
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 18/48 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)
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:

\[(q, v) = \text{Vision}(Img), \quad y = \text{VL}(w, q, v), \quad (1)\]

where \(Img\) and \(w\) are the inputs of the vision and language modalities, respectively. The output of the Vision module consists of \(q\) and \(v\). \(q\) is the semantic representation of the image, such as tags or detected objects, and \(v\) 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 \(v\), while the recently proposed OSCAR [20] model shows that \(q\) can serve as anchors for learning better vision-language joint representations and and thus can improve the performance on various VL tasks. \(w\) and \(y\) of the VL module of Equation (1) vary among different VL tasks. In VQA, \(w\) is a question and \(y\) is an answer to be predicted. In text-image retrieval, \(w\) 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, $w$ 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.

<table>
<thead>
<tr>
<th>Source</th>
<th>VG</th>
<th>COCO w/ stuff</th>
<th>Objects365</th>
<th>OpenImagesV5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image classes</td>
<td>97k</td>
<td>111k</td>
<td>609k</td>
<td>1.67M</td>
<td>2.49M</td>
</tr>
<tr>
<td>Sampling $\times k$</td>
<td>$\times 8$</td>
<td>$\times 8$</td>
<td>CA-2k, $\times 2$</td>
<td>CA-2k</td>
<td>5.43M</td>
</tr>
</tbody>
</table>

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.

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.

---

2We 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.

5581
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 \((q, v)\) for downstream VL tasks. Here, \(q\) is the set of detected object names (in text) and \(v\) 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 \(w\) and machine-generated \(^5\) image tags as \(q\), including COCO [24], Conceptual Captions (CC) [31], SBU captions [27] and flicker30k [41]; (2) visual QA datasets with questions as \(w\) and human-annotated answers as \(q\), including GQA [12], VQA [8] and VG-QAs; (3) image tagging datasets with machine-generated \(^6\) captions as \(w\) and human-annotated tags as \(q\), 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.

<table>
<thead>
<tr>
<th>Loss (w'/q')</th>
<th>(w, q/q', v)</th>
<th>(w,w'/q, q)</th>
<th>(w/w',q,v)</th>
<th>3-way contrastive</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQA (dev)</td>
<td>69.3±0.06</td>
<td>70.1±0.08</td>
<td>69.5±0.05</td>
<td>69.8±0.06</td>
</tr>
<tr>
<td>COCO-IR</td>
<td>73.9±0.2</td>
<td>75.0±0.2</td>
<td>73.0±0.7</td>
<td>73.4±0.3</td>
</tr>
</tbody>
</table>

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).

\[
\mathcal{L}_{\text{Pre-training}} = \mathcal{L}_{\text{MTL}} + \mathcal{L}_{\text{CL3}}.
\]  

\(\mathcal{L}_{\text{MTL}}\) is the Masked Token Loss defined on the text modality \((w, q)\), 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-...
image matching [6]. As shown in Equation 3, $\mathcal{L}_{\text{CLS}}$ takes into account two types of training samples $x$: 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.

$$x \triangleq (w, q, v) \text{ or } (w, q', v)$$ 

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” $(w', q, v)$ and the other the polluted “answers” $(w, q', v)$. 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 $(w, q, v)$, 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 $w$ ($c = 1$), or contains a polluted $q$ ($c = 2$). The 3-way contrastive loss is defined as

$$\mathcal{L}_{\text{CLS}} = -\mathbb{E}_{(w, q, v, c) \in \tilde{D}} \log p(c | f(w, q, v)),$$

where the dataset $(w, q, v, c) \in \tilde{D}$ contains 50% matched triplets, 25% $w$-polluted triplets, and 25% $q$-polluted triplets. For efficient implementation, the polluted $w'$ is uniformly sampled from all $w$'s (captions and questions) and $q'$ is uniformly sampled from all $q$'s (tags and answers) in the corpus. As demonstrated in Table 3, when only the answer-polluted triplets are used, i.e., $(w, q', v)$ with $q'$ sampled from $q$'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$^+$ and OSCAR$^+$+, which are initialized with parameters $\theta_{\text{BERT}}$ 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 $W$. The trainable parameters are $\theta = \{\theta_{\text{BERT}}, W\}$. OSCAR$^+$ is trained for at least 1M steps, with learning rate $1e^{-4}$ and batch size 1024. OSCAR$^+$+ is trained for at least 1M steps, with learning rate $3e^{-5}$ and batch size 1024. The sequence length of language tokens $[w, q]$ and region features $v$ 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 VINV$L$(short for VINV$L$) on seven VL tasks, compared to previous SoTAs. VINV$L$ outperforms previous SoTA models on all tasks, 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$^+$+ model outperforms the best ensemble model (InterBERT large [21]) on the VQA leaderboard as of Dec. 12, 2020. (ii) The GQA results are shown in Table 6, where OSCAR$^+$+ 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. 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

---

5853
Image Retrieval

cross-entropy optimization

Text Retrieval

CIDEr SPICE

Table 4: An overall comparison with SoTAs on seven tasks. \( \Delta \) 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].

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

Table 6: Evaluation results on GQA.

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

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

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.

Table 10: Text and Image retrieval evaluation on the COCO 1K and 5K test sets. (B for Base, L for Large)
Table 11: Evaluation results on NLVR2.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAC</th>
<th>VisualBERT</th>
<th>LXMERT</th>
<th>UNITER</th>
<th>OSCAR</th>
<th>VILLA</th>
<th>OSCAR+V/ VinVL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>base</td>
<td>base</td>
<td>base</td>
<td>base</td>
<td>large</td>
<td>large</td>
<td>large</td>
</tr>
<tr>
<td>Dev</td>
<td>50.8</td>
<td>67.40</td>
<td>74.90</td>
<td>–</td>
<td>77.14</td>
<td>78.40</td>
<td>78.07</td>
</tr>
<tr>
<td>Test-P</td>
<td>51.4</td>
<td>67.00</td>
<td>74.50</td>
<td>78.87</td>
<td>77.87</td>
<td>79.50</td>
<td>78.36</td>
</tr>
<tr>
<td></td>
<td>80.37</td>
<td>79.47</td>
<td>81.47</td>
<td>82.05</td>
<td>82.67</td>
<td>83.08</td>
<td>83.98</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Vision</th>
<th>VL</th>
<th>OSCAR</th>
<th>OSCAR+VL</th>
</tr>
</thead>
<tbody>
<tr>
<td>R101-C4 [2]</td>
<td>no VLP</td>
<td>68.52 ±0.11</td>
<td>72.38 ±0.05</td>
</tr>
<tr>
<td>VinVL (ours)</td>
<td>–</td>
<td>71.34 ±0.17</td>
<td>74.90 ±0.05</td>
</tr>
</tbody>
</table>

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 \( y = VL(w, q, v) \) has \( w \) as the question and \( y \) as the answer. We focus on studying the effect of visual features \( v \) produced by different Vision models Vision(Img) to better understand their relative contribution in the VQA performance. To eliminate the impact of using different tags \( q \), 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 model with R101-C4 features as the baseline, the OSCAR+ 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 \( \rightarrow \) 72.46) and the vision pre-training (improved visual features) 95% (i.e., 72.46 \( \rightarrow \) 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 Vision(Img) and VL model VL(w, q, 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-C4 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-trained 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.
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. 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 V inVL) 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.

### 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).

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>#obj &amp; #attr</th>
<th>ImageNet</th>
<th>VG-obj</th>
<th>VG w/o attr</th>
<th>VG [2]</th>
<th>VG</th>
<th>4Sets→VG</th>
</tr>
</thead>
<tbody>
<tr>
<td>R50-C4 + BERT</td>
<td>1000 &amp; 0</td>
<td>66.13±0.04</td>
<td>64.25±0.16</td>
<td>66.51±0.11</td>
<td>67.63±0.25</td>
<td>67.86±0.31</td>
<td>68.39±0.16</td>
</tr>
</tbody>
</table>

### Table 16: Effect of different region proposals on VQA.

<table>
<thead>
<tr>
<th>Model</th>
<th>GT-Obj</th>
<th>GT-Obj&amp;Stuff</th>
<th>Anderson et al. [2]</th>
<th>V inVL (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anderson et al. [2]</td>
<td>63.81±0.94</td>
<td>66.68±0.16</td>
<td>68.52±0.11</td>
<td>69.05±0.06</td>
</tr>
<tr>
<td>V inVL (ours)</td>
<td>65.60±0.21</td>
<td>68.13±0.26</td>
<td>70.25±0.05</td>
<td>71.34±0.17</td>
</tr>
</tbody>
</table>

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.
References


[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.