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

Zi-Yi Dou[1]*, Yichong Xu[2], Zhe Gan[2], Jianfeng Wang[2], Shuohang Wang[2],
Lijuan Wang[2], Chenguang Zhu[2], Nanyun (Violet) Peng[1], Zicheng Liu[2], Michael Zeng[2]
[1]University of California, Los Angeles, [2]Microsoft Corporation
{zdou,violetpeng}@cs.ucla.edu
{yicxu,zhgan,jianfw,shuohang.wang,lijuanw,chezhu,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 are often degraded significantly. In this paper, we present METER *(**M**ultimodal **E**nd-to-end **T**ransform**ER**), through which we systematically 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 (e.g., merged attention vs. co-attention), architecture design (e.g., encoder-only vs. encoder-decoder), and pre-training objectives (e.g., masked image modeling). We conduct comprehensive experiments on a wide range of VL tasks, and provide insights on how to train a performant VL transformer while maintaining fast inference speed. Notably,* METER *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 VinVL model by +1.04%, and outperforming the previous best fully transformer-based ALBEF model by +1.6%.*[1]

1. Introduction

Vision-and-language (VL) tasks, such as visual question answering (VQA) [1] and image-text retrieval [26, 33], 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 [4, 23, 25, 31, 41, 43]. Specifically, large amounts of image-caption pairs are fed into a model that takes both images and text as inputs, after which the pre-trained representations can contain rich multimodal knowledge that is helpful for downstream tasks.

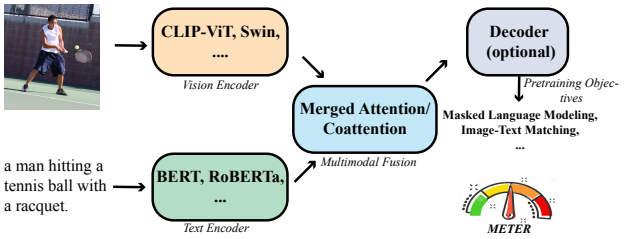


Figure 1. An overview of the proposed METER framework. We systematically investigate how to train a performant fully transformer-based model for vision-and-language tasks, and dissect the model designs along multiple dimensions: vision encoder, text encoder, multimodal fusion, encoder-only vs. encoder-decoder, and pre-training objectives.

Transformers [47] are prevalent in natural language processing, and have recently shown promising performance in computer vision. While almost all the VLP models adopt transformers as their main multimodal fusion architectures, most previous work [4, 23, 25, 31, 41, 43] use pre-trained object detectors (*e.g.*, Faster RCNN [37]) on the vision side to extract *region* features from images, and feed them to the models. This leads to several problems. First, the object detectors cannot be perfect, but are usually kept frozen during VLP, which limits the capacity of VLP models. Second, it is time-consuming to extract region features [19]. To alleviate these problems, researchers have tried to extract *grid* features directly from convolutional networks [16, 40]. Vision transformer (ViT) [10, 29] has been an increasingly hot research topic, that has great potential to outperform convolutional networks for pure vision tasks. Then, a natural question comes in: *can we train a fully transformer-based VLP model by using ViT as the image encoder? What do the recent rapid advances in ViT mean for VLP?*

Recent work [19, 22, 52] have tried to answer this question; however, they have not shown satisfactory performance, and typically underperform state-of-the-art region-

* Work was done when the author interned at Microsoft.
[1]Code: https://github.com/zdou0830/METER.

Model	Vision Encoder	Text Encoder	Multimodal Fusion	Decoder	Pre-training Objectives
ViLBERT [31]	OD+Xformer	Xformer	Co-attn.	✗	MLM+ITM+MIM
LXMERT [43]					MLM+ITM+MIM+VQA
VisualBERT [23]	OD	Emb.	Merged-attn.		MLM+ITM
VL-BERT [23]					MLM+MIM
UNITER [4]					MLM+ITM+MIM+WRA
OSCAR [4]					MLM+ITM
VinVL [55]					MLM+ITM
VL-T5 [5]				✓	MLM+ITM+VQA+Grounding+Captioning
PixelBERT [16]	CNN	Emb.	Merged-attn.	✗	MLM+ITM
SOHO [15]					MLM+ITM+MIM
CLIP-ViL [40]					MLM+ITM+VQA
SimVLM [50]				✓	PrefixLM
ViLT [19]	Patch Emb.	Emb.	Merged-attn.	✗	MLM+ITM
Visual Parsing [52]					MLM+ITM+MIM
ALBEF [22]	Xformer	Xformer	Co-attn.		MLM+ITM+ITC
METER (Ours)					MLM+ITM
CLIP [34]	CNN/Xformer	Xformer	None	✗	ITC
ALIGN [17]	CNN				

Table 1. Glossary of representative VLP models. OD: object detection. Xformer: transformer. Emb.: embedding. MLM/MIM: masked language/image modeling. ITM: image-text matching. WRA: word-region alginment. ITC: image-text contrastive learning.

based VLP models (*e.g.*, VinVL [55]) on tasks such as VQA by a large margin. To close the performance gap, we present METER (**M**ultimodal **E**nd-to-end **T**ransform**ER**), 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 [34], Swin transformer [29]), text encoders (*e.g.*, RoBERTa [28], DeBERTa [13]), multimodal fusion (*e.g.*, merged attention vs. co-attention), architecture 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 METER and its variants on four commonly used image-caption datasets, including COCO [26], Conceptual Captions [39], SBU Captions [32], and Visual Genome [20], and testing them on visual question answering [1], visual reasoning [42], image-text retrieval [26, 33], and visual entailment [51]. Through extensive analysis, we summarize our findings as follows.

- Vision transformer (ViT) plays a more vital role than language transformer, and the performance of ViT on ImageNet classification 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 fair comparison, an encoder-only VLP model performs better than an encoder-decoder model for VQA and zero-shot image-text retrieval tasks.
- Masked image modeling is *not* a critical pre-training objective for VLP.

These insights, in combination with other useful tips and tricks (detailed later), help us train a strong METER model that achieves an accuracy of 77.64% on the VQAv2 test-std set, surpassing the previous best region-feature-based VinVL model [55] by 1.04%, outperforming the previous best ViT-based ALBEF model [22] by +1.6%, and is competitive with the state-of-the-art SimVLM-base model [50] scoring 78.14%, pre-trained over 1.8B images.

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.

Region Features. Most previous work on VLP use pre-trained object detectors to extract visual features. Among them, ViLBERT [31] and LXMERT [43] use co-attention for multimodal fusion, where two transformers are applied independently to region features and text, and another transformer fuses the representations of the two modalities in a later stage. On the other hand, VisualBERT [23], VL-BERT [41], and UNITER [4] use a merged attention fusion module that feeds both region features and text together into a single transformer. OSCAR [25] and VinVL [55] feed additional image tags into the transformer model, and demonstrate state-of-the-art performance across VL tasks. While region-based VLP models have achieved impressive performance, extracting region features can be time-consuming. In addition, the pre-trained object detectors are usually frozen during pre-training, which can limit 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 [40] propose to feed grid features from convolutional neural networks (CNNs) and text di-

rectly 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 [40] use AdamW [30] 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 [29], motivating researchers to develop ViT-based VLP models.

ViT-based Grid Features. ViLT [19] 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, [52] and ALBEF [22] use ViT as 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. In this paper, we aim to close the performance gap, and investigate how we can pre-train a ViT-based model in an end-to-end manner that achieves state-of-the-art performance while maintaining fast inference speed.

3. The METER Framework

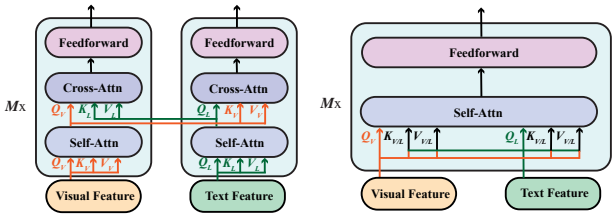
In this section, we illustrate our METER framework and its default settings, which paves the way for our extensive analysis hereinafter.

Overview. For a VLP model, given a text sentence $\mathbf{l}$ and an image $\mathbf{v}$, both text features $\mathbf{l} = \langle l_1, \cdots, l_N \rangle$ and visual features $\mathbf{v} = \langle v_1, \cdots, v_M \rangle$ are first extracted via a *text encoder* and a *vision encoder*, respectively, then the text and visual features are fed into a *multimodal fusion module* to produce cross-modal representations. The cross-modal representations are then optionally fed into a decoder[2] before generating the final outputs. In Figure 1, we show how METER instantiates each specific module, with details introduced below.

3.1. Model Architecture

Vision Encoder. In this paper, we focus on grid 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 since its birth [2, 10, 29, 44, 44, 45, 54], and has been introduced into VLP recently [19, 22, 52]. However, all these models only achieve inferior performance compared to state-of-the-art region-based models (*e.g.*, VinVL [55]).

[2]The decoder here means a sequence decoder, the specific additional heads for masked language modeling and image-text matching are not considered as a decoder in this paper.



(a) Coattention model. **(b)** Merged attention model.

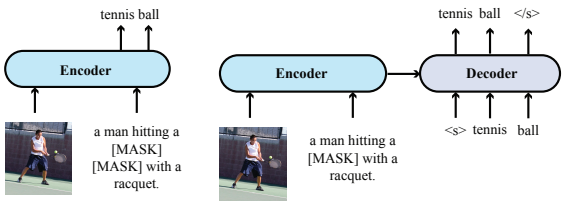
Figure 2. Illustration of two types of multimodal fusion modules: (a) Co-attention, and (b) Merged attention.

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 [44], Distilled-DeiT [44], CaiT [45], VOLO [54], BEiT [2], Swin Transformer [29] and CLIP-ViT [34], to provide a comprehensive analysis on the role of vision transformers.

Text Encoder. Following BERT [9] and RoBERTa [28], VLP models [4, 23, 25, 31, 41, 43] first segment the input sentence into a sequence of subwords [38], 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 [4, 23], or to several text-specific layers [31, 43] before the fusion. For the former, the fusion module is typically initialized via BERT, and the role of text encoding and multimodal fusion is therefore entangled and absorbed in a single BERT model. Here, we aim to dissect the two roles separately, and use a text encoder first before sending the features into the fusion module.

Language model pre-training has achieved significant advances since the birth of BERT; however, most VLP models still only use BERT for initialization [4]. In this work, we study the use of BERT [8], RoBERTa [28], ELECTRA [6], ALBERT [21], 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 [4, 55].

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 the two modalities. Since end-to-end VLP models are becoming increasingly popular, in



(a) Encoder-Only **(b)** Encoder-Decoder

Figure 3. Illustration of the Encoder-Only and Encoder-Decoder model architectures for VLP.

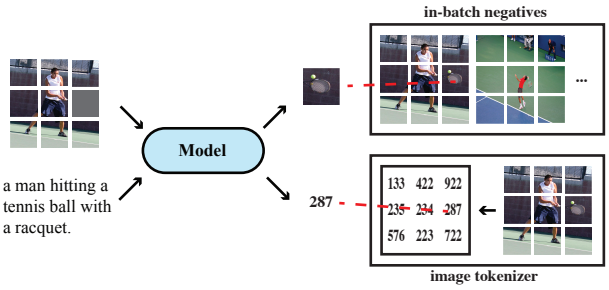


Figure 4. Illustration of masked patch classification.

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 [23] 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 [5] and SimVLM [50], 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, 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 METER model. 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, 28]. 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 $\mathbf{l}^{mask}$ and its corresponding visual input $\mathbf{v}$.

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 $\langle \mathbf{v}, \mathbf{l} \rangle$ 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 [43] and UNITER [4], mask some of the input regions, and the model is trained to regress the original region features. Formally, given a sequence of visual features $\mathbf{v} = \langle v_1, \cdots, v_M \rangle$, where v_i is typically a region feature, we randomly mask some of the visual features, and the model outputs the reconstructed visual features $\mathbf{o_v}$ given the rest of the visual features and the unmasked tokens $\mathbf{t}$, and regression aims to minimize the loss below:

$$\mathcal{L}_{\text{MIM}} = \text{MSE}(\mathbf{v}, \mathbf{o_v}), \tag{1}$$

where MSE is short for mean squared error. Researchers [4, 31, 43] 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.

Notably, recent state-of-the-art models (*e.g.*, ALBEF [22], VinVL [55]) do not apply MIM during VLP. In addition, in ViLT [19], 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 $\{\langle \mathbf{v}^k, \mathbf{l}^k \rangle\}_{k=1}^{B}$, where B is the batch size. We treat all the patches in $\{\mathbf{v}^k\}_{k=1}^{B}$ 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 $\{\mathbf{v}^k\}_{k=1}^B$ as $\{c(\mathbf{v}^k)\}_{k=1}^B$ and $\{h(\mathbf{v}^k)\}_{k=1}^B$, respectively, we can represent the output probability at position i for the k-th instance as:

$$p(\mathbf{v}_i^k|[\mathbf{v}^{k,mask};\mathbf{l}^k]) = \frac{e^{h(\mathbf{v}_i^k)^{\mathrm{T}}c(\mathbf{v}_i^k)}}{\sum_{j,k'} e^{h(\mathbf{v}_i^k)^{\mathrm{T}}c(\mathbf{v}_j^{k'})}}. \quad (2)$$

The model is trained to maximize this 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 representations of the input patches, and the model is then trained to reconstruct the discrete tokens. Specifically, we first use the VQ-VAE [46] model in DALL-E [36] 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 of the masked patches.

3.3. The Default Settings for METER

As discussed above, there are many different model designs that we can explore for METER. Below, we detail our default settings, which paves the way for our extensive analysis hereinafter.

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 [4, 19], we pre-train METER on four commonly used datasets, including COCO [26], Conceptual Captions [39], SBU Captions [32], and Visual Genome [20]. The statistics of these datasets is shown in Table 2. The combined training data consists of about 4 million images in total.

Dataset	#Images	#Captions
COCO	113K	567K
Visual Genome	108K	5.4M
Conceptual Captions	3.1M	3.1M
SBU Captions	875K	875K

Table 2. Statistics of the pre-training datasets.

Downstream Tasks. For ablation and analysis, we mainly focus on VQAv2 [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 VQAv2, we follow the standard practice [4] 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, we also evaluate METER on visual reasoning (NLVR^2 [42]), visual entailment (SNLI-VE [51]), and image-text retrieval (COCO [26] and Flickr30k [33]) tasks. For retrieval tasks, we evaluate models in both zero-shot and finetuning settings.

Implementation Details. For all pre-training experiments, we pre-train METER using the AdamW optimizer [30] for 100k steps. The learning rates for the bottom and top layers are set to 1e-5 and 5e-5 during pre-training. The warm-up ratio is set to 10%, and the learning rate is linearly decayed to 0 after 10% of the total training steps. We use center-crop to resize each image into the size of 224×224 or 384×384 depending on the adopted vision transformers.

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 encoders without VLP for efficiency. Concretely, we initialize the bottom layers with specific pre-trained vision and text encoders, and randomly initialize the top layers. Then, we directly finetune the models on three tasks, including VQAv2, SNLI-VE, and Flickr30k retrieval. The learning rates for the bottom and top layers are set to 1e-5 and 1e-4, respectively, and the number of training epochs is set to 10 for all the tasks.

Text Enc.	VQAv2 Acc.	VE Acc.	IR R@1	TR R@1	SQuAD EM	MNLI Acc.
BERT	69.56	76.27	49.60	66.60	76.3	84.3
RoBERTa	69.69	76.53	49.86	68.90	84.6	87.6
ELECTRA	69.22	76.57	41.80	58.30	86.8	**88.8**
DeBERTa	69.40	**76.74**	51.50	67.70	**87.2**	**88.8**
ALBERT	**69.94**	76.20	52.20	68.70	86.4	87.9
Emb-only	67.13	74.85	49.06	68.20	-	-
CLIP	69.31	75.37	**54.96**	**73.80**	-	-

Table 3. Comparisons of different text encoders without VLP. CLIP-ViT-224/32 is used as the vision encoder. All the text encoders are in base model size, except ALBERT, which is xlarge. Emb-only: only using word embeddings as text encoder. IR/TR: Flickr30k image/text retrieval. EM: exact match. The results of SQuAD and MNLI are copied from their corresponding papers. All the results on VL tasks are from their test-dev/val sets.

Vision Encoder	VQAv2	VE	IR	TR	ImageNet
ViT B-384/16	69.09	76.35	40.30	59.80	83.97
DeiT B-384/16	68.92	75.97	33.38	50.90	82.9
Dis. DeiT B-384/16	67.84	76.17	34.84	52.10	85.2
CaiT M-384/32	71.52	76.62	38.96	61.30	86.1
VOLO 4-448/32	71.44	76.42	40.90	61.40	**86.8**
Swin B-384/32	**72.38**	**77.65**	52.30	69.50	86.4
CLIP B-224/32	69.69	76.53	49.86	68.90	-
CLIP B-224/16	71.75	77.54	**57.64**	**76.90**	-
BEiT B-224/16	68.45	75.28	32.24	59.80	85.2

Table 4. Comparisons of different vision encoders without VLP. RoBERTa is used as the default text encoder. IR/TR: Flickr30k image/text retrieval. The results of ImageNet classification are copied from their corresponding papers. All the results on VL tasks are from their test-dev/val sets.

When comparing different text encoders, we choose CLIP-ViT-224/32 [34] as the visual encoder; when comparing different visual encoders, we choose RoBERTa [28] as the text encoder.

Impact of Text Encoders. As shown in Table 3, there are no 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 4, 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 [48] or ImageNet classification [7], that does not necessarily indicate that they are more suitable for VL tasks.

Text Enc.	Vision Enc.	VQAv2	Flickr-ZS IR	Flickr-ZS TR
Embed-only	CLIP-32	73.99	60.32	90.38
BERT	CLIP-32	74.98	66.08	78.10
	CLIP-16	76.70	74.52	87.20
RoBERTa	CLIP-32	74.67	65.50	76.60
	CLIP-16	**77.19**	**76.64**	**89.60**
	Swin	76.43	71.68	85.30

Table 5. Comparisons of different vision and text encoders with VLP. Results on VQAv2 are on test-dev set. ZS: zero-shot.

Bottom LR	Top LR	VQAv2	Flickr-ZS IR	Flickr-ZS TR
1e-5	1e-5	73.16	48.80	63.70
2e-5	2e-5	73.66	53.14	67.20
3e-5	3e-5	73.77	56.48	70.90
5e-5	5e-5	73.54	52.48	65.90
1e-5	5e-5	**74.98**	**66.08**	**78.10**

Table 6. Using different learning rates for the randomly-initialized and pre-trained parameters is better than using the same learning rate. Results on VQAv2 are on test-dev set. ZS: zero-shot.

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 5, 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 [55] models. Note that VinVL uses VQA datasets during pre-training, while our model achieves better performance by just using the image-caption data, indicating the potential of ViT-based VLP 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 [27]. As shown in Table 6, 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

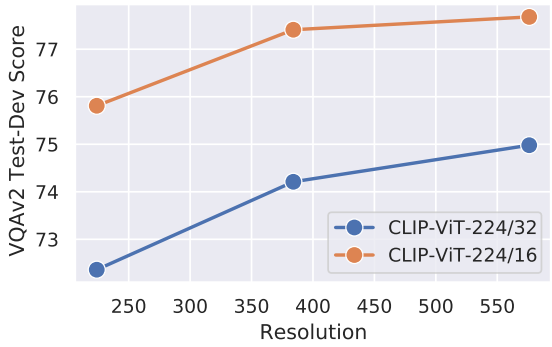


Figure 5. Increasing the image resolution during finetuning can improve the model performance by a large margin on the VQAv2 test-dev set.

and language, and finetuning them aggressively can result in the loss of these valuable information.

Second, similar to several previous work [19,54], 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 and 1 point for the CLIP-ViT-224/32 and CLIP-ViT-224/16 model, respectively. Note that to increase the image resolution, we first need to interpolate the positional embedding matrices of the vision transformers.

4.2. Analysis of Multimodal Fusion Modules

Now, following the default setting in Section 3.3, we perform investigations on multimodal fusion. First, we design both *merged attention* and *co-attention* models and investigate their performance. For the merged attention model (Figure 2b), the top transformer consists of M_{merged} encoding layer, with each layer consisting of one self-attention block and one feed-forward network block. To help the model distinguish between the two modalities, we add a modality embedding to the input features before feeding them to the top transformer. For the co-attention model (Figure 2a), we feed the text and visual features to two M_{co}-layer transformers separately, and each top transformer encoding layer consists of one self-attention block, one cross-attention block, and one feed-forward network block. Compared with merged attention, co-attention allows separate transformation functions for the vision and language modalities. We set $M_{merged} = 12$ and $M_{co} = 6$ so that the numbers of parameters of the two models are roughly comparable to each other.

Results. Table 7 reports the downstream performance of the two models.The co-attention model performs better than the merged attention model in our setting, indicating that it is important to have different sets of parameters for the two modalities. Note that this contradicts with the findings in region-based VLP models [3], possibly because (i) findings of region-based VLP models cannot directly apply to ViT-based VLP models; (ii) most region-based VLP models only use pre-trained visual encoders, and also do not have a pre-trained text encoder included, thus the inconsistency between the two modalities will not favor symmetrical architecture like the co-attention model.

Fusion	Decoder	VQAv2	Flickr-ZS	
			IR	TR
Merged attention	✗	74.00	57.46	73.10
Co-attention	✗	**74.98**	**66.08**	**78.10**
Co-attention	✓	74.73	48.96	71.60

Table 7. Co-attention performs better than merged attention in our setting, and adding a decoder is not helpful for our discriminative VL tasks. Results on VQAv2 are on test-dev set. ZS: zero-shot.

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 [5], we adopt T5-style [35] 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 7, the encoder-only model can outperform the encoder-decoder model on our two discriminative tasks, which is consistent with the findings in [5]. 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.

4.4. Ablation 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 10, 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 VQAv2 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

Model	VQAv2		NLVR2		SNLI-VE		Flickr-ZS					
	test-dev	test-std	dev	test	dev	test	IR@1	IR@5	IR@10	TR@1	TR@5	TR@10
Pre-trained with >10M images												
ALBEF (14M) [22]	75.84	76.04	82.55	83.14	80.80	80.91	82.8	96.3	98.1	94.1	99.5	99.7
SimVLM$_{\text{BASE}}$ (1.8B) [50]	77.87	78.14	81.72	81.77	84.20	84.15	-	-	-	-	-	-
SimVLM$_{\text{HUGE}}$ (1.8B) [50]	80.03	80.34	84.53	85.15	86.21	86.32	-	-	-	-	-	-
Pre-trained with <10M images												
UNITER$_{\text{LARGE}}$ [4]	73.82	74.02	79.12	79.98	79.39	79.38	68.74	89.20	93.86	83.60	95.70	97.70
VILLA$_{\text{LARGE}}$ [11]	74.69	74.87	79.76	81.47	80.18	80.02	-	-	-	-	-	-
UNIMO$_{\text{LARGE}}$ [24]	75.06	75.27	-	-	**81.11**	80.63	-	-	-	-	-	-
VinVL$_{\text{LARGE}}$ [55]	76.52	76.60	**82.67**	**83.98**	-	-	-	-	-	-	-	-
PixelBERT [16]	74.45	74.55	76.5	77.2	-	-	-	-	-	-		
CLIP-ViL (ResNet50x4) [40]	76.48	76.70	-	-	80.61	80.20	-	-	-	-	-	-
ViLT [55]	71.26	-	75.70	76.13	-	-	55.0	82.5	89.8	73.2	93.6	96.5
Visual Parsing [52]	74.00	74.17	77.61	78.05	-	-	-	-	-	-	-	-
ALBEF (4M) [22]	74.54	74.70	80.24	80.50	80.14	80.30	76.8	93.7	96.7	90.5	**98.8**	**99.7**
METER-Swin	76.43	76.42	82.23	82.47	80.61	80.45	71.68	91.80	95.30	85.30	97.70	99.20
METER-CLIP-ViT	**77.68**	**77.64**	82.33	83.05	80.86	**81.19**	**79.60**	**94.96**	**97.28**	**90.90**	98.30	99.50

Table 8. Comparisons with previous models on visual question answering, visual reasoning, visual entailment, and Flickr30k zero-shot retrieval tasks. The best scores are in **bold**, and the second best scores are underlined.

Model	Flickr						COCO					
	IR@1	IR@5	IR@10	TR@1	TR@5	TR@10	IR@1	IR@5	IR@10	TR@1	TR@5	TR@10
Pre-trained with >10M images												
ALBEF (14M) [22]	85.6	97.5	98.9	95.9	99.8	100.0	60.7	84.3	90.5	77.6	94.3	97.2
Pre-trained with <10M images												
UNITER$_{\text{LARGE}}$ [4]	75.56	94.08	96.76	87.30	98.00	99.20	52.93	79.93	87.95	65.68	88.56	93.76
VILLA$_{\text{LARGE}}$ [11]	76.26	94.24	96.84	87.90	97.50	98.80	-	-	-	-	-	-
UNIMO$_{\text{LARGE}}$ [24]	78.04	94.24	97.12	89.40	98.90	99.80	-	-	-	-	-	-
VinVL$_{\text{LARGE}}$ [55]	-	-	-	-	-	-	**58.8**	**83.5**	**90.3**	75.4	92.9	96.2
PixelBERT [16]	71.5	92.1	95.8	87.0	98.9	99.5	50.1	77.6	86.2	63.6	87.5	93.6
ViLT [55]	64.4	88.7	93.8	83.5	96.7	98.6	42.7	72.9	83.1	61.5	86.3	92.7
Visual Parsing [52]	73.5	93.1	96.4	87.0	98.4	99.5	-	-	-	-	-	-
ALBEF (4M) [22]	**82.8**	**96.7**	**98.4**	**94.3**	99.4	99.8	56.8	81.5	89.2	73.1	91.4	96.0
METER-Swin	79.02	95.58	98.04	92.40	99.00	99.50	54.85	81.41	89.31	72.96	92.02	96.26
METER-CLIP-ViT	82.22	96.34	**98.36**	**94.30**	**99.60**	**99.90**	57.08	82.66	90.07	**76.16**	**93.16**	**96.82**

Table 9. Comparisons with previous models on Flickr30k and COCO retrieval tasks. The best scores are in **bold**, and the second best scores are underlined.

Model	VQAv2	Flickr-ZS	
		IR	TR
MLM	74.19	-	-
ITM	72.63	53.74	71.00
MLM+ITM	**74.98**	**66.08**	**78.10**
MLM+ITM + MIM with in-batch negatives	74.01	62.12	76.90
MLM+ITM + MIM with discrete code	74.21	59.80	76.30

Table 10. 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 VQAv2 are on test-dev set. ZS: zero-shot.

performance drop is due to the conflicts between different objectives, and some techniques in multi-task optimization [49,53] 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 State-of-the-Arts

In this section, we evaluate our best-performing models (*i.e.*, RoBERTa-base+Swin Transformer and RoBERT-base+CLIP-ViT-224/16 with co-attention fusion module and with the image resolutions set to 384×384 and 288×288 respectively), and compare with state-of-the-art models. We evaluate the models on visual question answering (VQAv2), visual reasoning (NLVR2), visual entailment (SNLI-VE), Flickr30k retrieval tasks in both zero-shot and fine-tuning settings, and COCO retrieval tasks.

Results. As summarized in Table 8 and Table 9, compared with models trained with our CLIP-based model (METER-CLIP-ViT) 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 VQAv2 test-std set using only 4M images for pre-training, surpassing the state-of-the-art region-feature-based VinVL model by +1.04%, and outperforming the previous best fully transformer-

based model (*i.e*. ALBEF) by 1.6%. In addition, while ALBEF has specially-designed objectives for retrieval, our model can outperform ALBEF on text and image retrieval tasks with the same amount of pre-training images, further demonstrating the effectiveness of our proposed strategies.

5. Conclusion

In this paper, we present METER, and investigate how to train a fully-transformer VLP model in an end-to-end manner. Specifically, we dissect the model designs along multiple dimensions, including vision encoder, text encoder, multimodal fusion, encoder-only vs. encoder-decoder, and pre-training objectives. Experiments on several downstream tasks demonstrate that we can achieve competitive performance with state-of-the-art models via using only 4M images for pre-training. In the future, we expect stronger vision transformer backbones can be further incorporated to boost the performance. In addition, we plan to perform detailed comparisons among visual features of different types and testing if we can combine the strengths of different features and further improve the model performance.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[3] Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. Multimodal pretraining unmasked: Unifying the vision and language BERTs. *arXiv preprint*, 2020.

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Universal image-text representation learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[5] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[6] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

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

[12] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[13] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[14] Lisa Anne Hendricks, John Mellor, Rosalia Schneider, Jean-Baptiste Alayrac, and Aida Nematzadeh. Decoupling the role of data, attention, and losses in multimodal transformers. *Transactions of the Association for Computational Linguistics*, 2021.

[15] Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing out of the box: End-to-end pre-training for vision-language representation learning. In *CVPR*, 2021.

[16] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2102.05918*, 2021.

[18] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint*, 2016.

[19] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv preprint arXiv:2102.03334*, 2021.

[20] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.

[21] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[22] Junnan Li, Ramprasaath R Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *arXiv preprint arXiv:2107.07651*, 2021.

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

[24] Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. *arXiv preprint arXiv:2012.15409*, 2020.

[25] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[27] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, 2019.

[28] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[31] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[32] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24:1143–1151, 2011.

[33] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

[36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.

[38] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, 2016.

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

[40] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

[41] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

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

[43] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[44] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[45] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021.

[46] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6309–6318, 2017.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[48] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.

[49] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2020.

[50] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

[51] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint*, 2019.

[52] Hongwei Xue, Yupan Huang, Bei Liu, Houwen Peng, Jianlong Fu, Houqiang Li, and Jiebo Luo. Probing intermodality: Visual parsing with self-attention for vision-language pre-training. *arXiv preprint arXiv:2106.13488*, 2021.

[53] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[54] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition, 2021.

[55] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. VinVL: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.