VinVL: Revisiting Visual Representations in Vision-Language Models

Pengchuan Zhang[♡†] Xiujun Li[♡♠†] Xiaowei Hu[♡] Jianwei Yang[♡] Lei Zhang[♡]
Lijuan Wang[♡] Yejin Choi[♠] Jianfeng Gao[♡]

Abstract

This paper presents a detailed study of improving visual representations for vision language (VL) tasks and develops an improved object detection model to provide object-centric representations of images. Compared to the most widely used bottom-up and top-down *model [2], the new model is bigger, better-designed for VL tasks, and pre-trained on much larger training corpora that combine multiple public annotated object detection datasets. Therefore, it can generate representations of a richer collection of visual objects and concepts. While previous VL research focuses mainly on improving the vision-language fusion model and leaves the object detection model improvement untouched, we show that visual features matter significantly in VL models. In our experiments we feed the visual features generated by the new object detection model into a Transformer-based VL fusion model* OSCAR *[20], and utilize an improved approach* OSCAR+ *to pre-train the VL model and fine-tune it on a wide range of downstream VL tasks. Our results show that the new visual features significantly improve the performance across all VL tasks, creating new state-of-the-art results on seven public benchmarks. Code, models and pre-extracted features are released at* https://github.com/pzzhang/VinVL.

1. Introduction

Vision language pre-training (VLP) has proved effective for a wide range of vision-language (VL) tasks [25, 35, 4, 33, 19, 18, 44, 20]. VLP typically consists of two stages: (1) an object detection model is pre-trained to encode an image and the visual objects in the image to feature vectors, and (2) a cross-modal fusion model is pre-trained to blend text and visual features. While existing VLP research focuses mainly on improving the cross-modal fusion model, this paper focuses on improving the object-centric visual representations and presents a comprehensive empirical study to demonstrate that visual features matter in VL models.

Among the aforementioned work, a widely-used object detection (OD) model [2] is trained on the Visual Genome dataset [15]. The OD model provides an object-centric representation of images, and has been used in many VL models as a black box. In this work, we pre-train a large-scale object-attribute detection model based on the ResNeXt-152 C4 architecture (short as X152-C4). Compared to the OD model of [2], the new model is better-designed for VL tasks, and is bigger and trained on much larger amounts of data, combining multiple public object detection datasets, including COCO [24], OpenImages (OI) [16], Objects365 [30] and Visual Genome (VG) [15]. As a result, our OD model achieves much better results on a wide range of VL tasks, as shown in Table 1. Compared to other typical OD models, such as X152-FPN trained on OpenImages, our new model can encode a more diverse collection of visual objects and concepts (e.g., producing visual representations for 1848 object categories and 524 attribute categories), as illustrated by an example in Figure 1.

To validate the effectiveness of the new OD model, we pre-train a Transformer-based cross-modal fusion model OSCAR+ [20] on a public dataset consisting of 8.85 million text-image pairs, where the visual representations of these images are produced by the new OD model and are fixed during OSCAR+ pre-training. We then fine-tune the pre-trained OSCAR+ for a wide range of downstream tasks, including VL understanding tasks such as VQA [8], GQA [12], NLVR2 [34], and COCO text-image retrieval [24], and VL generation tasks such as COCO image captioning [24] and NoCaps [1]. Our results show that the object-centric representations produced by the new OD model significantly improve the performance across all the VL tasks, often by a large margin over strong baselines using the classical OD model [2], creating new state of the arts on all these tasks, including GQA on which none of the published pre-trained models has surpassed the deliberately designed neural state machine (NSM) [11]. We will release the new OD model to the research community.

The main contributions of this work can be summarized as follows: (i) We present a comprehensive empirical study to demonstrate that visual features matter in VL models. (ii)

[♡]Microsoft Corporation [♠]University of Washington
† indicates equal contributions.

Visual feature	VQA		GQA		Image Captioning				NoCaps		Image Retrieval			Text Retrieval			NLVR2	
	test-dev	test-std	test-dev	test-std	B@4	M	C	S	C	S	R@1	R@5	R@10	R@1	R@5	R@10	dev	test-P
Anderson *et al.* [2]	73.16	73.44	61.58	61.62	40.5	29.7	137.6	22.8	86.58	12.38	54.0	80.8	88.5	70.0	91.1	95.5	78.07	78.36
Ours	**75.95**	**76.12**	**65.05**	**64.65**	**40.9**	**30.9**	**140.6**	**25.1**	**92.46**	**13.07**	**58.1**	**83.2**	**90.1**	**74.6**	**92.6**	**96.3**	**82.05**	**83.08**
Δ	**2.79** ↑	**2.68** ↑	**3.47** ↑	**3.03** ↑	**0.4** ↑	**1.2** ↑	**3.0** ↑	**2.3** ↑	**5.9** ↑	**0.7** ↑	**4.1** ↑	**2.4** ↑	**1.6** ↑	**4.6** ↑	**1.5** ↑	**0.8** ↑	**3.98** ↑	**4.71** ↑

Table 1: Uniform improvements on seven VL tasks by replacing visual features from Anderson *et al.* [2] with ours. The NoCaps baseline is from VIVO [9], and our results are obtained by directly replacing the visual features. The baselines for rest tasks are from OSCAR [20], and our results are obtained by replacing the visual features and performing OSCAR+ pre-training. All models are Bert-Base size. As analyzed in Section 4.2, the new visual features contributes 95% of the gain.

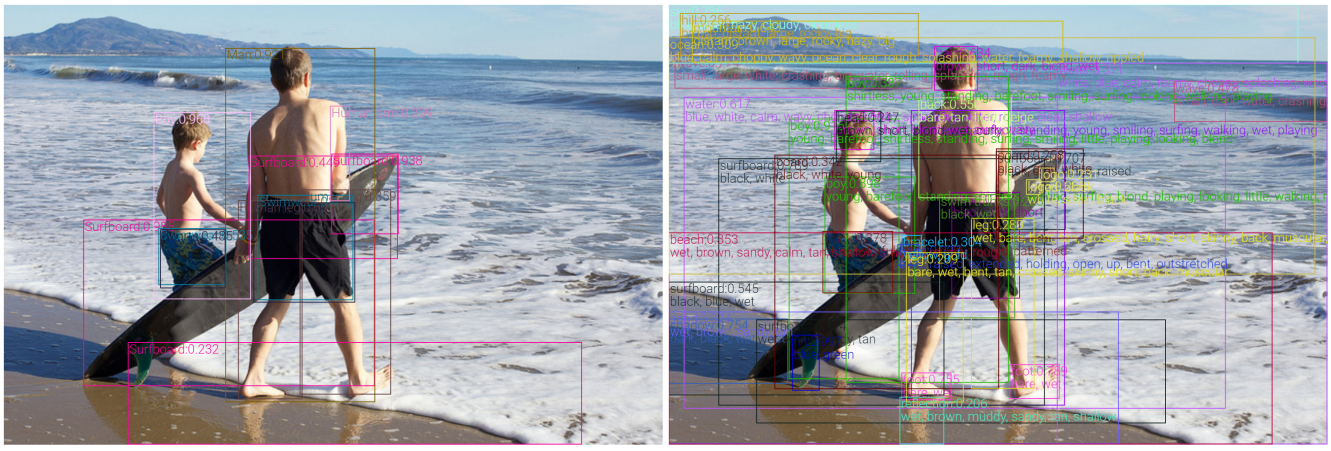

Figure 1: Predictions from an X152-FPN model trained on OpenImages (Left) and our X152-C4 model trained on four public object detection datasets (Right). Our model contains much richer semantics, such as richer visual concepts and attribute information, and the detected bounding boxes cover nearly all semantically meaningful regions. Compared with those from the common object classes in typical OD models (Left), the rich and diverse region features from our model (Right) are crucial for vision-language tasks. For concepts detected by both models, e.g., "boy", attributes from our model offer richer information, e.g., "young barefoot shirtless standing surfing smiling little playing looking blond boy". There are object concepts that are detected by our model but not by the OpenImages model, including fin, wave, foot, shadow, sky, hair, mountain, water, (bare, tan, light, beige) back, (blue, colorful, floral, multi colored, patterned) trunk, sand, beach, ocean, (yellow, gold) bracelet, logo, hill, head, (black, wet) swim trunks, black, wet swim trunks. Compared to the R101-C4 model of [2], our model produces more accurate object-attribute detection results and better visual features for VL applications; see Appendix A for the full pictures and predictions from [2].

We have developed a new object detection model that can produce better visual features of images than the classical OD model [2] and substantially uplifts the state-of-the-art results on all major VL tasks across multiple public benchmarks. (*iii*) We provide a detailed ablation study of our pre-trained object detection model to investigate the relative contribution to the performance improvement due to different design choices regarding diversity of object categories, visual attribute training, training data scale, model size, and model architecture.

2. Improving Vision in Vision Language

Deep learning-based VL models typically consist of two modules: an image understanding module **Vision** and a cross-modal understanding module **VL**:

$$(\boldsymbol{q}, \boldsymbol{v}) = \mathbf{Vision}(Img), \quad y = \mathbf{VL}(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v}), \qquad (1)$$

where Img and $\boldsymbol{w}$ are the inputs of the vision and language modalities, respectively. The output of the **Vision** module consists of $\boldsymbol{q}$ and $\boldsymbol{v}$. $\boldsymbol{q}$ is the semantic representation of the image, such as tags or detected objects, and $\boldsymbol{v}$ the distributional representation of the image in a high-dimensional latent space represented using e.g., the box or region[1] features produced by a VG-pre-trained Faster-RCNN model [2]. Most **VL** models use only the visual features $\boldsymbol{v}$, while the recently proposed OSCAR [20] model shows that $\boldsymbol{q}$ can serve as anchors for learning better vision-language joint representations and and thus can improve the performance on various VL tasks. $\boldsymbol{w}$ and y of the **VL** module of Equation (1) vary among different VL tasks. In VQA, $\boldsymbol{w}$ is a question and y is an answer to be predicted. In text-image retrieval, $\boldsymbol{w}$ is a sentence and y is the matching score

[1] We use the terms region and box interchangeably.

of a sentence-image pair. In image captioning, $\boldsymbol{w}$ is not given and y is a caption to be generated.

Inspired by the great success of pre-trained language models to various natural language processing tasks, vision-language pre-training (VLP) has achieved remarkable success in improving the performance of the cross-modal understanding module **VL** by (1) unifying vision and language modeling **VL** with Transformer and (2) pre-training the unified **VL** with large-scale text-image corpora. However, most recent works on VLP treat the image understanding module **Vision** as a black box and leave the visual feature improvement untouched since the development of the classical OD model [2] three years ago, despite that there has been much research progress on improving object detection by 1) developing much more diverse, richer, and larger training datasets (e.g. OpenImages and Objects 365), 2) gaining new insights in object detection algorithms such as feature pyramid network [22], one-stage dense prediction [23], and anchor-free detectors [36], and 3) leveraging more powerful GPUs for training bigger models.

In this work, we focus on improving **Vision** for better visual representations. We developed a new OD model by enriching the visual object and attribute categories, enlarging the model size and training on a much larger OD dataset, and thus advanced the state of the arts on a wide range of VL tasks. We detail how the new OD model is developed in the rest of this section and then describe the use of OSCAR+ for **VL** pre-training in Section 3.

2.1. Object Detection Pre-training

To improve the OD model for VL tasks, we utilize four public object detection datasets. As most datasets do not have attribute annotations, we adopt a *pre-training and fine-tuning* strategy to build our OD model. We first pre-train an OD model on a large-scale corpus consisting of four public datasets, and then fine-tune the model with an additional attribute branch on Visual Genome, making it capable of detecting both objects and attributes.

Data. Table 2 summarizes the statistics of the four public datasets used in our object detection pre-training, including COCO, OpenImagesV5 (OI), Objects365V1, and Visual Genome (VG). These datasets have complementary characters, and are extremely unbalanced in terms of data size, object vocabulary, and the number of annotations in each class. For example, the VG dataset has a rich and diverse set of annotations for both objects and their attributes with an open vocabulary. But its annotations are noisy and suffer from the missing-annotation problem. The COCO dataset, on the other hand, is very well annotated. But the coverage of visual objects and attributes is much lower than that in VG although we use both its 80 object classes and 91 stuff classes to include as diverse visual concepts as possible. We take the following steps to build a unified corpus by combining the four datasets.

1. First of all, to enhance visual concepts of tail classes, we perform class-aware sampling for OpenImages and Objects365 to get at least 2000 instances per class, resulting in 2.2M and 0.8M images, respectively.

2. To balance the contribution of each dataset, we merge the four datasets with 8 copies of COCO (8×0.11M), 8 copies of VG (8×0.1M), 2 copies of class-aware sampled Objects365 (2×0.8M) and one copy of the class-aware sampled OpenImages (2.2M).

3. To unify their object vocabularies, we use the VG vocabulary and its object aliases as the base vocabulary, merge a class from the other three datasets into a VG class if their class names or aliases match, and add a new class if no match is found.

4. Finally, we keep all VG classes that contain at least 30 instances, resulting in 1594 VG classes and 254 classes from the other three datasets that cannot be mapped to the VG vocabulary, resulting in a merged object detection dataset that contains 1848 classes.

Source	VG	COCO w/ stuff	Objects365	OpenImagesV5	Total
Image	97k	111k	609k	1.67M	2.49M
classes	1594	171	365	500	1848
Sampling	×8	×8	CA-2k, ×2	CA-2k	5.43M

Table 2: The Vision pre-training datasets. In sampling, $\times k$ means k copies in one epoch and "CA-2k" means class-aware sampling with at least 2K instances per class.

Model Architecture (FPN vs C4). Although [22] shows that the FPN model outperforms the C4 model for object detection, recent studies [13] demonstrate that FPN does not provide more effective region features for VL tasks than C4, which is also confirmed by our experimental results [2]. We thus conduct a set of carefully designed experiments, as to be detailed in Appendix E, and find two main reasons for this. The first is that all layers in the C4 model used for region feature extraction are pre-trained using the ImageNet dataset while the multi-layer-perceptron (MLP) head of the FPN model are not. It turns out that the VG dataset is still too small to train a good enough visual features for VL tasks and using ImageNet-pre-trained weights is beneficial. The second is due to the different network architectures (CNN vs. MLP). The convolutional head used in C4 has a better inductive bias for encoding visual information than the MLP head of FPN. Therefore, in this study we use C4 architecture for VLP.

[2]We find in our experiments that using the same training process, the X152-C4 model even produces better object detection result than the X152-FPN model. See Appendix E for details.

Model Pre-Training. Following the common practice in object detection training, we freeze the first convolution layer, the first residual block, and all the batch-norm layers. We also use several data augmentation methods, including horizontal flipping and multi-scale training. To train a detection model with the X152-C4 architecture, we initialize the model backbone from an ImageNet-5K checkpoint [39] and train for 1.8M iterations with a batch size of 16 images.

2.2. Injecting attribute information into the model

Following [2], we add an attribute branch to the pre-trained OD model, and then fine-tune the OD model on VG to inject attribute information (524 classes). Since the object representations are pre-trained in the object detection pre-training stage, we can focus the VG fine-tuning on learning attributes by picking a much larger attribute loss weight 1.25, compared to 0.5 used in [2, 13]. Thus, our fine-tuned model significantly outperforms previous models [2, 13] in detecting objects and attributes on VG.

2.3. Efficient region feature extractor for VL tasks

With a richer set of visual objects and attributes, the classical class-aware non-maximal suppression (NMS) post-processing takes a significantly larger amount of time to remove overlapped bounding boxes, making the feature extraction process extremely slow. To improve the efficiency, we replace the class-aware NMS with the class-agnostic NMS that only conducts the NMS operation once[3]. We also replace the time-consuming conv layers with dilation=2 used in [2] with conv layers without dilation. These two replacements make the region feature extraction process much faster than that in [2] without any accuracy drop on VL downstream tasks. We report the end-to-end inference time of VL models with different vision models on a Titan-X GPU and a CPU with a single thread in Table 22 in Appendix F.

In summary, the pre-trained OD model serves as the image understanding module, as in Equation (1), to produce vision presentations $(\boldsymbol{q}, \boldsymbol{v})$ for downstream VL tasks. Here, $\boldsymbol{q}$ is the set of detected object names (in text) and $\boldsymbol{v}$ is the set of region features. Each region feature is denoted as $(\hat{v}, z)$, where $\hat{v}$ is a P-dimensional representation from the input of the last linear classification layer of the detection head (*i.e.* $P = 2048$) and z is a R-dimensional position encoding of the region (*i.e.* $R = 6$)[4].

3. OSCAR+ Pre-training

The success of VLP lies in the use of a unifying model architecture for a wide range of VL tasks and the large-scale pre-training of the unified model using objectives that correlate with the performance metrics of these downstream VL tasks. In this study we pre-train an improved version of OSCAR [20], known as OSCAR+ models, to learn the joint image-text representations using image tags as anchors for image-text alignment.

3.1. Pre-training corpus

We build our pre-training corpus based on three types of existing vision and VL datasets: (1) image captioning datasets with human-annotated captions as $\boldsymbol{w}$ and machine-generated [5] image tags as $\boldsymbol{q}$, including COCO [24], Conceptual Captions (CC) [31], SBU captions [27] and flicker30k [41]; (2) visual QA datasets with questions as $\boldsymbol{w}$ and human-annotated answers as $\boldsymbol{q}$, including GQA [12], VQA [8] and VG-QAs; (3) image tagging datasets with machine-generated [6] captions as $\boldsymbol{w}$ and human-annotated tags as $\boldsymbol{q}$, including a subset of OpenImages (1.67M images). In total, the corpus contains 5.65 million unique images, 8.85 million text-tag-image triples. The detailed statistics are presented in Table 17 in the Appendix. The size of the pre-training corpus could have been significantly increased by combining large-scale image tagging datasets, such as the full set of OpenImages (9M images) and YFCC (92M images). We leave it to future work to leverage much larger corpora for model pre-training.

Loss	$(\boldsymbol{w}, \boldsymbol{q}/\boldsymbol{q}', \boldsymbol{v})$		$(\boldsymbol{w}/\boldsymbol{w}', \boldsymbol{q}, \boldsymbol{v})$	3-way contrastive	
$\boldsymbol{w}'/\boldsymbol{q}'$	All $\boldsymbol{q}$'s (OSCAR)	$\boldsymbol{q}$'s from QA	All $\boldsymbol{w}$'s	All (OSCAR+)	$\boldsymbol{q}$'s from QA
VQA (dev)	**69.8**±0.08	**70.1**±0.08	69.5±0.05	**69.8**±0.06	**69.7**±0.06
COCO-IR	73.9±0.2	**75.0**±0.2	**75.0**±0.7	**78.3**±0.3	**77.7**±0.7

Table 3: Effects of different pre-training contrastive losses on downstream tasks (R50-C4 as **Vision** module and 4-layer Transformer as **VL** module in (1)). COCO-IR metric is Image-to-Text retrieval R@1 at COCO 1K test set. **Blue** indicates the best result for a task and **Black** indicates the runner-up.

3.2. Pre-training Objectives

There are two terms in the OSCAR+ pre-training loss as in Equation (2).

$$\mathcal{L}_{\text{Pre-training}} = \mathcal{L}_{\text{MTL}} + \mathcal{L}_{\text{CL3}}. \tag{2}$$

$\mathcal{L}_{\text{MTL}}$ is the Masked Token Loss defined on the text modality ($\boldsymbol{w}$ and $\boldsymbol{q}$), following closely [20]. (See Appendix B.2 for details.) $\mathcal{L}_{\text{CL3}}$ is a novel *3-way Contrastive Loss*. Different from the binary contrastive loss used in OSCAR [20], the proposed *3-way Contrastive Loss* to effectively optimize the training objectives used for VQA [40] and text-

[3]Counting the NMS in the RPN module, there are in total 2 NMS operations in our efficient region feature extractor.

[4]It includes coordinates of the bounding boxes, and height & width.

[5]We use the same model to extract visual features.

[6]We use the captioning model released by OSCAR [20].

image matching [6][7]. As shown in Equation 3, $\mathcal{L}_{\text{CL3}}$ takes into account two types of training samples $\boldsymbol{x}$: the {caption, image-tags, image-features} triplets of the image captioning and image tagging data, and the {question, answer, image-features} triplets of the VQA data.

$$\boldsymbol{x} \triangleq (\underbrace{\boldsymbol{w}}_{\text{caption}}, \underbrace{\boldsymbol{q}, \boldsymbol{v}}_{\text{tags\&image}}) \quad \text{or} \quad (\underbrace{\boldsymbol{w}, \boldsymbol{q}}_{\text{Q\&A}}, \underbrace{\boldsymbol{v}}_{\text{image}}) \tag{3}$$

To compute contrastive losses, negative examples need to be constructed. We construct two types of negative (unmatched) triplets for the two types of training samples, respectively. One is the polluted "captions" $(\boldsymbol{w}', \boldsymbol{q}, \boldsymbol{v})$ and the other the polluted "answers" $(\boldsymbol{w}, \boldsymbol{q}', \boldsymbol{v})$. To classify whether a caption-tags-image triplet contains a polluted caption is a text-image matching task. To classify whether a question-answer-image triplet contains a polluted answer is an answer selection task for VQA. Since the encoding of [CLS] can be viewed as a representation of the triplet $(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v})$, we apply a fully-connected (FC) layer on top of it as a 3-way classifier $f(.)$ to predict whether the triplet is matched ($c = 0$), contains a polluted $\boldsymbol{w}$ ($c = 1$), or contains a polluted $\boldsymbol{q}$ ($c = 2$). The 3-way contrastive loss is defined as

$$\mathcal{L}_{\text{CL3}} = -\mathbb{E}_{(\boldsymbol{w},\boldsymbol{q},\boldsymbol{v};c)\sim\tilde{\mathcal{D}}} \log p(c|f(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v})), \tag{4}$$

where the dataset $(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v}; c) \in \tilde{\mathcal{D}}$ contains 50% matched triples, 25% $\boldsymbol{w}$-polluted triples, and 25% $\boldsymbol{q}$-polluted triples. For efficient implementation, the polluted $\boldsymbol{w}'$ is uniformly sampled from all $\boldsymbol{w}$'s (captions and questions) and $\boldsymbol{q}'$ is uniformly sampled from all $\boldsymbol{q}$'s (tags and answers) in the corpus. As demonstrated in Table 3, when only the answer-polluted triplets are used, i.e., $(\boldsymbol{w}, \boldsymbol{q}', \boldsymbol{v})$ with $\boldsymbol{q}'$ sampled from $\boldsymbol{q}$'s from QA corpus, the contrastive loss simulates closely the objective for the VQA task but not the text-image retrieval task. As a result, the pre-trained model can be effectively adapted to VQA, but not so to text-image retrieval. By contrast, the proposed 3-way contrastive loss transfers well to both tasks.

3.3. Pre-trained models

We pre-train two model variants, denoted as OSCAR+$_{\text{B}}$ and OSCAR+$_{\text{L}}$, which are initialized with parameters $\boldsymbol{\theta}_{\text{BERT}}$ of BERT base ($L = 12, H = 768, A = 12$) and large ($L = 24, H = 1024, A = 16$), respectively, where L is the number of layers, H the hidden size, and A the number of self-attention heads. To ensure that the image region features have the same input embedding size as BERT, we transform the position-augmented region features using a linear projection via matrix $\mathbf{W}$. The trainable parameters are $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\text{BERT}}, \mathbf{W}\}$. OSCAR+$_{\text{B}}$ is trained for at least 1M steps, with learning rate $1e^{-4}$ and batch size 1024. OSCAR+$_{\text{L}}$ is trained for at least 1M steps, with learning rate $3e^{-5}$ and batch size 1024. The sequence length of language tokens $[\boldsymbol{w}, \boldsymbol{q}]$ and region features $\boldsymbol{v}$ are 35 and 50, respectively.

3.4. Adapting to VL Tasks

We adapt the pre-trained models to seven downstream VL tasks, including five understanding tasks and two generation tasks. Each task poses different challenges for adaptation. We refer to Appendix C for details about the seven tasks and our fine-tuning strategies.

4. Experiments and Analysis

4.1. Main Results

To account for model parameter efficiency, we group the SoTA models in three categories: (i) SoTA$_S$ indicates the best performance achieved by small models prior to the Transformer-based VLP models. (ii) SoTA$_B$ indicates the best performance produced by VLP models of BERT base size. (iii) SoTA$_L$ indicates the best performance yielded by VLP models of BERT large size.

Table 4 gives an overview of the results of OSCAR+ with VINVL(short for VINVL) on seven VL tasks, compared to previous SoTAs[8]. VINVLoutperforms previous SoTA models on all tasks[9], often by a significantly large margin. The result demonstrates the effectiveness of the region features produced by the new OD model.

In Tables 5 to 11, we report the detailed results for each downstream task, respectively. (i) The **VQA** results are shown in Table 5, where our single OSCAR+$_{\text{B}}$ model outperforms the best ensemble model (InterBERT large [21]) on the VQA leaderboard as of Dec. 12, 2020 [10]. (ii) The **GQA** results are shown in Table 6, where OSCAR+w/VINVLis the first VLP model that outperforms the neural state machine (NSM) [11] which contains some sophisticated reasoning components deliberately designed for the task. (iii) The **Image Captioning** results on the public "Karpathy" 5k test split are shown in Table 7. Table 8 shows on a concise version of the COCO image captioning online leaderboard[11]. The online testing setting reports the results on 40K images, with 5 reference captions (c5) and 40 reference captions (c40) per image. At the time of submitting this paper, our single model achieves No.1 on the

[7] [6] uses a deep-learning-based text-image matching model to select the best caption candidate for a given image.

[8] All the (single-model) SoTAs are from the published results. For all the tables in this paper, **Blue** indicates the best result for a task, and gray background indicates results produced by VINVL.

[9] The only exception is B@4 on image captioning.

[10] VQA leaderboard: https://eval.ai/web/challenges/challenge-page/514/leaderboard/1386

[11] Image Captioning Leaderboard: https://competitions.codalab.org/competitions/3221#results

Task	VQA		GQA		Image Captioning				NoCaps		Image Retrieval			Text Retrieval			NLVR2	
	test-dev	test-std	test-dev	test-std	B@4	M	C	S	C	S	R@1	R@5	R@10	R@1	R@5	R@10	dev	test-P
SoTA$_S$	70.55	70.92	–	**63.17**	38.9	29.2	129.8	22.4	61.5	9.2	39.2	68.0	81.3	56.6	84.5	92.0	54.10	54.80
SoTA$_B$	73.59	73.67	61.58	61.62	40.5	29.7	137.6	22.8	86.58	12.38	54.0	80.8	88.5	70.0	91.1	95.5	78.39	79.30
SoTA$_L$	74.75	74.93	–	–	**41.7**	30.6	140.0	24.5	–	–	57.5	82.8	89.8	73.5	92.3	96.0	79.76	81.47
VINVL$_B$	**75.95**	**76.12**	**65.05**	**64.65**	40.9	**30.9**	**140.6**	**25.1**	**92.46**	**13.07**	**58.1**	**83.2**	**90.1**	**74.6**	**92.6**	**96.3**	**82.05**	**83.08**
VINVL$_L$	**76.52**	**76.60**	–	–	**41.0**	**31.1**	**140.9**	**25.2**	–	–	**58.8**	**83.5**	**90.3**	**75.4**	**92.9**	**96.2**	**82.67**	**83.98**
Δ	**1.77** ↑	**1.67** ↑	**3.47** ↑	**1.48** ↑	0.7 ↓	**0.5** ↑	**0.9** ↑	**0.7** ↑	**5.9** ↑	**0.7** ↑	**1.3** ↑	**0.7** ↑	**0.5** ↑	**1.9** ↑	**0.6** ↑	**0.3** ↑	**2.91** ↑	**2.51** ↑

Table 4: An overall comparison with SoTAs on seven tasks. Δ indicates the improvement over SoTA. SoTA with subscript S, B, L indicates performance achieved by small models, and models with the model size similar to BERT base and large, respectively. SoTAs: VQA is from ERNIE-VIL [42], GQA is from NSM [11], NoCaps is from VIVO [9], NLVR2 is from VILLA [7], the rest tasks are from OSCAR [20].

Method	ViLBERT	VL-BERT	VisualBERT	LXMERT	12-in-1	UNITER		OSCAR		VILLA		ERNIE-VIL		InterBERT	OSCAR+w/ VINVL	
	Base	Base	Base	Base	Base	Base	Large	Base	Large	Base	Large	Base	Large	**Ensemble***	Base	Large
Test-dev	70.63	70.50	70.80	72.42	73.15	72.27	73.24	73.16	73.61	73.59	73.69	72.62	74.75	-	**75.95**	**76.52**
Test-std	70.92	70.83	71.00	72.54	–	72.46	73.40	73.44	73.82	73.67	74.87	72.85	74.93	76.10	**76.12**	**76.60**

Table 5: Evaluation results on VQA. * denotes the No.1 ensemble model of InterBERT Large on the VQA leaderboard.

Method	LXMERT	MMN [3]	12-in-1	OSCAR$_B$	NSM [11]	OSCAR+$_B$ w/ VINVL
Test-dev	60.00	–	–	**61.58**	–	**65.05**
Test-std	60.33	60.83	60.65	61.62	**63.17**	**64.65**

Table 6: Evaluation results on GQA.

Method	cross-entropy optimization				CIDEr optimization			
	B@4	M	C	S	B@4	M	C	S
BUTD [2]	36.2	27.0	113.5	20.3	36.3	27.7	120.1	21.4
VLP [44]	36.5	28.4	117.7	21.3	39.5	29.3	129.3	23.2
AoANet [10]	37.2	28.4	119.8	21.3	38.9	29.2	129.8	22.4
OSCAR$_B$ [20]	36.5	30.3	123.7	23.1	40.5	29.7	137.6	22.8
OSCAR$_L$ [20]	37.4	**30.7**	127.8	**23.5**	**41.7**	30.6	140.0	24.5
OSCAR+$_B$ w/ VINVL	**38.2**	30.3	**129.3**	**23.6**	40.9	**30.9**	**140.4**	**25.1**
OSCAR+$_L$ w/ VINVL	**38.5**	**30.4**	**130.8**	23.4	**41.0**	**31.1**	**140.9**	**25.2**

Table 7: Image captioning evaluation results (single model) on COCO "Karpathy" test split. (Note: B@4: BLEU@4, M: METEOR, C: CIDEr, S: SPICE.)

Method	BLEU@4		METEOR		ROUGE-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40
BUTD [2]	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
AoANet [10]	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
X-Transformer [28]	40.3	72.4	29.6	39.2	59.5	75.0	131.1	133.5
OSCAR+ w/ VINVL	**40.4**	**74.9**	**30.6**	**40.8**	**60.4**	**76.8**	**134.7**	**138.7**

Table 8: Leaderboard of the state-of-the-art image captioning models on the COCO online testing.

entire leaderboard, outperforming all 263 models, including many ensemble (and anonymous) models. (*iv*) The Novel Object Captioning (**NoCaps**) results are shown in Table 9. Without any VLP, i.e. by directly training a BERT-based captioning model on COCO, the model with our new visual features (denoted as VinVL) already surpasses the human performance in CIDEr[12]. By adding VIVO [9] pre-training, our VinVL improves the original VIVO result by 6 CIDEr points and creates a new SoTA. (*v*) Overall, on all these

[12]NoCaps leaderboard: https://eval.ai/web/challenges/challenge-page/355/leaderboard/1011

Method	CIDEr	SPICE	CIDEr	SPICE
	Validation Set		Test Set	
UpDown$^+$	74.3	11.2	73.1	11.2
OSCAR$_B$*	81.1	11.7	78.8	11.7
OSCAR$_L$*	83.4	11.4	80.9	11.3
Human [1]	87.1	**14.2**	85.3	**14.6**
VIVO* [9]	88.3	12.4	**86.6**	12.4
VinVL*	**90.9**	12.8	85.5	12.5
VinVL+VIVO	**98.0**	**13.6**	**92.5**	**13.1**

Table 9: NoCaps evaluation "overall" results. All the models are trained on COCO without additional image-caption pairs following the restriction of NoCaps. (UpDown$^+$ is UpDown+ELMo+CBS, the models with * is +SCST+CBS, VinVL+VIVO is with SCST only.) We refer to Table 18 in Appendix C for results on different subsets.

Method	BERT	Text Retrieval			Image Retrieval		
		R@1	R@5	R@10	R@1	R@5	R@10
1K Test Set							
Unicoder-VL [18]	B	84.3	97.3	99.3	69.7	93.5	97.2
OSCAR	B	88.4	99.1	99.8	75.7	95.2	98.3
	L	89.8	**98.8**	**99.7**	**78.2**	**95.8**	**98.3**
OSCAR+ w/ VINVL	B	**89.8**	**98.8**	**99.7**	**78.2**	95.6	98.0
	L	**90.8**	**99.0**	**99.8**	**78.8**	**96.1**	**98.5**
5K Test Set							
Unicoder-VL [18]	B	62.3	87.1	92.8	46.7	76.0	85.3
UNITER [4]	B	63.3	87.0	93.1	48.4	76.7	85.9
	L	66.6	89.4	94.3	51.7	78.4	86.9
OSCAR	B	70.0	91.1	95.5	54.0	80.8	88.5
	L	73.5	92.2	96.0	57.5	82.8	89.8
OSCAR+ w/ VINVL	B	**74.6**	**92.6**	**96.3**	**58.1**	**83.2**	**90.1**
	L	**75.4**	**92.9**	**96.2**	**58.8**	**83.5**	**90.3**

Table 10: Text and Image retrieval evaluation on the COCO 1K and 5K test sets. (B for Base, L for Large)

tasks (VQA in Table 5, Image Captioning in Table 7, NoCaps in Table 9, Image-Text Retrieval in Table 10, NLVR2 in Table 11), we show that OSCAR+$_B$ can match or outperform previous SoTA large models, and OSCAR+$_L$ substantially uplifts the SoTA.

Method	MAC	VisualBERT	LXMERT	12-in-1	UNITER		OSCAR		VILLA		OSCAR+w/ VINVL	
		base	base	base	base	large	base	large	base	large	base	large
Dev	50.8	67.40	74.90	–	77.14	78.40	78.07	79.12	78.39	79.76	**82.05**	**82.67**
Test-P	51.4	67.00	74.50	78.87	77.87	79.50	78.36	80.37	79.47	81.47	**83.08**	**83.98**

Table 11: Evaluation results on NLVR2.

vision \ vl	no VLP	OSCAR$_B$ [20]	OSCAR+$_B$ (ours)
R101-C4 [2]	68.52 ±0.11	72.38	72.46±0.05
VinVL (ours)	71.34 ±0.17	–	74.90±0.05

Table 12: Effects of vision (V) and vision-language (VL) pre-training on VQA.

data \ model	R50-FPN	R50-C4	R101-C4 [2]	X152-C4
VG	67.35±0.26	67.86±0.31	68.52 ±0.11	69.10±0.06
4Sets→VG	68.3±0.11	68.39±0.16	–	71.34 ±0.17

Table 13: Ablation of model size and data size on training vision models.

Model	R50-FPN		R50-C4		X152-C4	
Pre-training dataset	ImageNet	4Sets	ImageNet	4Sets	ImageNet5k	4Sets
COCO mAP	40.2 [39]	44.78*	38.4 [39]	42.4	42.17	50.51
VG obj mAP^{50}	9.6	11.3	9.6	12.1	11.2	13.8
attr mAP with gt boxes	5.4	5.5	6.3	6.1	6.6	7.1

* Since our four pre-training datasets contain Objects365, it is not surprising that we obtain better results than 42.3 mAP^{50} in [30], which is obtained by pre-training on Objects365.

Table 14: Effect of vision pre-training on object detection tasks.

4.2. Ablation Analysis

We select the VQA task for the ablation study because its evaluation metric is well-defined and the task has been used as a testbed for all VLP models. To assist our analysis, we create a local validation set, vqa-dev, out of the standard validation set to select the best model during training for evaluation. vqa-dev contains randomly sampled 2K images and their corresponding questions, amounting to 10.4K image-QA pairs in total. Except for Table 4 and 5, all our VQA results are reported on this vqa-dev set. Unless otherwise specified, the reported STD is half of the difference of two runs of the VQA training with different random seeds.

In VQA, the VL model $y = \mathbf{VL}(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v})$ has $\boldsymbol{w}$ as the question and y as the answer. We focus on studying the effect of visual features $\boldsymbol{v}$ produced by different Vision models $\mathbf{Vision}(Img)$ to better understand their relative contribution in the VQA performance. To eliminate the impact of using different tags $\boldsymbol{q}$, we use the same tags in the VQA models of OSCAR [20]. All the ablation experiments are conducted using models of the BERT-base size.

How much do the V and VL matter to the SoTA? Table 12 shows the VQA results with different vision models, i.e., R101-C4 model from [2] and our X152-C4 model pre-trained with 4 datasets (VinVL), and with different VLP methods, i.e., no VLP, OSCAR [20] and our OSCAR+. Taking the OSCAR$_B$ model with R101-C4 features as the baseline, the OSCAR+$_B$ model with our X152-C4 features improves the absolute accuracy from 72.38 to 74.90, in which the OSCAR+ pre-training contributes 5% of the gain (i.e., 72.38 → 72.46) and the vision pre-training (improved visual features) 95% (i.e., 72.46 → 74.90). This demonstrates that vision representations matter significantly in VLP and downstream tasks.

Taking the "no VLP" model with R101-C4 features as the baseline, Table 12 shows that the gains of VinVL ($71.34 - 68.52 = 2.82$) and VLP ($72.46 - 68.52 = 3.94$) are additive ($74.90 - 68.52 \approx 2.82 + 3.94$). This is intuitive because vision pre-training and VLP improve the Vision model $\mathbf{Vision}(Img)$ and VL model $\mathbf{VL}(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v})$ separately. This also indicates that our pre-trained vision model can be utilized in any VL models by directly replacing their vision models, such as R101-C4 [2], with ours.

How much do data and model sizes matter to the new vision model? The improvement of VQA from R101-C4 [2] to VinVL (ours) in Table 12 is a compound effect of increasing model size (from R101-C4 to X152-C4) and data size (from VG to our merged four OD datasets). Table 13 shows the ablation of the two factors without VLP. Although VG's large object and attribute vocabulary allows to learn rich semantic concepts, VG does *not* contain large amounts of annotations for effective training of deep models. Vision models trained using the merged four OD datasets perform much better than VG-only-trained models, and the improvement is larger with the increase of the model size.[13]

How much does OD model architecture matter? The choice of model architecture affects the VQA performance. Table 13 shows that R50-FPN under-performs R50-C5 when they are trained only on VG; but the performance gap diminishes when both are trained on the merged dataset (4Sets). A detailed comparison between FPN and C4 architectures is presented in Appendix E.

How much does OD pre-training matter for object detection tasks? Table 14 presents the object detection results on COCO and the object-attribute detection results on VG (1594 object classes, 524 attribute classes). The results show that OD pre-training benefits the object detection tasks. Note that the mAP on VG is much lower than that on

[13] The R101-C4 model in Table 13 is exactly the VG-pre-pretrained model from [2]. We do not train this model on our merged OD dataset because this model architecture is old-fashioned and is slow to train.

Dataset name #obj & #attr	ImageNet 1000 & 0	VG-obj 317 & 0	VG w/o attr 1594 & 0	VG [2] 1600 & 400	VG 1594 & 524	4Sets→VG 1848 & 524
R50-C4 + BERT$_B$	66.13±0.04	64.25±0.16	66.51±0.11	67.63±0.25	67.86±0.31	68.39±0.16

Table 15: Effect of object-attribute vocabulary. We use all grid features (maximal 273) for the ImageNet classification model (first column), and maximal 50 region features for OD models (other columns).

typical OD datasets (such as COCO) due to two reasons: (1) VG contains a large number of object classes with limited and extremely unbalanced annotations, (2) there are many missing annotations in the VG evaluation data.[14] Although the mAP numbers are low, the detection result using X152-C4 is reasonably good; see Appendix A for visualizations. We also see that FPN models perform consistently worse in attribute detection than C4 models, neither do FPN models show any advantage in object detection on VG. This contributes to the inferior performance of FPN, compared to C4, on downstream VL tasks, as discussed in Section 2.1.

How much does the diversity of visual concepts, i.e., object and attribute vocabularies, matter? We directly train vision models on different datasets, including (1) standard ImageNet with 1K classes (ImageNet), (2) Visual Genome with 317 object classes (VG-obj) that are shared with COCO 80 classes and OpenImagesV5 500 classes, (3) VG with all 1594 object classes (VG w/o attr), (4) VG with 1594 object classes and 524 attribute classes (VG), and (5) the merged OD dataset (4Sets) for pre-training and VG for fine-tuning. For all the OD models (the last four columns in Table 15), we initialize the OD training with an ImageNet-pre-trained classification model, and use maximal 50 *region* features per image as input to the VL fusion module. For the ImageNet pre-trained classification model (the second column in Table 15), we use all the *grid* features (maximal 273) for each image. The results show that

- In general, vocabularies with richer objects lead to better VQA results: VG-obj < ImageNet < VG w/o attr. The VG-obj vocabulary contains 79 of 80 COCO classes (only missing `potted plant`) and 313 of 500 OpenImagesV5 classes, and is a good approximation of common object classes of typical OD tasks. However, our results show that this vocabulary is not rich enough for VL tasks because it misses many important visual concepts (e.g., `sky`, `water`, `mountain`, etc.) which are crucial for VL tasks, as also illustrated by the comparison of detected regions in Figure 1. [15].
- Attribute information is crucial to VL tasks: models trained with attributes (VG and 4Sets→VG) are significantly better than those without attributes.
- Even for the small vision model R50-C4, vision pre-training improves visual features for VQA, i.e., 4Sets→VG is the best performer.

In Table 16, we use different kinds of region proposals to extract image features. COCO groundtruth object regions (GT-Obj, 80 classes) and object-stuff regions (GT-Obj&Stuff, 171 classes) are perfect in terms of localization, but their vocabulary sizes are limited. Regions proposed by VG-trained models ([2] and VinVL) are imperfect in localization but using a larger vocabulary. For the VQA task, COCO GT boxes are much worse than the proposals generated by VG-trained models. The result demonstrates the difference between the typical OD tasks and the OD tasks in VL: OD in VL requires much richer visual semantics to align with the rich semantics in the language modality. This further echoes our claim that an image understanding module trained using richer vocabularies performs better for VL tasks.

model \ region	GT-Obj	GT-Obj&Stuff	Anderson et al. [2]	VinVL (ours)
Anderson et al. [2]	63.81 ±0.94	66.68 ±0.16	68.52 ±0.11	69.05 ±0.06
VinVL (ours)	65.60 ±0.21	68.13 ±0.26	70.25 ±0.05	71.34 ±0.17

Table 16: Effect of different region proposals on VQA.

5. Conclusion

In this paper we have presented a new recipe to pre-train an OD model for VL tasks. Compared to the most widely used *bottom-up and top-down* model [2], the new model is bigger, better-designed for VL tasks, and pre-trained on much larger text-image corpora, and thus can generate visual features for a richer collection of visual objects and concepts that are crucial for VL tasks. We validate the new model via a comprehensive empirical study where we feed the visual features to a VL fusion model which is pre-trained on a large-scale paired text-image corpus and then fine-tuned on seven VL tasks. Our results show that the new OD model can substantially uplift the SoTA results on all seven VL tasks across multiple public benchmarks. Our ablation study shows that the improvement is mainly attributed to our design choices regarding diversity of object categories, visual attribute training, training data scale, model size, and model architecture.

[14] As a reference, the R101-C4 model from [2] on VG with 1600 objects and 400 attributes has mAP of 8.7/7.8 evaluated in our code, whereas it was reported as 10.2/7.8 due to differences in OD evaluation pipeline.

[15] Using the same training procedure on VG, we trained an R50-C4 model on the OpenImagesV5 dataset (500 classes). Using the region features produced by this model, the VQA performance is 63.55±0.14. The result is slightly worse than that of VG-obj because both VG and VQA images are from the COCO dataset but OpenImages images are not.

References

[1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *ICCV*, 2019. 1, 6, 16, 17

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 1, 2, 3, 4, 6, 7, 8, 11, 13, 14, 15, 16, 17, 18, 20, 21

[3] Wenhu Chen, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu. Meta module network for compositional visual reasoning. *arXiv preprint arXiv:1910.03230*, 2019. 6, 14

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019. 1, 6, 16

[5] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612*, 2(7):8, 2017. 16

[6] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1473–1482, 2015. 5

[7] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *arXiv preprint arXiv:2006.06195*, 2020. 6

[8] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 1, 4, 14

[9] Xiaowei Hu, Xi Yin, Kevin Lin, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. Vivo: Surpassing human performance in novel object captioning with visual vocabulary pre-training. *arXiv preprint arXiv:2009.13682*, 2020. 2, 6, 17

[10] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, 2019. 6, 16

[11] Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. In *NeurIPS*, 2019. 1, 5, 6

[12] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. *arXiv preprint arXiv:1902.09506*, 2019. 1, 4, 14

[13] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276, 2020. 3, 4, 18

[14] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 16

[15] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 1

[16] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 1, 16

[17] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *ECCV*, 2018. 16

[18] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019. 1, 6, 16

[19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 1

[20] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 1, 2, 4, 6, 7, 12, 14, 15, 16, 19

[21] Junyang Lin, An Yang, Yichang Zhang, Jie Liu, Jingren Zhou, and Hongxia Yang. Interbert: Vision-and-language interaction for multi-modal pretraining. *arXiv preprint arXiv:2003.13198*, 2020. 5

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 3

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 4, 14, 16

[25] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. VilBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 1

[26] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-Task vision and language representation learning. *arXiv preprint arXiv:1912.02315*, 2019. 16

[27] Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011. 4

[28] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10971–10980, 2020. 6

[29] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. 16

[30] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 8430–8439, 2019. 1, 7

[31] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Annual Meeting of the Association for Computational Linguistics*, 2018. 4, 16

[32] Botian Shi, Lei Ji, Pan Lu, Zhendong Niu, and Nan Duan. Knowledge aware semantic concept expansion for image-text matching. In *IJCAI*, 2019. 16

[33] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 1

[34] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018. 1, 16

[35] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. *EMNLP*, 2019. 1

[36] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: A simple and strong anchor-free object detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3

[37] Yaxiong Wang, Hao Yang, Xueming Qian, Lin Ma, Jing Lu, Biao Li, and Xin Fan. Position focused attention network for image-text matching. *arXiv preprint arXiv:1907.09748*, 2019. 16

[38] Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. CAMP: Cross-Modal adaptive message passing for text-image retrieval. In *ICCV*, 2019. 16

[39] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4, 7, 19, 21

[40] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016. 4

[41] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 4

[42] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. *arXiv preprint arXiv:2006.16934*, 2020. 6

[43] Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, and Yi-Dong Shen. Dual-path convolutional image-text embedding with instance loss. *arXiv preprint arXiv:1711.05535*, 2017. 16

[44] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. *AAAI*, 2020. 1, 6, 15, 16