An Empirical Study of Training End-to-End Vision-and-Language Transformers

Zi-Yi Dou\textsuperscript{1,}, Yichong Xu\textsuperscript{2}, Zhe Gan\textsuperscript{2}, Jianfeng Wang\textsuperscript{2}, Shuohang Wang\textsuperscript{2}, Lijuan Wang\textsuperscript{2}, Chenguang Zhu\textsuperscript{2}, Pengchuan Zhang\textsuperscript{2}, Lu Yuan\textsuperscript{2}, Nanyun Peng\textsuperscript{1}, Zicheng Liu\textsuperscript{2}, Michael Zeng\textsuperscript{2}
\textsuperscript{1}University of California, Los Angeles, \textsuperscript{2}Microsoft Corporation
{zdou,violetpeng}@cs.ucla.edu
{yicxu,zhgan,jianfw,shuowa,lijuanw,chezhu,penzhan,luyuan,zliu,nzeng}@microsoft.com

Abstract

Vision-and-language (VL) pre-training has proven to be highly effective on various VL downstream tasks. While recent work has shown that fully transformer-based VL models can be more efficient than previous region-feature-based methods, their performance on downstream tasks often degrades significantly. In this paper, we present METER, a Multimodal End-to-end Transform\textsuperscript{E}R framework, through which we investigate how to design and pre-train a fully transformer-based VL model in an end-to-end manner. Specifically, we dissect the model designs along multiple dimensions: vision encoders (e.g., CLIP-ViT, Swin transformer), text encoders (e.g., RoBERTa, DeBERTa), multimodal fusion module (e.g., merged attention vs. co-attention), architectural design (e.g., encoder-only vs. encoder-decoder), and pre-training objectives (e.g., masked image modeling). We conduct comprehensive experiments and provide insights on how to train a performant VL transformer. \textsuperscript{E}R achieves an accuracy of 77.64\% on the VQAv2 test-std set using only 4M images for pre-training, surpassing the state-of-the-art region-feature-based model by 1.04\%, and outperforming the previous best fully transformer-based model by 1.6\%. Notably, when further scaled up, our best VQA model achieves an accuracy of 80.54\%. Code and pre-trained models are released at \url{https://github.com/zdou0830/METER}.

1. Introduction

Vision-and-language (VL) tasks, such as visual question answering (VQA) \cite{chao2018} and image-text retrieval \cite{li2019,xiong2019}, require an AI system to comprehend both the input image and text contents. Vision-and-language pre-training (VLP) has now become the de facto practice to tackle these tasks \cite{yu2018,xiong2019,liu2019,li2020,he2020,carion2020}. Specifically, large amounts of image-caption pairs are fed into a model that consumes both images and texts to pretrain representations that contain rich multimodal knowledge and is helpful for downstream tasks.

Transformers \cite{vaswani2017} are prevalent in natural language processing and have recently shown promising performance in computer vision \cite{dosovitskiy2021}. While almost all the existing VLP models adopt transformers for their multimodal fusion module, most of them \cite{yu2018,xiong2019,liu2019,li2020,he2020,carion2020} use pre-trained object detectors (e.g., Faster RCNN \cite{ren2015}) on the vision side to extract region features from images. This can lead to several problems: first, the object detectors are not perfect, but are usually kept frozen during VLP, which limits the capacity of the VLP models; second, it is time-consuming to extract region features \cite{dosovitskiy2021}. On the other hand, vision transformers (ViTs) have been an increasingly active research topic in computer vision and have shown great potential in vision feature extraction. Therefore, a natural question arises: \textit{can we train a fully transformer-based VLP model with ViTs as the image encoder?}

Recent works \cite{dosovitskiy2021,carion2020,dosovitskiy2020} that tried to adopt vision transformers have not shown satisfactory performance and typically underperform state-of-the-art region-feature-based VLP models (e.g., VinVL \cite{yu2020}). To close the performance gap, we present METER, a Multimodal End-to-end Transform\textsuperscript{E}R framework, through which we thoroughly investigate how to design and pre-train a fully transformer-
based VLP model in an end-to-end manner. Specifically, as shown in Figure 1, we dissect the model designs along multiple dimensions, including vision encoders (e.g., CLIP-ViT [35], Swin transformer [30]), text encoders (e.g., RoBERTa [29], DeBERTa [13]), multimodal fusion module (e.g., merged attention vs. co-attention), architectural design (e.g., encoder-only vs. encoder-decoder), and pre-training objectives (e.g., masked image modeling [2]).

We perform the investigation by pre-training models under METER on four commonly used image-caption datasets: COCO [27], Conceptual Captions [40], SBU Captions [33], and Visual Genome [21]. We test them on visual question answering [1], visual reasoning [43], image-text retrieval [27, 34], and visual entailment [52] tasks. Through extensive analyses, we summarize our findings as follows:

- Vision transformer (ViT) plays a more vital role than language transformer in VLP, and the performance of transformers on pure vision or language tasks is not a good indicator for its performance on VL tasks.
- The inclusion of cross-attention benefits multimodal fusion, which results in better downstream performance than using self-attention alone.
- Under a fair comparison setup, the encoder-only VLP model performs better than the encoder-decoder model for VQA and zero-shot image-text retrieval tasks.
- Adding the masked image modeling loss in VLP will not improve downstream task performance in our settings.

These insights, combined with other useful tips and tricks detailed in later sections, enable us to train a strong model that achieves an accuracy of 77.64% on the VQAv2 test-std set, surpassing the previous best region-feature-based VinVL model [57] by 1.04% and outperforming the previously best ViT-based model (i.e., ALBEF [23]) by 1.6%.

Notably, when further scaled up, our best METER model achieves an accuracy of 80.54% on the VQAv2 test-std set.

2. Glossary of VLP Models

In this section, we provide an overview of representative VLP models, and divide them into three categories based on how they encode images, as summarized in Table 1.

**OD-based Region Features.** Most previous work use pre-trained object detectors (ODs) to extract visual features. Among them, ViLBERT [32] and LXMERT [44] use co-attention for multimodal fusion, where two transformers are applied independently to region and text features, and another transformer fuses the representations of the two modalities in a later stage. On the other hand, VisualBERT [24], VL-BERT [42], and UNITER [5] use a merged attention fusion module that feeds both region and text features together into a single transformer. OSCAR [26] and VinVL [57] feed additional image tags into the transformer model, and demonstrate state-of-the-art performance across VL tasks. However, extracting region features can be time-consuming, and the pre-trained ODs are usually frozen during pre-training, which limits the capacity of VLP models.

**CNN-based Grid Features.** To tackle the above two issues, researchers have tried different ways to pre-train VL models in an end-to-end fashion. Among them, PixelBERT [16] and CLIP-ViL [41] propose to feed grid features from convolutional neural networks (CNNs) and text directly into a transformer. SOHO [15] proposes to to first discretize grid features using a learned vision dictionary, then feed the discretized features into their cross-modal module. While using grid features directly can be efficient, inconsistent optimizers are typically used for CNN and transformer. For example, PixelBERT [16] and CLIP-ViL [41]
use AdamW [31] for transformer and SGD for CNN. Recent work on vision transformers (ViTs) has also shown that CNN can achieve slightly worse accuracy/FLOPs trade-offs than their ViT counterparts [30], motivating researchers to develop ViT-based VLP models.

**ViT-based Patch Features.** ViLT [20] directly feeds image patch features and text token embeddings into a pre-trained ViT model, and fine-tunes the model on image-caption datasets. More recently, visual parsing [53] and ALBEF [23] use ViT as the image encoder and design different pre-training objectives for ViT-based VLP models. However, all these models lag behind the state-of-the-art performance on downstream tasks such as visual question answering. In this paper, we investigate how to pre-train a ViT-based model in an end-to-end manner that closes the performance gap while maintaining fast inference speed.

### 3. The METER Framework

Based on the previous work, we identify several important modules in VLP models as in Figure 1. In this section, we first illustrate our METER framework, then our default settings, which paves the way for our analyses hereinafter.

**Overview.** Given a text sentence $l$ and an image $v$, a VLP model first extracts both text features $l = \langle l_1, \cdots, l_N \rangle$ and visual features $v = \langle v_1, \cdots, v_M \rangle$ via a text encoder and a vision encoder. The text and visual features are then fed into a multimodal fusion module to produce cross-modal representations, which are then optionally fed into a decoder before generating the final outputs.

#### 3.1. Model Architecture

**Vision Encoder.** In this paper, we focus on patch features, and study the use of vision transformers (ViTs) [10] for vision encoder. Specifically, in ViT, an image is first segmented into patches, and then the patches are fed into a transformer model. ViT has become a popular research topic recently [2, 10, 30, 45, 45, 46, 56], and has been introduced into VLP [20, 23, 53]. However, all these models only achieve inferior performance compared to state-of-the-art region-feature-based models (e.g., ViT-L [57]). Also, different pre-trained ViTs are used, lacking a systematic study of which ViTs are the best for VLP. In this work, we compare the original ViT [10], DeiT [45], Distilled-DeiT [45], CaiT [46], VOLO [56], BEiT [2], Swin Transformer [30] and CLIP-ViT [35], to provide a comprehensive analysis on the role of vision transformers.

**Text Encoder.** Following BERT [9] and RoBERTa [29], VLP models [5, 24, 26, 32, 42, 44] first segment the input sentence into a sequence of subwords [39], then insert two special tokens at the beginning and the end of the sentence to generate the input text sequence. After we obtain the text embeddings, existing works either feed them directly to the multimodal fusion module [5, 24], or to several text-specific layers [32, 44] before the fusion. For the former, the fusion module is typically initialized with BERT, and the role of text encoding and multimodal fusion is therefore entangled and absorbed in a single BERT model. Here, we aim to disentangle the two modules, and use a text encoder first before sending the features into the fusion module.

Language model (LM) pre-training has demonstrated impressive performance across tasks and different pre-trained LMs have been proposed; however, most VLP models still only use BERT for initialization [5]. In this work, we study the use of BERT [9], RoBERTa [29], ELECTRA [7], ALBERT [22], and DeBERTa [13] for text encoding. Besides, we also experiment on only using a simple word embedding look-up layer initialized with the BERT embedding layer as used in many previous works [5, 57].

**Multimodal Fusion.** We study two types of fusion modules, namely, merged attention and co-attention [14], as illustrated in Figure 2. In the merged attention module, the text and visual features are simply concatenated together, then fed into a single transformer block. In the co-attention module, on the other hand, the text and visual features are fed into different transformer blocks independently, and techniques such as cross-attention are used to enable cross-modal interaction. For region-based VLP models, as shown in [3], the merged attention and co-attention models can achieve comparable performance. Yet, the merged attention module is more parameter-efficient, as the same set of parameters are used for both modalities. Since end-to-end VLP models are becoming increasingly popular, in this work, we re-examine the impact of both types of fusion modules in our new context.

**Encoder-Only vs. Encoder-Decoder.** Many VLP models such as VisualBERT [24] adopt the encoder-only architecture, where the cross-modal representations are directly fed into an output layer to generate the final outputs. Recently, VL-T5 [6] and SimVLM [51], on the other hand, advocate the use of a transformer encoder-decoder architecture, where the cross-modal representations are first fed into a decoder and then to an output layer. In their models,
Figure 3. Illustration of the encoder-only and encoder-decoder model architectures for VLP.

the decoder attends to both the encoder representations and the previously generated tokens, producing the outputs autoregressively. Figure 3 shows the difference between them when performing the masked language modeling task. For the encoder-decoder model, when performing classification tasks such as VQA, we feed the text inputs into its encoder and feed a classification token into the decoder, and the decoder then generates the output class accordingly.

3.2. Pre-training Objectives

Now, we introduce how we pre-train our models. Specifically, we will first briefly review masked language modeling and image-text matching, which have been used extensively in almost every VLP model. Then, we will shift our focus to how we can design and explore interesting masked image modeling tasks.

Masked Language Modeling. The masked language modeling (MLM) objective is first introduced in pure language pre-training [9, 29]. In VLP, MLM with images has also proven to be useful. Specifically, given an image-caption pair, we randomly mask some of the input tokens, and the model is trained to reconstruct the original tokens given the masked tokens $^{\text{mask}}$ and its corresponding visual input $v$.

Image-Text Matching. In image-text matching, the model is given a batch of matched or mismatched image-caption pairs, and the model needs to identify which images and captions correspond to each other. Most VLP models treat image-text matching as a binary classification problem. Specifically, a special token (e.g., [CLS]) is inserted at the beginning of the input sentence, and it tries to learn a global cross-modal representation. We then feed the model with either a matched or mismatched image-caption pair $(v, l)$ with equal probability, and a classifier is added on top of the [CLS] token to predict a binary label $y$, indicating if the sampled image-caption pair is a match.

Masked Image Modeling. Similar to the MLM objective, researchers have tried masked image modeling (MIM) on the vision side. For example, many previous work, such as LXMERT [44] and UNITER [5], mask some of the input regions, and the model is trained to regress the original region features. Formally, given a sequence of visual features $v = (v_1, \ldots, v_M)$, where $v_i$ is typically a region feature, we randomly mask some of the visual features, and the model outputs the reconstructed visual features $o_v$ given the rest of the visual features and the unmasked tokens $t$, and regression aims to minimize the mean squared error loss. Researchers [5, 32, 44] have also tried to first generate object label for each region using a pre-trained object detector, which can contain high-level semantic information, and the model is trained to predict the object labels for the masked regions instead of the original region features.

Notably, recent state-of-the-art models (e.g., ALBEF [23], VinVL [57]) do not apply MIM during VLP. In addition, in ViLT [20], the authors also demonstrate that masked patch regression is not helpful in their setting. These results make it questionable whether MIM is truly effective for VLP models. To further investigate this, we treat masked image modeling as a masked patch classification task, and propose two ways of implementing the idea.

1) Masked Patch Classification with In-batch Negatives. By imitating MLM which uses a text vocabulary, we first propose to let the model reconstruct input patches by using a dynamically constructed vocabulary constructed with in-batch negatives. Concretely, at each training step, we sample a batch of image-caption pairs $\{(v^k, l^k)\}_{k=1}^B$, where $B$ is the batch size. We treat all the patches in $\{v^k\}_{k=1}^B$ as candidate patches, and for each masked patch, we mask 15% of the input patches, and the model needs to select the original patch within this candidate set. Denoting the original patch representations and our model’s output representations of $\{v^k\}_{k=1}^B$ as $\{c(v^k)\}_{k=1}^B$ and $\{h(v^k)\}_{k=1}^B$, respectively, we can represent the output probability at position $i$ for the $k$-th instance as:

$$p(v^k_i | v^{k,\text{mask}}_i; l^k_i) = \frac{e^{h(v^k_i)^T c(v^k_i)}}{\sum_{j,k'} e^{h(v^k_{i'})^T c(v^k_{i'})}}. \quad (1)$$

The model is trained to maximize its probability similar to noise contrastive estimation [12, 18].

2) Masked Patch Classification with Discrete Code. Inspired by BEiT [2], we also propose to obtain discrete repre-
sentations of the input patches, and the model is then trained to reconstruct the discrete tokens. Specifically, we first use the VQ-VAE [47] model in DALL-E [37] to tokenize each image into a sequence of discrete tokens. We resize each image so that the number of patches is equal to the number of tokens, and thus each patch corresponds to a discrete token. Then, we randomly mask 15% of the patches and feed the masked image patches to the model as before, but now the model is trained to predict the discrete tokens instead of the masked patches.

3.3. Our Default Settings for METER

There are many different model designs under METER, and we detail our default settings in this part.

Model Architecture. The default setting of model architecture is shown in Figure 2a. In the bottom part, there are one pre-trained visual encoder and one pre-trained text encoder. On top of each encoder, we stack $M = 6$ transformer encoding layers, with each layer consisting of one self-attention block, one cross-attention block, and one feed-forward network block. Unless otherwise stated, the hidden size is set to 768, and the number of heads is set to 12 for the top layers. Note that there is no decoder and no parameter sharing between the vision and language branches.

Pre-training Objectives. Unless otherwise stated, we pre-train the models with masked language modeling (MLM) and image-text matching (ITM) only. For MLM, we mask 15% of the input text tokens, and the model tries to reconstruct the original tokens. For ITM, we feed the model with either matched or mismatched image-caption pairs with equal probability, and the model needs to identify whether the input is a match.

Pre-training Datasets. Following previous work [5, 20, 23], we pre-train models on four commonly used datasets, including COCO [27], Conceptual Captions [40], SBU Captions [33], and Visual Genome [21]. The statistics of these datasets is shown in Appendix. The combined training data consists of about 4M images in total.

Downstream Tasks. For ablation and analysis, we mainly focus on VQA v2 [1], arguably the most popular dataset for VLP evaluation. We also test on Flickr30k zero-shot image-text retrieval to remove any confounders that may be introduced during finetuning [14]. For VQA v2, we follow the standard practice [5] to train the models with both training and validation data, and test the models on the test-dev set. For Flickr30k, we follow the standard splits.

For comparison with state of the arts, the models are trained on the following datasets: Captions [33], and Visual Genome [21]. The statistics of these datasets is shown in Appendix. The combined training data consists of about 4M images in total.

Table 2. Comparisons of different text encoders without VLP.

<table>
<thead>
<tr>
<th>Text Enc.</th>
<th>VQA v2</th>
<th>VE</th>
<th>IR</th>
<th>TR</th>
<th>SQuAD</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emb-only</td>
<td>67.13</td>
<td>74.85</td>
<td>49.06</td>
<td>68.20</td>
<td>86.8</td>
<td>88.8</td>
</tr>
<tr>
<td>ELECTRA</td>
<td>69.22</td>
<td>76.57</td>
<td>41.80</td>
<td>58.30</td>
<td>86.4</td>
<td>87.6</td>
</tr>
<tr>
<td>CLIP</td>
<td>69.31</td>
<td>75.37</td>
<td>54.96</td>
<td>73.80</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DeBERTa</td>
<td>67.74</td>
<td>51.50</td>
<td>67.70</td>
<td>-</td>
<td>87.2</td>
<td>88.8</td>
</tr>
<tr>
<td>BERT</td>
<td>69.56</td>
<td>72.67</td>
<td>49.60</td>
<td>66.60</td>
<td>76.3</td>
<td>84.3</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>69.69</td>
<td>75.63</td>
<td>49.86</td>
<td>68.90</td>
<td>84.6</td>
<td>87.6</td>
</tr>
<tr>
<td>ALBERT</td>
<td>69.94</td>
<td>76.20</td>
<td>52.20</td>
<td>68.70</td>
<td>86.4</td>
<td>87.9</td>
</tr>
</tbody>
</table>

Table 3. Comparisons of different vision encoders without VLP.

<table>
<thead>
<tr>
<th>Vision Encoder</th>
<th>VQA v2</th>
<th>VE</th>
<th>IR</th>
<th>TR</th>
<th>SQuAD</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT B-384/16</td>
<td>67.84</td>
<td>76.17</td>
<td>34.84</td>
<td>52.10</td>
<td>85.2</td>
<td></td>
</tr>
<tr>
<td>BEiT B-224/16</td>
<td>68.37</td>
<td>75.80</td>
<td>32.24</td>
<td>59.80</td>
<td>85.2</td>
<td></td>
</tr>
<tr>
<td>DeiT B-384/16</td>
<td>69.82</td>
<td>75.97</td>
<td>33.38</td>
<td>50.90</td>
<td>82.9</td>
<td></td>
</tr>
<tr>
<td>ViT B-384/16</td>
<td>69.09</td>
<td>76.35</td>
<td>40.30</td>
<td>59.80</td>
<td>83.97</td>
<td></td>
</tr>
<tr>
<td>CLIP B-224/32</td>
<td>69.69</td>
<td>76.53</td>
<td>49.86</td>
<td>68.90</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>VOLO 4-448/32</td>
<td>71.44</td>
<td>76.42</td>
<td>40.90</td>
<td>61.40</td>
<td>86.8</td>
<td></td>
</tr>
<tr>
<td>CaT-M-384/32</td>
<td>71.52</td>
<td>76.62</td>
<td>38.96</td>
<td>61.30</td>
<td>86.1</td>
<td></td>
</tr>
<tr>
<td>CLIP B-224/16</td>
<td>71.25</td>
<td>77.54</td>
<td>57.64</td>
<td>76.90</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Swin B-384/32</td>
<td>72.38</td>
<td>77.65</td>
<td>57.64</td>
<td>76.90</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

4. Experiments

In this section, we provide comprehensive analysis of each individual module design. Specifically, (i) we study the impact of vision and language encoders in Section 4.1, (ii) we perform analysis on multimodal fusion designs in Section 4.2, (iii) we compare encoder-only and encoder-decoder architectures in Section 4.3, and (iv) we ablate pre-training objectives in Section 4.4. Finally, we compare with state of the arts in Section 4.5.

4.1. Impact of Vision and Language Encoders

4.1.1 Explorations without VLP

Since pre-training is time-consuming, we first perform an exploration study by comparing different text and visual en-
Impact of Text Encoders. As shown in Table 2, there are significant differences between the model performance of different text encoders. RoBERTa seems to achieve the most robust performance in this setting. Also, as can be seen from the Emb-only results, it is necessary to have a pre-trained encoder because otherwise the downstream task performance will be degraded.

Impact of Vision Encoders. As summarized in Table 3, both CLIP-ViT-224/16 and Swin Transformer can achieve decent performance in this setting. Notably, Swin Transformer can achieve an VQA score of 72.38 on the test-dev set without any VLP, which is already comparable to some VLP models after pre-training.

Conclusion. If we directly finetune the models on downstream tasks without any VLP, RoBERTa and Swin Transformer or CLIP-ViT perform the best. While models such as DeBERTa and BEiT can achieve better performance than the two models on pure language or vision tasks such as MNLI [49] or ImageNet classification [8], that does not necessarily indicate that they are more suitable for VL tasks.

4.1.2 Results with VLP

Now, we follow the default setting in Section 3.3, and compare different vision/text encoders with VLP. Based on the previous results, we compare Embed-only, BERT, and RoBERTa on the text side, and CLIP-ViT-224/32, CLIP-ViT-224/16, and Swin Transformer on the image side.

Results. As shown in Table 4, after VLP, the difference between BERT and RoBERTa seems to be diminished, but it is still important to have a pre-trained text encoder on the bottom (Embed-only vs. RoBERTa). For vision encoder, both CLIP-ViT-224/16 and Swin Transformer can achieve pretty good performance. Especially, CLIP-ViT-224/16 can achieve a VQA score of 77.19/77.20 on the test-dev/test-std sets, respectively, outperforming the previous state-of-the-art region-based VinVL [57] models.

Useful Tricks. In experiments, we found two tricks for ViT-based VLP models that can greatly boost the performance. First, it is better to use a larger learning rate for the randomly initialized parameters than parameters initialized with pre-trained models, which is also found useful in some other NLP tasks [28]. As shown in Table 5, using the same learning rate for all parts of the model will lead to degraded performance, possibly because the pre-trained parameters already contain certain amounts of knowledge about vision and language, and finetuning them aggressively can result in the loss of these valuable information.

Second, similar to several previous work [20,56], we find that increasing the image resolution during finetuning can improve the model performance by a large margin, especially when the ratio of image resolution to patch size is low. Figure 5 shows that increasing the image resolution from 224 to 576 can improve the VQA score by about 3 points for the CLIP-ViT-224/32 and CLIP-ViT-224/16 model, respectively.

4.2. Analysis of the Multimodal Fusion Module

Now, following the default setting in Section 3.3, we perform investigations on multimodal fusion. First, we design two tricks for art region-based VinVL models.

<table>
<thead>
<tr>
<th>Text Enc.</th>
<th>Vision Enc.</th>
<th>VQA2</th>
<th>Flickr-ZS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emb-only</td>
<td>CLIP-32</td>
<td>73.99</td>
<td>60.32</td>
</tr>
<tr>
<td>BERT</td>
<td>CLIP-32</td>
<td>76.70</td>
<td>74.52</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>CLIP-32</td>
<td>74.67</td>
<td>65.50</td>
</tr>
<tr>
<td>Swin</td>
<td>CLIP-32</td>
<td>77.19</td>
<td>76.64</td>
</tr>
</tbody>
</table>

Table 4. Comparisons of different vision and text encoders with VLP. Results on VQA2 are on test-dev set. ZS: zero-shot.

<table>
<thead>
<tr>
<th>Bottom LR</th>
<th>Top LR</th>
<th>VQA2</th>
<th>Flickr-ZS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e-5</td>
<td>1e-5</td>
<td>73.16</td>
<td>48.80</td>
</tr>
<tr>
<td>2e-5</td>
<td>2e-5</td>
<td>73.66</td>
<td>53.14</td>
</tr>
<tr>
<td>3e-5</td>
<td>3e-5</td>
<td>73.77</td>
<td>56.48</td>
</tr>
<tr>
<td>5e-5</td>
<td>5e-5</td>
<td>73.54</td>
<td>52.48</td>
</tr>
</tbody>
</table>

Table 5. Using different learning rates for the randomly-initialized and pre-trained parameters is better than using the same learning rate. Results on VQA2 are on test-dev set. ZS: zero-shot.

Figure 5. Increasing the image resolution during finetuning greatly improves the performance on the VQA2 test-dev set.
In this section, we evaluate our best-performing models (i.e., RoBERTa-base+Swin Transformer and RoBERTa-base+CLIP-ViT-224/16 with co-attention fusion module, and with image resolutions set to 384 and 288, respectively), and compare them with previous work. We evaluate the models on visual question answering (VQA2v), visual reasoning (NLVR2), visual entailment (SNLI-VE), Flickr30k retrieval tasks in zero-shot and finetuning settings, and COCO retrieval tasks in the finetuning setting.

**Main Results.** As in Table 8 and 9, compared with models pre-trained with fewer than 10M images, our CLIP-based model (METER-CLIP-ViTBASE) can achieve either the best or the second best scores on all the downstream tasks. Notably, our model can achieve a VQA score of 77.64% on the VQA2 test-std set using only 4M images for pre-training.

### 4.4. Ablations on Pre-training Objectives

In all the previous experiments, we pre-train our models with different objectives, following the default setting in Section 3.3. Now, we alter the pre-training objectives.

**Results.** As summarized in Table 7, both masked language modeling and image-text matching can bring performance improvements on downstream tasks. However, both of our masked image modeling objectives can lead to degraded performance on both VQA2v and Flickr30k retrieval tasks. This further indicates that conclusions in region-based VLP models may not necessarily hold in vision transformer-based models. We hypothesize that the performance drop is due to the conflicts between different objectives, and some techniques in multi-task optimization [50, 54] may be borrowed to resolve the conflicts, which we list as one of the future directions. Another possible reason is that image patches can be noisy, thus the supervisions on reconstructing these noisy patches can be uninformative.

**4.5. Comparison with Prior Arts**

In this section, we evaluate our best-performing models (i.e., RoBERTa-base+Swin Transformer and RoBERTa-base+CLIP-ViT-224/16 with co-attention fusion module, and with image resolutions set to 384 and 288, respectively), and compare them with previous work. We evaluate the models on visual question answering (VQA2v), visual reasoning (NLVR2), visual entailment (SNLI-VE), Flickr30k retrieval tasks in zero-shot and finetuning settings, and COCO retrieval tasks in the finetuning setting.

**Main Results.** As in Table 8 and 9, compared with models pre-trained with fewer than 10M images, our CLIP-based model (METER-CLIP-ViTBASE) can achieve either the best or the second best scores on all the downstream tasks. Notably, our model can achieve a VQA score of 77.64% on the VQA2 test-std set using only 4M images for pre-training.

### 4.3. Encoder-Only vs. Encoder-Decoder

We then compare the encoder-only and encoder-decoder architecture. For the encoder-only model, we use the same co-attention model as in Section 4.2. For the encoder-decoder model, we set the number of layers to 3 for both the encoder and decoder, and each decoding layer has two separate cross-attention blocks that attend to the vision and text representations, respectively. According to [6], we adopt T5-style [36] language modeling objective as it works well for their model. Specifically, we mask 15% of input text tokens and replace contiguous text span with sentinel tokens, and the decoder is trained to reconstruct the masked tokens. For image-text matching, we feed the decoder with a special class token and it generates a binary output.

**Results.** As shown in Table 6, the encoder-only model can outperform the encoder-decoder model on our two discriminative tasks, which is consistent with the findings in [6].

### Table 6. Co-attention performs better than merged attention in our setting, and adding a decoder is not helpful for our discriminative VL tasks. Results on VQA2v are on test-dev set. ZS: zero-shot.

<table>
<thead>
<tr>
<th>Pre-training Objectives</th>
<th>VQA2v</th>
<th>Flickr-ZS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLM</td>
<td>74.19</td>
<td>-</td>
</tr>
<tr>
<td>ITM</td>
<td>72.63</td>
<td>53.74</td>
</tr>
<tr>
<td>MLM+ITM</td>
<td>74.98</td>
<td>66.08</td>
</tr>
<tr>
<td>MLM+ITM + MIM (In-batch Negatives)</td>
<td>74.01</td>
<td>62.12</td>
</tr>
<tr>
<td>MLM+ITM + MIM (Discrete Code)</td>
<td>74.21</td>
<td>59.80</td>
</tr>
</tbody>
</table>

**Table 7. Masked language modeling (MLM) and image-text matching (ITM) can both improve the model performance, but both of our designed masked image modeling (MIM) objectives lead to degraded performance on downstream tasks. Results on VQA2v are on test-dev set. ZS: zero-shot.**

However, it should be noted that the encoder-decoder architecture is more flexible, as it can perform tasks such as image captioning which may not be that straightforward for an encoder-only model to be applied to.

### 4.2. Methods

In all the previous experiments, we pre-train our models with different objectives, following the default setting in Section 3.3. Now, we alter the pre-training objectives.

**Results.** As summarized in Table 7, both masked language modeling and image-text matching can bring performance improvements on downstream tasks. However, both of our masked image modeling objectives can lead to degraded performance on both VQA2v and Flickr30k retrieval tasks. This further indicates that conclusions in region-based VLP models may not necessarily hold in vision transformer-based models. We hypothesize that the performance drop is due to the conflicts between different objectives, and some techniques in multi-task optimization [50, 54] may be borrowed to resolve the conflicts, which we list as one of the future directions. Another possible reason is that image patches can be noisy, thus the supervisions on reconstructing these noisy patches can be uninformative.
Comparisons with models pre-trained with $<10$M images on visual question answering, visual reasoning, visual entailment, and zero-shot image retrieval (IR) and text retrieval (TR) tasks. The best scores are in **bold**, and the second best scores are in *underlined*.

<table>
<thead>
<tr>
<th>Model</th>
<th>VQA2</th>
<th>NLVR2</th>
<th>SNLI-VE</th>
<th>Flickr-ZS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>test-dev</td>
<td>test-std</td>
<td>dev</td>
<td>test</td>
</tr>
<tr>
<td>ALBEF (4M) [23]</td>
<td>75.84</td>
<td>76.04</td>
<td>82.55</td>
<td>83.14</td>
</tr>
<tr>
<td>SimVL{BASE, HUGE} (1.8B) [51]</td>
<td>77.87</td>
<td>78.14</td>
<td>81.72</td>
<td>81.77</td>
</tr>
</tbody>
</table>

Comparisons with models pre-trained with $>10$M images on VQA2, surpassing previous models trained with 1.8B images.

<table>
<thead>
<tr>
<th>Model</th>
<th>VQA2</th>
<th>NLVR2</th>
<th>SNLI-VE</th>
<th>Flickr-ZS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>test-dev</td>
<td>test-std</td>
<td>dev</td>
<td>test</td>
</tr>
<tr>
<td>ALBEF (4M) [23]</td>
<td>82.83</td>
<td>83.11</td>
<td>89.16</td>
<td>89.41</td>
</tr>
<tr>
<td>SimVL{BASE, HUGE} (1.8B) [51]</td>
<td>80.03</td>
<td>80.34</td>
<td>84.53</td>
<td>85.15</td>
</tr>
</tbody>
</table>

## Table 8.

### Table 9.

Comparisons with models pre-trained with $<10$M images on Flickr30k and COCO image retrieval (IR) and text retrieval (TR) tasks in the finetuning setting. The best scores are in **bold**, and the second best scores are in *underlined*.

<table>
<thead>
<tr>
<th>Model</th>
<th>Flickr</th>
<th>COCO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IR@1</td>
<td>IR@5</td>
</tr>
<tr>
<td>ALBEF (4M) [23]</td>
<td>80.33</td>
<td>80.54</td>
</tr>
<tr>
<td>SimVL{BASE, HUGE} (1.8B) [51]</td>
<td>81.08</td>
<td>81.34</td>
</tr>
</tbody>
</table>

### Table 10.

Pre-training a huge model under the METER framework with 14M images can lead to state-of-the-art performance on VQA2, surpassing previous models trained with 1.8B images.

**Scaling the Model.** We also investigate if the METER framework is scalable. To this end, we pre-train our model with more images and larger vision backbone. Specifically, we pre-train the model with COCO, CC, CC12M [4], SBU, and VG datasets, consisting of about 14M images and 20M image-caption pairs in total. We use CoSwin-Huge [55] as our vision backbone and RoBERTa-base as our text backbone. The hidden size of the fusion module remains unchanged. As shown in Table 10, our model can achieve state-of-the-art performance on VQA2, surpassing previous models trained with 1.8B images. The results indicate that our METER framework is scalable.

**Further Analysis.** We also conduct experiments on image captioning, investigate multi-scale feature fusion, study the model performance on unimodal tasks after VLP, and provide visualization of learned attention maps. All these results are provided in Appendix.

## 5. Conclusion

We present METER, through which we systematically investigate how to train a fully-transformer VLP model in an end-to-end manner. Experiments demonstrate that we can achieve competitive performance with state-of-the-art models with only 4M images for pre-training. When further scaled up, METER achieves new state of the art on VQA.
References


[55] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang,
