Training Vision-Language Transformers from Captions Alone

Liangke Gui\(^1\)\(^*\) 
Alex Hauptmann\(^1\) 
Yonatan Bisk\(^1,2\) 
Qiuyuan Huang\(^2\) 
Jianfeng Gao\(^2\)

\(^1\)Carnegie Mellon University 
\{liangkeg, alex, ybisk\}@cs.cmu.edu 
\(^2\)Microsoft Research, Redmond 
\{qihu, jfgao\}@microsoft.com

Abstract

We show that Vision-Language Transformers can be learned without human labels (e.g. class labels, bounding boxes, etc). Existing work, whether explicitly utilizing bounding boxes \(^1\)\(^2\)\(^3\) or patches \(^4\), assumes that the visual backbone must first be trained on ImageNet \(^5\) class prediction before being integrated into a multimodal linguistic pipeline. We show that this is not necessary and introduce a new model Vision-Language from Captions (VLC) built on top of Masked Auto-Encoders \(^6\) that does not require this supervision. In fact, in a head-to-head comparison between ViLT, the current state-of-the-art patch-based vision-language transformer which is pretrained with supervised object classification, and our model, VLC, we find that our approach 1. outperforms ViLT on standard benchmarks, 2. provides more interpretable and intuitive patch visualizations, and 3. is competitive with many larger models that utilize ROIs trained on annotated bounding-boxes. Code and pretrained models are released at \url{https://github.com/guilk/VLC}.

1 Introduction

Should vision guide language understanding or does language structure visual representations? Vision-language transformers have put language first. Most popular vision-language transformers \(^1\)\(^2\)\(^7\)\(^3\) only integrate vision from selected bounding boxes extracted by pretrained ImageNet \(^5\) classifiers. In this paradigm, the bag of visual tokens are embedded into an existing linguistic space (i.e. the lexical embeddings of BERT \(^8\)). The introduction of ViT \(^9\) empowered the community to flip the paradigm. Notably, ViLT \(^4\) initializes with ViT \(^9\), so the initial semantic representation is vision based and language must project into the patch space. This flipped paradigm places visual representations as the initial conceptual space to which language must adhere. Additionally, there are engineering benefits to this new paradigm as it removes the computationally expensive need for ROI extraction. However, because ViT is trained with supervised class labels, its representation may be constrained by the limited concepts ImageNet covers and yet the space is still somewhat linguistic in nature when initialized and requires expensive data annotation, a hindrance to scaling to arbitrarily many visual classification categories. We take the important next

\(^*\) Work done when Liangke interned at Microsoft Research, Redmond.

Preprint. Under review.
step and remove the need for supervised pretraining. A truly unsupervised visual semantics is learned via Masked Auto-Encoders before language is integrated. This leads to both a better performing and more general model. In addition, every component can be improved and scaled with unsupervised and weakly aligned found data – removing the need for future annotation efforts while still scaling to open-vocabulary domains in the wild.

Our Vision-Language from Captions (VLC) model matches or outperforms nearly all vision-language transformers despite being 1. Smaller, 2. Avoiding use of ROIs, and 3. Not leveraging supervised pretraining. We evaluate across several popular benchmarks in addition to retrieval and probing. The model performance also appears to continue to improve with data scaling, and as it relies only on weak alignment of image-text pairs, future work with access to large compute may be able to continue driving up performance. Finally, we provide several analyses on the underlying patch/lexical representations to understand what our models are learning and guide future VL transformer research.

2 Related Work

Vision-Language Modeling. Based on how they encode images, most existing works on vision-language modeling fall into three categories. The first category focuses on using pre-trained object detectors to extract region-level visual features (e.g., by Faster R-CNN). In particular, OSCAR and VinVL further boost the performance by feeding additional image tags into the transformer model. However, extracting region-level features requires pretrained object detectors with high-resolution inputs that can be time-consuming. To tackle these two issues, the second category proposes to encode images by using grid features from convolutional neural networks. SOHO first discretize the grid features by a learnable vision dictionary, and feed the discretized features to their cross-modal module. The third category uses a Vision Transformer (ViT) as the image encoder and design different objective functions for vision-language pretraining. To minimize the computation overhead, ViLT adopts a linear projection layer to encode images, but lags behind the state-of-the-art performance. In our work, we follow ViLT by using a linear projection layer to encode images that is different from previous work with complex ResNe(X)t or object detectors. We investigate how to pretrain a ViT-based model in an end-to-end manner that closes the performance gap while maintaining fast inference speed.

Masked Language Modeling. Masked language modeling (MLM) and its auto-regressive counterparts are widely used in natural language processing for learning text representations. MLM trains a model to predict a random sample of input tokens that have been masked in a multi-class setting. In vision-language modeling, we randomly mask some of the input tokens, and the model is trained to reconstruct the original tokens given the masked tokens and its corresponding visual inputs.

Masked Image Modeling. Masked image modeling (MIM) is a pretext task to learn representations from images corrupted by masking. Inspired by the success of masked language modeling (MLM) in NLP, different masked prediction objectives have been proposed for image tasks. iGPT predicts unknown pixels of a sequence. ViT predicts mean colors of masked patches. BEiT proposes to use a pre-trained discrete variational autoencoder (dVAE) to encode masked patches. MaskFeat predicts HoG features of the masked image regions. SimMIM and MAE predict RGB values of raw pixels by direct regression. MIM has also been explored in the field of vision-language representation learning by either regressing the masked feature values or predicting a distribution over semantic classes for corresponding image region.

3 Method

3.1 Model Architecture

Our aim is a parameter-efficient vision-language transformer without the need for supervised pretraining. To this end, we use a ViT-based framework to learn multi-modal representations by 1) intra-modal reconstruction through masked image/language modeling; 2) inter-modal alignment through image-text matching. The architecture of our proposed VLC framework is illustrated in Figure. VLC consists of a modality-specific projection module 3.2, a multi-modal encoder 3.3, and
Theyellowandbluebirdisstandingonabranch…a[MASK]
The[CLS]
Wordembedding
Modality-specificembeddingPatchpositionembedding
Modality-specificProjectionMultimodalEncoderTask-specificDecoder

three task-specific decoders. We aim for minimal visual and textual embedding designs during pretraining. The red and blue arrows are the information flows of image and text, respectively.

3.2 Modality-specific Projection Module

While most of existing methods rely on complex ResNeXt or object detection components, we use a trainable linear projection layer to map flattened visual patches to the visual embedding space. The patch embeddings are represented as \( v = \{v_1, \ldots, v_n\} \in \mathbb{R}^{n \times d} \), where \( n \) is the number of image patches and \( d \) is the hidden dimension of our model. For text embedder, we follow BERT to tokenize the input sentence into WordPieces. We then adopt a word embedding lookup layer to project tokenized words to the textual embedding space. Here we use \( w = \{w_{CLS}, w_1, \ldots, w_m\} \in \mathbb{R}^{m \times d} \) to represent the token embeddings, where \( m \) is the number of tokens and the special token \( \text{CLS} \) denotes the start of the token sequence. We encode patch and token positions separately by \( v^{\text{pos}} \in \mathbb{R}^{1 \times d} \) and \( w^{\text{pos}} \in \mathbb{R}^{1 \times d} \). We use \( v^{\text{type}} \in \mathbb{R}^{1 \times d} \) and \( w^{\text{type}} \in \mathbb{R}^{1 \times d} \) as modality-type embeddings to distinguish the modality difference between patch and token embeddings. The final representations of each patch \( v_i \) and token \( w_j \) are calculated as

\[
\hat{v}_i = \text{LayerNorm}(v_i + v^{\text{pos}}_i + v^{\text{type}}) \quad \text{and} \quad \hat{w}_j = \text{LayerNorm}(w_j + w^{\text{pos}}_j + w^{\text{type}}).
\]

3.3 Multi-modal Encoder

To learn the contextual representations from both visual and textual modality, we follow single-stream approaches and use the ViT-B/16 architecture as our multi-modal encoder. ViT-B/16 consists 12 alternating layers of multiheaded self-attention (MSA) and MLP blocks. LayerNorm comes before every block and residual connections after after every block.

We use a merged-attention mechanism to fuse the visual and textual modalities. More specifically, we concatenate the token and patch embeddings together as \( \{\hat{v}_{CLS}, \hat{v}_1, \ldots, \hat{v}_{m}, \hat{w}_1, \ldots, \hat{w}_n\} \), then feed them into the transformer blocks to get the contextual representations \( \{h_{CLS}, h^v_1, \ldots, h^w_1, \ldots, h^v_n\} \). Compared with dual-stream approaches, our model design is more parameter-efficient, as the same set of parameters are shared across modalities. As a key difference from existing approaches, we initialize our model with MAE pretrained on ImageNet-1K with no labels.
3.4 Pretraining Objectives

To learn a universal visual and textual representation for vision-and-language tasks, we apply self-supervised methods to pre-train a model on a large aggregated dataset. Unlike previous approaches that only mask text tokens, we randomly mask both image patches and text tokens simultaneously. We train our model with three objectives: masked image modeling (MIM), masked language modeling (MLM) and image-text matching (ITM).

Masked Language Modeling. In language pretraining, MLM randomly masks input tokens, and the model is trained to reconstruct the original tokens based on unmasked context. Following BERT [8], we randomly mask text tokens with a probability of 0.15, and replace the masked ones \(w_m\) with a special token [MASK]. The goal is to predict the masked tokens based on both non-masked text tokens \(w_{\backslash m}\) and image patches \(v_{\backslash m}\). The learning target \(L_{MLM}\) can be formulated as

\[
L_{MLM} = -E_{(w, v) \sim D} \log p(w_{\backslash m} | w_{\backslash m}, v_{\backslash m}).
\]

We use a linear layer with default parameters [29] as the MLM head to output logits over the vocabulary, which are used to compute the negative log likelihood loss for the masked text tokens.

Masked Image Modeling. Existing approaches explore MIM either by regressing the masked features values [1, 4, 20] or by predicting a distribution over semantic classes for a certain image region [1, 4, 19]. In contrast, we follow MAE [6] to randomly mask image patches with a probability of 0.6, and reconstruct the missing pixels based on both non-masked tokens \(w_{\backslash m}\) and patches \(v_{\backslash m}\). The learning target \(L_{MIM}\) can be formulated as

\[
L_{MIM} = E_{(w, v) \sim D} f(v_{\backslash m} | w_{\backslash m}, v_{\backslash m}),
\]

where the feature regression objective \(f\) is to regress the masked image patch representations to pixel values. We use 8-layer transformer as the MIM head \(r\). For a masked image patch \(v_j\), the objective \(f\) can be formulated as: \(f(v_j | w_{\backslash m}, v_{\backslash m}) = ||r(h_j^v) - v_j||^2\). Each output of the MIM head is a vector of pixel values representing a patch.

Image-Text Matching. Given a batch of image and text pairs, the ITM head identifies if the sampled pair is aligned. We randomly replace the aligned image with a different one with a probability of 0.5. We use the special token [CLS] as the fused representation of both modalities, and feed \(h_{CLS}\) to the ITM head. The learning target \(L_{ITM}\) can be formulated as

\[
L_{ITM} = -E_{(w, v) \sim D} \log p(y | w, v),
\]

where \(y \in \{0, 1\}\) indicates whether the image and text is matched (\(y = 1\)) or not (\(y = 0\)). We use a single linear layer as the ITM head and compute negative log likelihood loss as our ITM loss.

We weight the pretraining objectives equally so the full pre-training objective is:

\[
L = L_{MLM} + L_{ITM} + L_{MIM}
\]
Table 1: We compare VLC to ViLT on text-image retrieval, as they have the same number of parameters. We see substantial gains across all settings. For a complete comparison, we include several state-of-the-art bounding box-based and supervised methods. ALBEF, the largest, outperforms all models, but our approach is nonetheless competitive in most settings.

4.2 Downstream Tasks

**Visual Question Answering (VQA [37])**. Given an input image and a question, the VQA task is to predict an answer from the visual content. We conduct experiments on VQAv2 dataset [37] that is built on MSCOCO. It contains 83K images for training, 41K for validation, and 81K for testing. We report performance on the test-dev and test-std splits. Following previous work [2; 1; 21], we use the training, validation splits and additional question-answer pairs from Visual Genome while reserving 1,000 validation image-question pairs for internal validation.

**Natural Language for Visual Reasoning (NLVR2 [39])**. Given a triplet of two images and a description, this task is to predict whether this description describes a pair of images. Following previous work [4; 1], we use the pair method which treats one input sample as two image-text pairs by repeating the text twice. Each pair is passed through our model and we take the concatenation of two pooled representations [CLS] from our model as the representation of one input sample.

**Image-Text Retrieval**. Image-Text retrieval contains two subtasks: image-to-text retrieval (TR) and text-to-image retrieval (IR). We evaluate our pre-trained models on the Karpathy splits [40] of MSCOCO [31] and Flickr30K [41] in fine-tuning settings. MSCOCO contains 123K images, and each image has five corresponding human-written captions. We split the data into 82K/5K/5K training/validation/test images. To be consistent with previous work [4; 1], we use the additional 30K images from MSCOCO validation set to improve the performance. Flickr30K contains 31K images with five captions for each image. We split the data into 30K/1K/1K as the training/validation/test set.

### Implementation Details

The multi-modal encoder uses a 85.8M parameter ViT-B/16 architecture initialized with MAE pre-trained on ImageNet-1K without labels. For text inputs, we tokenize text with the bert-base-uncased tokenizer. The text embedding parameters are learned from scratch, in lieu of loading pre-trained BERT weights. We randomly mask image patches with a probability of 0.6 and text tokens with a probability 0.15. To accelerate training, we follow MAE [6] and skip the mask token [MASK] in the encoder and only apply it in the lightweight decoder. We use AdamW [42] with a weight decay of 0.01. The learning rate is warmed-up to $1e^{-4}$ in the first 10% of total training steps and is decayed to zero for the rest of the training following a linear schedule. During pre-training, we resize the shorter edge of input images to 384, take random image crops of resolution 384 × 384, and apply RandAugment [43] with the hyper-parameters of $N = 2$, $M = 9$. We pre-train for 200k steps with a batch size of 4,096 on 128 NVIDIA V100 GPUs that takes 80 hours. For all downstream tasks, we fine-tune for 10 epochs with a batch size of 256 for VQAv2/retrieval tasks and 128 for NLVR2. For the parameter estimation, we exclude the textual embedder as it is shared by all vision-language transformers following ViLT [4]. We also exclude the parameters of all the auxiliary heads as they are only required during pretraining. More implementation details can be found in Appendix A.2.

### Adapt VLC to Domain-Specific Tasks and Evaluation

**Image-Text Retrieval Tasks**. We begin with a proof of concept experiment, evaluating our model on the Karpathy splits of the Flickr30K [41] and MSCOCO [31] benchmarks. In Table [1], we compare
<table>
<thead>
<tr>
<th>Model</th>
<th>Params</th>
<th>VQA v2 test-dev</th>
<th>VQA v2 test-std</th>
<th>NLVR2 dev</th>
<th>NLVR2 test</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Supervised ImageNet Bounded Boxes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ViLBERT (3)</td>
<td>274M</td>
<td>70.55</td>
<td>70.92</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LXMERT (2)</td>
<td>240M</td>
<td>72.42</td>
<td>72.54</td>
<td>74.90</td>
<td>74.50</td>
</tr>
<tr>
<td>VisualBERT (7)</td>
<td>170M</td>
<td>70.80</td>
<td>71.00</td>
<td>67.4</td>
<td>67.0</td>
</tr>
<tr>
<td>UNITER (1)</td>
<td>155M</td>
<td>72.70</td>
<td>72.91</td>
<td>77.18</td>
<td>77.85</td>
</tr>
<tr>
<td>OSCAR (11)</td>
<td>155M</td>
<td>73.16</td>
<td>73.44</td>
<td>78.07</td>
<td>78.36</td>
</tr>
<tr>
<td>VinVL (12)</td>
<td>157M</td>
<td><strong>75.95</strong></td>
<td><strong>76.12</strong></td>
<td><strong>82.05</strong></td>
<td><strong>83.08</strong></td>
</tr>
<tr>
<td><strong>Supervised ImageNet Classes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALBEF* (21)</td>
<td>187M</td>
<td><strong>74.54</strong></td>
<td><strong>74.70</strong></td>
<td><strong>80.24</strong></td>
<td><strong>80.50</strong></td>
</tr>
<tr>
<td>Visual Parsing (20)</td>
<td>180M</td>
<td>74.00</td>
<td>74.17</td>
<td>77.61</td>
<td>78.05</td>
</tr>
<tr>
<td>PixelBERT (16)</td>
<td>144M</td>
<td>74.45</td>
<td>74.55</td>
<td>76.5</td>
<td>77.2</td>
</tr>
<tr>
<td>ViLT (4)</td>
<td>87M</td>
<td>71.26</td>
<td>-</td>
<td>75.70</td>
<td>76.13</td>
</tr>
</tbody>
</table>

| No supervised classes or bounding boxes |
| VLC (ours – 4M)            | 87M    | 72.98           | 73.03           | 77.04     | 78.51      |
| VLC (ours – 5.6M)          | 87M    | **74.02**       | **74.0**        | **77.70** | **79.04**  |

Table 2: Comparison with our model with state-of-the-art pre-trained methods on vision-language understanding tasks. Our model (VLC), unlike all others, is only pre-trained with weakly-aligned image-caption pairs. Again, our approach outperforms ViLT (the closest comparison model) and is competitive with larger and more heavily supervised approaches. Rows are highlighted in shades of gray to mark use of bounding boxes and ImageNet classes. *ALBEF uses an additional 6-layer 81M parameter transformer decoder to generate answers on the VQA task which increases the model parameters to around 270M.

several strong multimodal transformers in the literature which leverage ROIs, more parameters, and are pretrained on ImageNet classification. Note that as most of detection-based models have the advantage of using Faster R-CNN (15) pre-trained on VG (32) or MSCOCO (31). ALBEF uses a pre-trained ViT-B/16 and BERT model as their backbone which doubles the model size. Additionally, they specifically design the coarse-to-fine objectives while we directly fine-tune the pre-trained ITM head for retrieval tasks. Thus we treat ALBEF as a strongest available baseline.

The closest comparison to our approach is ViLT as it is the same model size, though still requires more supervised data in the form of ImageNet classification pretraining for ViT (9). In addition, UNITER uses a frozen object detector and a trainable BERT model as their backbone which has a comparable model size. We can see substantial gains on both tasks compared with UNIERT.

**Image-Text Understanding Tasks.** Table 2 presents VLC results on two popular image-text understanding datasets: VQA v2 and NLVR2. For VQA v2, we report the test-dev and test-std scores returned from the evaluation server. For NLVR2, we evaluate our models on both dev and test-P split.

Comparison to models supervised/initialized with ImageNet bounded boxes. Most of these models use object detectors pretrained on VG (32) or MSCOCO (31) to extract region features. Object detectors help in VQA tasks as they mainly ask about objects. Within the similar scale of pretraining data, our model achieves competitive performance on both tasks. Note that our model uses 384 × 384 or 576 × 576 as input resolution during our fine-tuning stages. This resolution is much lower compared with previous work using 800 × 1333 (31). In particular, *VinVL* (12) has a multi-stage pre-training for its object detector that has access to ImageNet-5K (44) (6.8M images from 5K classes) and four object detection datasets (33; 32; 31; 38) (2.5M images with bounding box annotations). The most comparable approach is UNITER as we use the same training data (i.e., 4M images) and trainable parameters. Our approach performs better than UNITER which uses a pretrained object detector and BERT as initialization.

Comparison to models with supervised ImageNet classes. Most of these approaches use additional visual embedders together with a pretrained BERT as their backbones. For example, ALBEF (21), Visual Parsing (20), PixelBERT (16) use pre-trained ViT-B/16, Swin transformer, ResNeXt-152 as initializers.
their visual embedder, respectively. All these embedders are trained with supervised ImageNet-1K. In addition, ALBEF uses a 6-layer transformer decoder to generate answers on the VQA task which further increases the model size. With the same pretraining data, our approach outperforms ViLT by 1.72% on the VQA test-dev split, 1.34% and 2.38% on NLVR2 dev and test split. We further verify the scalability of our model by using the same scale pre-training data as VinVL. Experiments show that our model achieves comparable results with larger and more heavily supervised approaches.

4.5 Ablation Study

To understand the impact of different components, we ablate and compare variants of our model (i.e., pretraining objectives and image resolutions used during fine-tuning) and report VQAv2 test-dev accuracy in Table 3. Note that UNITER without pretraining feeds object detections to a pretrained BERT (accuracy is copied from their Table 2 Line 2). Comparing models without vision-language pretraining, theirs outperforms ours by 3.94% which indicates pretrained object detectors and BERT are strong priors. Our experiments show that both MIM and ITM are important throughout pretraining, which contrasts to findings in previous work (19; 4; 21).

Table 3: Ablation study on objectives and image resolutions (top lines show model performance without pretraining). Our experiments show that both image and text masked modeling improve the performance. Additionally, there is a consistent improvement by increasing the image resolution during fine-tuning. *The input image size of UNITER is 800 × 1333.

<table>
<thead>
<tr>
<th>Train Objective</th>
<th>Resolution</th>
<th>VQAv2 Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIM MLM ITM</td>
<td>3842 4802 5762</td>
<td>69.39 65.45 69.06 68.98 69.52 69.69 69.97</td>
</tr>
</tbody>
</table>

5 Understanding the models

While simpler and more efficient, patch-based models differ in important ways from traditional bounding-box based approaches. In particular, while the visual stack is traditionally frozen in those models, now the entire “backbone” is learnable. Also, where previously, the goal was to “map” vision to language, now the two are learned jointly. We therefore take this opportunity to investigate the models to better understand how their behaviors differ due to the two (pre-)training objectives.

Figure 3: Visualization of patch clusters for an example image as produced from ViLT (many densely clustered patches) versus VLC’s more fine-grained and diffuse representations.

Understanding Patches. We begin with a simple patch clustering visualization (Figure 3). Without the inclusion of any language, we can simply cluster (and color) the visual patch embeddings of ViLT and VLC. ViLT relies on on larger patches (32×32) for higher resolution (384×640). We instead use smaller patches and lower resolution (16×16 for 384×384). It is also easy to see how both models are identifying key semantic regions of the image (e.g. the rug, painting and plant). Also note, both models incorrectly place the painting and plant in the same cluster.

To investigate this representation collapse at scale, we leverage the nocaps dataset. Nocaps provides captions for images based on object classes in COCO, similar to COCO, and out of domain. By visualizing the embedding similarities of nouns from these three classes with patches in the images, we can determine: 1. Are ViLT patches more tightly clustered – perhaps due to the discriminative training objective and 2. How do both models’ behaviors change for classes more (or less) like the ImageNet pretraining. In Figure 4 we see several trends. First, ViLT’s “most similar” patch to
Similarity Scores Visualization on Nocaps

Noun tokens/image patches

Notes:
1. The score measures the similarities between image patch and text token. Not exactly attention weights but have a positive correlation.
2. Domain splits are based on MSCOCO dataset, may not relate to ImageNet-1k
3. ViLT scores are more concentrated than ours.

Figure 4: These plots are the top noun-patch similarity per image as produced by both ViLT and VLC. ViLT rarely produces a high similarity lexical score, likely due to its discriminative pretraining objective and its score distribution shifts down as we move further away from its supervised pretraining data. In contrast, VLC has a much smoother distribution and high lexical alignment across all settings.

the noun rarely has a passes 0.1, perhaps indicating that they are not shifting from their pretrained representations. Second, we see the mass shift slightly lower as we move from left to right (in-domain to out-of-domain), indicating the model has a harder time finding alignments to novel words. VLC has a markedly different behavior, with a smoother overall set of similarities – often able to to find a visual patch with high similarity to the query across all conditions. VLC also exhibits an opposite trend where the model’s scores climb as we shift out of domain. These plots do not show if the alignment is semantically meaningful, but they do show starkly different behaviors. This concentration of embeddings by ViLT can also be seen visually in examples in the Appendix A.3.

Image Classification. Given that the underlying visual representations are shifting through the cross-modal training, we run a simple image classification experiment to see the effects language training has on the underlying visual “backbone”. We compare VLC with state-of-the-art models on ImageNet-1K classification and report top-1 validation accuracy of a single $384 \times 384$ crop. During end-to-end fine-tuning on ImageNet-1K, we follow the supervised ViT training procedure: AdamW [42] with a base learning rate of $1e^{-3}$ and a weight decay of 0.05, batch size of 1024, and the first 5 epochs are used as warm up and decayed to $1e^{-6}$ following a cosine schedule. As our model is not pre-trained with a discriminative loss, we use a global pooling of encoder outputs as image representations.

As shown in Table 4, VLC learns generic representations which are transferable to vision tasks. With only fine-tuning on ImageNet-1K, our model matches the performance of Swin-B [48] that is trained with supervised labels. Note that BEiT [24] is a two-stage pre-training model of which the tokenizer is trained on 250M examples of DALLE [25] data. Compared with MAE [6], our model learns competitive multi-modal representations from vision-language pre-training while retains high-quality image representations.

<table>
<thead>
<tr>
<th>Model</th>
<th>Size</th>
<th>Top-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ViT-B/16</td>
<td>384^2</td>
<td>77.9</td>
</tr>
<tr>
<td>DeiT-B</td>
<td>384^2</td>
<td>83.1</td>
</tr>
<tr>
<td>Swin-B</td>
<td>384^2</td>
<td>84.5</td>
</tr>
<tr>
<td>Self-supervised</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DINO</td>
<td>224^2</td>
<td>82.8</td>
</tr>
<tr>
<td>MoCo v3</td>
<td>224^2</td>
<td>83.2</td>
</tr>
<tr>
<td>MaskFeat</td>
<td>224^2</td>
<td>83.6</td>
</tr>
<tr>
<td>SimMIM</td>
<td>224^2</td>
<td>83.8</td>
</tr>
<tr>
<td>BEiT*</td>
<td>384^2</td>
<td>84.6</td>
</tr>
<tr>
<td>MAE</td>
<td>224^2</td>
<td>83.6</td>
</tr>
<tr>
<td>VLC</td>
<td>384^2</td>
<td>84.5</td>
</tr>
</tbody>
</table>

Table 4: Models are pretrained on ImageNet 1K and self-supervised models are evaluated by end-to-end fine-tuning. *BEiT uses a DALLE [25] pre-trained tokenizer.

6 Visualizations

These patch-language transformer architectures allow for intuitive visualizations of the lexical alignment. Doing so provides a simple way to explore what the model is learning to represent about an image. In Figure 5, we show results from visualizing three different words in the same caption for an image from COCO. Not that for the word branch, the model is actively attempting to avoid the abundant leaves. Second, since there is nothing about our model besides the MAE initialization that should be biased (as shown previously) towards ImageNet classes, we present three images in Figure 6 that highlight words not present in the standard ImageNet1K training split used by other models. Specifically, a noun (string), adjective (yellow), and verb (swinging). These demonstrate the general trend of ViLT often focusing on surprising locations.
The bird is on the branch with leaves alone

Figure 5: Lexical-Patch alignment for an image in MS COCO. We visualize three different words from the same caption to see how the model uniquely represents them. This is a particularly challenging case as the model attempts to isolate patches for branches separate from those with leaves.

<table>
<thead>
<tr>
<th>Caption with focus</th>
<th>Original Image</th>
<th>VīLT</th>
<th>VLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A person on a beach holding a kite <strong>string</strong> and a kite is in the air</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>A cat sitting on a chair, that is blue and <strong>yellow</strong></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>A baseball player <strong>swinging</strong> a baseball bat at a baseball</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 6: To investigate concepts not present in COCO or ImageNet, we present three images and highlighted words which are out of domain (i.e. not in ImageNet-1K). Specifically, we are visualizing a noun (top), adjective (middle) and verb (bottom). The model again delicately avoids nearby but distinct concepts (e.g. the cat on the chair or irrelevant parts of the baseball field). More examples and analysis can be found in Appendix A.3.

7 Conclusion

We present a VLP architecture, Vision-Language from Captions (VLC), pretrained with image-caption pairs. While VLC only uses a linear projection layer as the image embedder, it achieves competitive performance on a diversified set of vision-language tasks to existing approaches that rely on object detectors or supervised CNN/ViT networks. We also evaluated the effectiveness of our vision-language pretraining on ImageNet-1K classification task to show that VLC retains high-quality image representations. Finally, our visualization demonstrates that VLC can accurately align image patches with text tokens and the performance scales with increased training data. This opens an exciting door to large scale weakly supervised open-domain vision-and-language models.
References


A Appendix

This supplementary material has three sections. Section A.1 describes the details of our pretraining datasets. Section A.2 describes our implementation details for downstream tasks. Section A.3 shows more visualization examples with more comparisons.

A.1 Pre-training dataset

The statistics of the pretraining dataset is shown in Table 5. Most of the existing approaches, such as UNITER (1) and ViLT (4), use MSCOCO, VG, GCC and SBU to pre-train their models. We denote this training set as base. To verify the scalability of our model, we follow VinVL (12) to further incorporate VQA, VG-QA, GQA, Flickr30K and OpenImages. As there are some overlaps among VG, MSCOCO and VQA, we exclude all those training images that appear in the downstream tasks via URL matching.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MSCOCO</th>
<th>VG</th>
<th>GCC</th>
<th>SBU</th>
<th>VGQA</th>
<th>GQA</th>
<th>VG-QA</th>
<th>Flickr30K</th>
<th>OpenImages</th>
</tr>
</thead>
<tbody>
<tr>
<td># Images</td>
<td>113K</td>
<td>100K</td>
<td>2.95M</td>
<td>860K</td>
<td>83K</td>
<td>79K</td>
<td>87K</td>
<td>29K</td>
<td>1.67M</td>
</tr>
<tr>
<td># Text</td>
<td>567K</td>
<td>769K</td>
<td>2.95M</td>
<td>860K</td>
<td>545K</td>
<td>1026K</td>
<td>931K</td>
<td>145K</td>
<td>1.67M</td>
</tr>
</tbody>
</table>

Table 5: Statistics of the pre-training dataset

A.2 Implementation Details for Downstream Tasks

For all downstream tasks, we fine-tune our model with a learning rate of $5 \times 10^{-4}$ for 10 epochs. We use a layer-wise learning rate decay (51) of 0.5. We use 576 × 576 as the input image resolution for the VQA task and 384 × 384 for NLVR2 and image-text retrieval tasks.

Visual Question Answering. We use a 2-layer MLP with a hidden size of 1,536 to adapt VLC to the VQA task. We follow the standard practice (4) to convert the task to a multilabel classification task with 3,192 answer classes. Following previous work (11, 21), we use additional question-answer pairs from VG for data augmentation. We select additional question-answer pairs if the corresponding images and answers appear in the VQA train and validation splits.

Natural Language for Visual Reasoning. As there are two input images and a single description, we follow OSCAR (11), ViLT (4) and VinVL (12) by using the pair method. Similar to the settings of the VQA task, we use a 2-layer MLP with a hidden size of 1,536 to adapt VLC to the NLVR2 task.

Image-Text Retrieval. We conduct experiments on both MSCOCO and Flickr30K datasets. Given an image, we use the corresponding text as a positive example while randomly sample 15 text as negative examples. We use a fully connected layer as our retrieval similarity head that is initialized from the pre-trained ITM head. We fine-tune our model with a cross-entropy loss to maximize the probabilities on positive pairs.

A.3 Analysis on More Examples

We show additional examples for nouns in Figure 7, adjectives in Figure 8 and verbs in Figure 9.
<table>
<thead>
<tr>
<th>Caption with focus</th>
<th>Original Image</th>
<th>ViLT</th>
<th>VLC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A hawk is perched on a metal bar</strong></td>
<td><img src="image1" alt="Image of a hawk on a metal bar" /></td>
<td><img src="image2" alt="ViLT representation" /></td>
<td><img src="image3" alt="VLC representation" /></td>
</tr>
<tr>
<td><strong>A gift wrapped with a ribbon sits on a table with a knife</strong></td>
<td><img src="image4" alt="Image of a gift with a ribbon and knife" /></td>
<td><img src="image5" alt="ViLT representation" /></td>
<td><img src="image6" alt="VLC representation" /></td>
</tr>
<tr>
<td><strong>A plate with pancakes, syrup, grits, and butter</strong></td>
<td><img src="image7" alt="Image of a plate with breakfast" /></td>
<td><img src="image8" alt="ViLT representation" /></td>
<td><img src="image9" alt="VLC representation" /></td>
</tr>
<tr>
<td><strong>There is a colorful parachute in the sky</strong></td>
<td><img src="image10" alt="Image of a parachute in the sky" /></td>
<td><img src="image11" alt="ViLT representation" /></td>
<td><img src="image12" alt="VLC representation" /></td>
</tr>
</tbody>
</table>

Figure 7: Visualized are OOD noun examples. Note that ViLT is often picking up on relevant features but has a single strongest correlation with a single, presumably predictive, patch.
<table>
<thead>
<tr>
<th>Caption with focus</th>
<th>Original Image</th>
<th>ViLT</th>
<th>VLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A <strong>red</strong> fire hydrant in front of a skyscraper</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>A monarch butterfly lands on a <strong>pink</strong> flower.</td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>A <strong>small</strong> orange and blue ladybug sitting on long green leaves</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
</tr>
<tr>
<td>A brown and white dog is holding a <strong>yellow</strong> Frisbee</td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 8: Visualized are OOD adjective examples. **VLC** produces more accurate and comprehensive masks. Note that the lady bug is correctly identified but not exclusively and likely not based on an understanding of the relative size **small**. Future work would ideally show results that indicate models understanding more abstract and comparative concepts.
<table>
<thead>
<tr>
<th>Caption with focus</th>
<th>Original Image</th>
<th>ViLT</th>
<th>VLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A person who is hitting a ball with a bat.</td>
<td><img src="image1.jpg" alt="Image" /></td>
<td><img src="image2.jpg" alt="Image" /></td>
<td><img src="image3.jpg" alt="Image" /></td>
</tr>
<tr>
<td>A person holding a cell phone in their hand</td>
<td><img src="image4.jpg" alt="Image" /></td>
<td><img src="image5.jpg" alt="Image" /></td>
<td><img src="image6.jpg" alt="Image" /></td>
</tr>
<tr>
<td>A green boat floating on top of a body of water</td>
<td><img src="image7.jpg" alt="Image" /></td>
<td><img src="image8.jpg" alt="Image" /></td>
<td><img src="image9.jpg" alt="Image" /></td>
</tr>
<tr>
<td>an orange and white cat sitting on a bed staring at the viewer</td>
<td><img src="image10.jpg" alt="Image" /></td>
<td><img src="image11.jpg" alt="Image" /></td>
<td><img src="image12.jpg" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 9: Visualized are OOD verb examples. Note that verbs from still images is a slightly strange concept, but there are key perceptual indicators that align to the verb’s semantics. For example, holding is aligned to the person’s hands and staring picks up on the cat’s eyes.