, 2014. ISSN 2307-387X. URL http://nlp.cs.illinois.edu/HockenmaierGroup/Papers/DenotationGraph.pdf.

Yu, L., Poirson, P., Yang, S., Berg, A. C., and Berg, T. L. Modeling context in referring expressions. In *ECCV*, 2016.

Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., and Berg, T. L. MAttNet : Modular Attention Network for Referring Expression Comprehension. In *CVPR*, 2018a. URL https://arxiv.org/abs/1801.08186.

Yu, Y., Kim, J., and Kim, G. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018b.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*, 2018.

Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. From Recognition to Cognition: Visual Commonsense Reasoning. In *CVPR*, 2019. URL http://arxiv.org/abs/1811.10830.

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., and Gao, I. VinVL: Making Visual Representations Matter in Vision-Language Models. 2021.

Zhou, L., Xu, C., and Corso, J. J. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.

Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J. J., and Gao, J. Unified Vision-Language Pre-Training for Image Captioning and VQA. In *AAAI*, 2020. URL http://arxiv.org/abs/1909.11059.

Zhu, L. and Yang, Y. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.

Zhu, Y., Groth, O., Bernstein, M., and Fei-Fei, L. Visual7W: Grounded Question Answering in Images. In *CVPR*, 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.540. URL http://arxiv.org/abs/1511.03416.

Table 10. Summary of baseline vision-and-language transformers. [a]Since not all models report exact parameter numbers, we provide rough estimates compared to BERT_{Base} (86M; noted as P), where word embedding parameters are excluded. [b]LXMERT and XGPT are not initialized from pretrained language models. LXMERT authors found pretraining from scratch was more effective than initialization from BERT_{Base} in their experiments. XGPT uses text pretraining on Conceptual captions and COCO captions with Masked LM (Devlin et al., 2019) and Masked Seq2Seq (Song et al., 2019) objectives before V&L pretraining. [c]LXMERT (text+visual+cross-modal) and ViLBERT (cross-modal) use dual-stream encoders. ViLBERT uses 768/1024-dim hidden states for text/visual streams respectively. XGPT uses AoA module (Huang et al., 2019) as visual encoder. Rest of the models use single-stream encoders. [d]For generation tasks, Unified VLP and Oscar use causal mask and reuse encoder as decoder similar to UniLM. [e]XGPT also uses shared parameters for encoder and decoder, but its decoder is right-shifted and predicts next tokens. [f]Unified VLP is initialized from UniLM, which is initialized from BERT_{Large}. [g]Oscar uses object tags as additional text inputs.

	V&L Pretraining		Hyperparameters						
	Dataset	# Imgs	Arch. type	Backbone	# Layers	# Params[a]	Hidden dim	# Regions	Position Emb
LXMERT	COCO+VG	180K	Encoder	-[b]	9+5+5[c]	2P	768	36	absolute
ViLBERT	CC	3M	Encoder	BERT_{Base}	12[c]	2.5P	768/1024[c]	10∼36	absolute
UNITER_{Base}	CC+SBU+COCO+VG	4M	Encoder	BERT_{Base}	12	P	768	10∼100	absolute
Unified VLP	CC	3M	Encoder[d]	UniLM[f]	12	P	768	100	absolute
Oscar_{Base}	CC+SBU+COCO+VG+Flickr30K	4M	Encoder[d]	BERT_{Base}	12	P	768	50[g]	absolute
XGPT	CC+COCO	3M	Enc-Dec[e]	-[b]	1[c]+12+12	P	768	100	absolute
VL-T5	COCO+VG	180K	Enc-Dec	T5_{Base}	12+12	2P	768	36	relative
VL-BART	COCO+VG	180K	Enc-Dec	BART_{Base}	6+6	P	768	36	absolute

Table 11. Pretraining tasks used in our vision-and-language pretraining. The images that have any intersection with evaluation set of downstream tasks (e.g., COCO caption, RefCOCOg) and the held-out validation set for pretraining are excluded.

Task	Image source	Text source	# Examples
Multimodal language modeling	COCO, VG	COCO caption, VG caption	4.9M (# captions)
Visual question answering	COCO, VG	VQA, GQA, Visual7W	2.5M (# questions)
Image-text matching	COCO	COCO caption	533K (# captions)
Visual grounding	COCO, VG	object&attribute tags	163K (# images)
Grounded captioning	COCO, VG	object&attribute tags	163K (# images)

A. Summary of Vision-and-Language Transformers

In Table 10, we compare the baseline vision-and-language transformers and our VL-T5 and VL-BART in detail.

B. Pretraining and Downstream Task Details

In Table 11 and Table 12, we show the detailed statistics of our pretraining and downstream datasets and tasks. In Table 13, we show the hyperparameters that we used in our pretraining and downstream task experiments.

Table 12. Statistics of the datasets used in downstream tasks. The data that are not used for training/validation (e.g., COCO test2015 images) and data for leaderboard submissions (e.g., test-dev/test-std for VQA, test for GQA) are excluded.

Datasets	Image source	# Images (train)	# Text (train)	Metric
VQA	COCO	123K (113K)	658K (605K)	VQA-score
GQA	VG	82.7K (82.3K)	1.08M (1.07M)	Accuracy
NLVR^2	Web Crawled	238K (206K)	100K (86K)	Accuracy
RefCOCOg	COCO	26K (21K)	95K (80K)	Accuracy
VCR	Movie Clips	110K (80K)	290K (212K)	Accuracy
COCO Caption	COCO	123K (113K)	616K (566K)	BLEU,CIDEr,METEOR,SPICE
Multi30K En-De	Flickr30K	31K (29K)	31K (29K)	BLEU

Table 13. Hyperparameters for pretraining and downtream tasks

Model	Task	Learning rate	Batch size	Epochs
VL-T5	Pretraining	1e-4	320	30
	VCR Pretraining	5e-5	80	20
	VQA	5e-5	320	20
	GQA	1e-5	240	20
	NLVR^2	5e-5	120	20
	RefCOCOg	5e-5	360	20
	VCR	5e-5	16	20
	COCO Caption	3e-5	320	20
	Multi30K En-De	5e-5	120	20
VL-BART	Pretraining	1e-4	600	30
	VCR Pretraining	5e-5	120	20
	VQA	5e-5	600	20
	GQA	1e-5	800	20
	NLVR^2	5e-5	400	20
	RefCOCOg	5e-5	1200	20
	VCR	5e-5	48	20
	COCO Caption	3e-5	520	20
	Multi30K En-De	5e-5	320	20