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 questions that have rare answers. Also, we show that our framework allows multi-task learning in a single architecture with a single set of parameters, achieving similar performance to separately optimized single-task models. Our code is publicly available at: https://github.com/j-min/VL-T5

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 firstly pretrained on 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.

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. Both answering the question “What is the man jumping over?” and grounding an image region corresponding to the phrase “yellow fire hydrant” require recognizing 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 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., 2020) and BART (Lewis et al., 2020) with visual understanding ability, named ‘VL-T5’ and ‘VL-BART’. In contrast to existing methods that train different architectures for each pretraining and downstream task, our models tackle all 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. 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 NLVR2 (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 task-specific architectures and objective functions. In addition, we found that our generative models have better generalization ability compared to the discriminative versions in the rare-answer scenario on visual question answering, when ground truth answers for given questions are rarely seen during training. Finally, we also experiment with our unified framework under the multi-task learning setup on all 7 downstream tasks. With a single architecture and a single set of weights, our model achieves similar performance to separately optimized single-task models.

### 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., 2020) have achieved remarkable success for many natural language understanding tasks (Rajpurkar et al., 2016; Zellers et al., 2018; Wang et al., 2018; Williams et al., 2017). Following this success, 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; Zhang 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 non-pretraining approaches (Yu et al., 2018a; Anderson et al., 2018; Kim et al., 2018; Yu et al., 2018b) in a wide range of discriminative (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. While existing image+text models mostly use task-specific architectures and objectives, we seek to design a unified framework across different tasks.

**Unified frameworks:** One line of work focus on solving natural language processing tasks in a unified format, such as question answering (McCann et al., 2018), span prediction (Keskar et al., 2019), or text generation (Raffel et al., 2020; 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 framework that unifies vision-and-language problems as multimodal conditional text generation. We introduce VL-T5 and VL-BART based on two pretrained transformer language models: T5<sub>Base</sub> (Raffel et al., 2020) and BART<sub>Base</sub> (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 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 $n=36$ object regions from a Faster R-CNN (Ren et al., 2015) trained on Visual
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.

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. This augmented input text is then tokenized as \( \{x_1, \ldots, x_{|x|}\} \) and encoded as learned embedding \( e^x = \{e^x_1, \ldots, e^x_{|x|}\} \). The embedding parameters are shared by the encoder, decoder, and language modeling head (Press & Wolf, 2017). Since the attention layers are permutation-invariant, BART learns positional embeddings (Vaswani et al., 2017; Devlin et al., 2019) for absolute to-discriminative tasks, as shown in Table 1. 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.

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, consisting of a self-attention layer and a fully-connected layer with residual connections. Our decoder is another stack of \( m \) transformer blocks similar to the multimodal encoder, where each block has an additional cross-attention layer. As shown in Fig. 2 (a), the encoder takes the concatenation of text and visual embeddings as input and outputs their contextualized joint representations \( h = \{h^x_1, \ldots, h^x_{|x|}, h^v_1, \ldots, h^v_{|v|}\} = \text{Enc}(e^x, e^v) \). Then 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 \( P_{\theta}(y_j | y_{<j}, x, v) = \text{Dec}(y_{<j}, h) \). We suggest readers to check Raffel et al. (2020); Lewis et al. (2020) for more details of our backbone models. For both pretraining (Sec. 4) and downstream tasks (Sec. 5), we train our model parameters \( \theta \) by minimizing the negative log-likelihood of label text \( y \) tokens given input text \( x \) and image \( v \):

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

3.4. Task-Specific Methods vs. Our Unified Framework

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). Visual question answering 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.,
Comparison between existing methods 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 the same language modeling architecture and maximum likelihood estimation on label text for all tasks.

Existing vision-and-language transformers are trained with different datasets and computational budgets, thus their results may not be directly comparable to each other. We show the number of their pretraining images in Table 2. We only use captions from COCO for this task, since many short captions from VG and visual questions are nondistinctive.

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. 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 for the visual question answering task. Details of these pretraining tasks are in Sec. 4.2. Overall, our pretraining dataset contains 9.18M image-text pairs on 180K distinct images. We show more details of the pretraining data in appendix.

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 appendix. In the rest of this section, we explain these tasks in detail.

Multimodal language modeling: We follow Raffel et al. (2020) and Lewis et al. (2020) to construct the language modeling pretraining task. For VL-T5, we mask 15% of input text tokens and replace contiguous text span with sentinel tokens (e.g., <text sentinel>). For VL-BART, we mask 30% of input text tokens with <mask> tokens. Then we predict the masked text. See Table 1 for examples.

Visual question answering: We include visual question answering in our pretraining tasks as in Tan & Bansal (2019). While previous methods (Tan & Bansal, 2019; Lu et al., 2019; Chen et al., 2020) tackle the task as classification over predefined answer candidates (illustrated in Fig. 3), we directly generate answers in their original text format.

Image-text matching: In this task, the model needs to verify whether a text corresponds to an image. We consider an image and its captions as positive pairs. With a probability

\text{score}(a, x, v) = \min(\#\text{humans that gave answer a} \times 0.3, 1)

of the pretraining data and illustrate how we formulate diverse vision-and-language pretraining tasks as multimodal conditional text generation.

4.3. Pretraining Tasks

In this section, we describe how we pretrain our VL-T5 and VL-BART models (Sec. 3). We start with the details...
of 50%, we randomly sample another training image’s caption to create a negative pair. The model then predicts the correspondence with “true” or “false” as shown in Table 1.

**Visual grounding:** We 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). 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 (e.g., attribute: “yellow” + object: “fire hydrant” → “yellow fire hydrant”). 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 grounded captioning as an inverse task of visual grounding. As shown in Table 1, given a visual sentinel token (which indicates an image region) as text input, the model is asked to generate a corresponding textual description of the image region.

### 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. We use batch size 320 and 600 for VL-T5 and VL-BART, respectively. We use AdamW (Loshchilov & Hutter, 2019) with $(\beta_1, \beta_2) = (0.9, 0.999)$ and learning rate 1e-4 with 5% linear warmup schedule. 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 compare our generative architectures VL-T5 and VL-BART on a diverse set of 7 downstream tasks (details in Appendix) with existing vision-and-language pre-trained transformers (Tan & Bansal, 2019; Lu et al., 2019; Chen et al., 2020; Zhou et al., 2020; Li et al., 2020b; Xia et al., 2020). As summarized in Table 2, our unified generative approach (with the input-output format in Table 1) shows performance close to the task-specific models, most of which are discriminative. In the rest of this section, we provide detailed comparisons w.r.t. the baselines.

#### 5.1. Visual Question Answering: VQA and GQA

The visual question answering task requires models to answer a question to a given context image. Table 2 compares our models VL-T5 and VL-BART with existing methods on VQA (Goyal et al., 2019) and GQA (Hudson & Manning, 2019). For both tasks, our models achieve comparable performance to existing approaches.

**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. This strategy achieves strong performance but not generalizes to real-world open-ended scenarios. To quantitatively compare the existing discriminative approaches and our generative approach, we break down VQA questions into in-domain and out-of-domain questions, in terms of whether the best answer $a^*$ for each question is included in the top-K ($K=3, 129$) answer candidates $A^{top_k}$. After this split, the in-domain subset contains 24,722 questions, and the out-of-domain subset contains 1,558 questions. Table 3 shows the performance. For discriminative baselines, we introduce a sigmoid MLP classifier on top of the decoder representation of `start-of-sequence` token `<s>`, following
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.

| 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 | Multi30K En-De test 2018 |
| 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*                    | 48.9                | 116.6                    | 28.1                     |

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: \(a^*/ A^{topk}\).

| Method       | In-domain | Out-of-domain | Overall |
|--------------|-----------|---------------|---------|
| Discriminative | UNITER\(_{Base}\) | 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   |

LXMERT and UNITER. Comparing models with the same backbone, we notice the generative models improve upon the discriminative baselines across all the subsets. This improvement is more significant on 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. 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 (about 3 points).

**Dataset-specific prefixes:** As shown in recent works (Gao et al., 2020; Shin et al., 2020; Li & Liang, 2021; Radford et al., 2021), different text prompts could result in different finetuning results. We thus experiment with a single prefix ‘vqa’ for both VQA and GQA in VL-T5 pretraining/finetuning. Interestingly, we found slight performance increases from the original dataset-specific prefix: VQA Karpathy-test (67.9 → 69.3); GQA test-dev (60.0 → 60.2). This shows that a single model can successfully handle multiple VQA tasks without dataset-specific prefixes (similar results were observed in text QA (Khashabi et al., 2020)).

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 images. To apply our model to this task, we concatenate region features from the two images and use different image id embeddings to disambiguate the regions from the two images. Then our model learns to generate text labels “true” and “false”. This is similar to the Triplet setting described in UNITER (Chen et al., 2020). In Fig. 4, we illustrate three common encoding settings for NLVR\(^2\).

Table 4 shows the model results on NLVR\(^2\) 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   |
5.3. Referring Expression Comprehension: RefCOCOg

Referring expression comprehension requires a model to correctly localize an object described by a given phrase (e.g., ‘the car on the left’). 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 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 region detections from Mask R-CNN (He et al., 2017) 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 2 compares our 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 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).

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), and it can be decomposed into two multiple choice 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. During inference, we use

\[
P(\text{true})/P(\text{true})+P(\text{false})
\]

to rank the choices and select the one with the highest score.

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. In Table 6, we show the experiment results with second stage pretraining on VCR. On VCR val split, comparing to the base models that do not pretrain, 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. On 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.

![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.](image_url)
5.5. Image Captioning: COCO Caption

We evaluate automatic caption generation performance on MS COCO Caption dataset (Chen et al., 2015). We use Karpathy split (Karpathy & Fei-Fei, 2015), which re-splits train2014 and val2014 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 where textual context is not meaningful, 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 experiment 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\(^5\).

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 object tag augmentation during pretraining like Oscar would further boost the performance of our models.

5.6. Multimodal Machine Translation: Multi30K

We evaluate multimodal machine translation performance on Multi30K dataset (Elliott et al., 2016), where a model
Table 9. Single-task vs. Multi-task finetuning results on 7 tasks. With a single set of parameters, our multi-task model achieves similar performance to separately optimized single-task models. We denote the number of parameters of single VL-T5 model as P.

| Method | Finetuning tasks | # Params | Discriminative tasks | Generative tasks |
|--------|-----------------|----------|----------------------|------------------|
|        |                 |          | VQA                  | GQA              | NLVR\(^2\) | RefCOCOg\(^3\) | VCR   | COCO Caption | Multi30K En-De |
|        |                 |          | Karpathy test Acc    | test-dev Acc     | test-P     | test\(^d\) Acc | test val Acc | Karpathy test CID\(\text{Er}\) | test2018 Bleu |
| VL-T5  | single task     | 7P       | 67.9                 | 59.3             | 60.0       | 73.6              | 71.3 | 57.5 | 116.1 | 38.6 |
| VL-T5  | all tasks       | P        | 67.2                 | 58.9             | 69.4       | 71.6              | 55.3 | 69.4 | 110.8 | 37.6 |

5.7. Multi-Task Finetuning

Single-task vs. 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 further, we finetune a single VL-T5 model for 20 epochs, where it tackles 7 different tasks with the same set of weights. At each finetuning step, we sample a mini-batch of examples from one of the 7 tasks in a round-robin fashion. For a fair comparison, we use single-task baselines without augmentations (e.g., no 2nd stage pretraining for VCR, no object tags for COCO Captioning). Table 9 shows that our multi-task model achieves comparable performance to the separately optimized single-task models on all 7 tasks with a single set of parameters.

Single shared head vs. Task-specific heads: We also experiment with the multi-task finetuning setup of ViLBERT-MT (Lu et al., 2020), where a task-specific head is finetuned for each of 7 downstream tasks while sharing backbone. The head parameters are initialized from the pre-trained LM head and separately updated during finetuning. The 7 task-specific heads (7H) add 7 \(\times\) 32K (vocab size) \(\times\) 768 (embedding size) = 172M parameters, which is 80% of original VL-T5’s 220M parameters (P), resulting around 400M parameters in total. Since the increased parameters make the training slow, we compare both models by 5th epoch checkpoints. Table 10 shows that VL-T5 with single shared head achieves almost equal performance with task-specific heads, while having much fewer total parameters.

Table 10. Multi-task finetuning with single/task-specific heads. While three tasks are included for brevity, the rest of the tasks also show the minimal differences between two setups.

| Method                 | # Params | VQA Karpathy test Acc | GQA test-dev Acc | COCO Caption Karpathy test CID\(\text{Er}\) |
|------------------------|----------|-----------------------|------------------|---------------------------------------------|
| Single shared head     | P        | 68.3                  | 59.3             | 110.6                                       |
| Task-specific heads    | P+7H=1.8P | 68.5                  | 59.3             | 110.9                                       |

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 seven different tasks simultaneously using a single architecture with single parameters without 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, and anonymous reviewers 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.
Unifying Vision-and-Language Tasks via Text Generation

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.

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. PixelBERT: Aligning Image Pixels with Text by Deep Multi-BERT 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.

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.

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.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. 2021.

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: Pre-training 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.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. W.-j. BLEU: a Method for Automatic Evaluation of Machine Translation. In ACL, 2002. ISBN 1-58608-883-4. doi: 10.3115/1073083.1073135. URL http://portal.acm.org/citation.cfm?doid=1073083.1073135.

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, 2020. 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.

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-Attention with Relative Position Representations. In NAACL, 2018.
Shin, T., Razeghi, Y., IV, R. L. L., Wallace, E., and Singh, S. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In EMNLP, 2020.

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/1806.05743.

Sun, C., Baradel, F., Murphy, K., and Schmid, C. Contrastive Bidirectional Transformer for Temporal Representation Learning. 2019a. URL http://arxiv.org/abs/1904.01766.

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.

Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., and Berg, T. L. MAAttNet : 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.
Unifying Vision-and-Language Tasks via Text Generation

Table 11. 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.

| Dataset          | Image source | # Ings | Arch. type | Backbone | # Layers | # Params | Hidden dim | # Regions | Position Emb |
|------------------|--------------|--------|------------|----------|----------|----------|------------|------------|--------------|
| LXMERT           | COCO+VG      | 180K   | Encoder    | BERT_{Base} | 12" 2.5P | 768/1024\* | 10~36       | absolute   |
| ViLBERT          | CC           | 3M     | Encoder    | BERT_{Base} | 12 P    | 768/1024\* | 10~100      | absolute   |
| UNITER_{Base}    | COCO+VG      | 4M     | Encoder    | BERT_{Base} | 12 P    | 768 100    | absolute    |
| Unified VLP      | CC           | 3M     | Encoder\*  | UniLM\*   | 12 P    | 768 100    | absolute    |
| Oscar_{Base}     | COCO+VG      | 3M     | Enc-Dec    | BERT_{Base} | 12 P    | 768 100    | absolute    |
| XGPT             | CC+COCO      | 3M     | Enc-Dec\*  | T5_{Base} | 12+12 P  | 768 36     | relative    |
| VL-T5            | COCO+VG      | 180K   | Enc-Dec    | T5_{Base} | 12+12 P  | 768 36     | absolute    |
| VL-BART          | COCO+VG      | 180K   | Enc-Dec    | BART_{Base} | 6+6 P   | 768 36     | absolute    |

Table 12. 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. Comparison with Baselines

In Table 11, we compare the baseline vision-and-language transformers with our VL-T5 and VL-BART in detail, including their pretraining datasets, architecture, etc.

B. Implementation Details

In Table 12 and Table 13, we show the detailed statistics of our pretraining and downstream datasets and tasks. In Table 14, we show the hyperparameters that we used in our pretraining and downstream task experiments. We provide the links to download pretraining and downstream datasets.

B.1. Pretraining Data

Overall, our pretraining dataset contains 9.18M image-text pairs on 180K distinct images. We carefully split our pretraining data to avoid any intersection between our training data and the validation/test sets of the downstream tasks (e.g., COCO Captioning, RefCOCOg). In this process, around 10K images are excluded from the training sets of COCO\textsuperscript{7} and Visual Genome\textsuperscript{8}. We use COCO Karpathy val split (Karpathy & Fei-Fei, 2015) with 5,000 images as our validation set to monitor pretraining performance.

B.2. Downstream Tasks

VQA\textsuperscript{9}, COCO caption For both VQA and COCO captioning tasks, we follow Karpathy split (Karpathy & Fei-Fei, 2015), which re-splits train2014 and val2014 COCO images (Lin et al., 2014) into 113,287 / 5,000 / 5,000 images for train / validation / test.

GQA\textsuperscript{10} Following LXMERT (Tan & Bansal, 2019), we use GQA-balanced version. We use train and val splits for training and use test-dev split for validation. Train / val / test-

---

\textsuperscript{7}https://cocodataset.org/#download

\textsuperscript{8}http://visualgenome.org/api/v0/api_home.html

\textsuperscript{9}https://visualqa.org/download.html

\textsuperscript{10}https://cs.stanford.edu/people/dorarad/gqa/download.html
Table 13. 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      |
| NLVR²          | 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 14. Hyperparameters for pretraining and downstream 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     |
|        | NLVR²             | 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     |
|        | NLVR²             | 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     |
dev splits consist of 943,000 / 132,062 / 12,578 questions, respectively.

**NLVR** \(^{11}\) Train / val / test-P splits consist of 86,373 / 6982 / 6967 sentences, respectively. We train our model on train split and use val split for validation.

**VCR** \(^{12}\) Train / val / test splits consist of 212,923 / 26,534 / 25,263 questions, respectively. We train our model on train split and use val split for validation.

**RefCOCOg** \(^{13}\) We use *umd* split, which consists of train / val / test sets with 42,226 / 2,573 / 5,023 sentences, respectively. Following UNITER (Chen et al., 2020) and MAttNet (Yu et al., 2018a), we use ground truth COCO boxes for training, and use the detected boxes from an off-the-shelf Mask R-CNN \(^{14}\) as candidates during inference.

**Multi30K En-De** \(^{15}\) The train / val / test2016 / test2017 / test2018 splits consist of 29,000 / 1,014 / 1,000 / 1,000 / 1,017 English-German sentence pairs, respectively.

\(^{11}\)http://lil.nlp.cornell.edu/nlvr/
\(^{12}\)https://visualcommonsense.com/download/
\(^{13}\)https://github.com/lichengunc/refer
\(^{14}\)https://github.com/lichengunc/MAttNet#pre-computed-detectionsmasks
\(^{15}\)https://github.com/multi30k/dataset