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], Pengchuan Zhang[2], Lu Yuan[2], Nanyun 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,shuowa,lijuanw,chezhu,penzhan,luyuan,zliu,nzeng}@microsoft.com

	COCO	VG	CC	SBU
#Images	113K	108K	3.1M	875K
#Captions	567K	5.4M	3.1M	875K

Table 1. Statistics of the pre-training datasets.

A. Implementation Details

Datasets. The statistics of our pre-training datasets is shown in Table 1. Following many previous work [3, 8, 10], we pre-train the models with four datasets, including COCO, Visual Genome, Conceptual Captions and SBU Captions, consisting of about 4M images and 9M image-caption pairs in total.

For the downstream tasks, we test the models on VQAv2 [1] for visual question answering, NLVR^2 [15] for visual reasoning, COCO [11] and Flickr30k [13] for image-text retrieval, and SNLI-VE [16] for visual entailment. We use the standard splits for all the datasets except for VQAv2, where we follow standard practice [3,8,10] to train the models with both its training and development data, and treat its test-dev set as the development set. Note that we do not use the Visual Genome VQA data for data augmentation in our VQA settings.

Pre-training Settings. We pre-train our best models using the AdamW optimizer [12] with the learning rates set to 1e-5 for the bottom image and text encoders and 5e-5 for the cross-modal module. The warm-up ratio is set to 10%, and the learning rate is linearly decayed to 0 after 10% of the total training steps. The batch size, hidden size, and number of heads are set to 4096, 768, 12, respectively. We pre-train the models for 100k steps on 8 NVIDIA A100 GPUs, which takes around 3 days for METER-CLIP-ViT$_{\text{BASE}-32}$ and 8 days for METER-Swin$_{\text{BASE}}$ and METER-CLIP-ViT$_{\text{BASE}-16}$.

Fine-tuning Settings. For the downstream tasks, we perform grid searches over the learning rates and image resolutions. The learning rates and image resolutions are selected from {1e-6, 2e-6, 5e-6, 1e-5} and {288, 384, 576}, respectively. We apply RandAugment [4] during finetuning following previous work [8, 10].

* Work was done when the author interned at Microsoft.

Model	Time ([8])	Time (ours)	VQAv2
ViLBERT	920	-	70.55
VisualBERT	925	-	70.80
LXMERT	900	-	72.42
UNITER-Base	900	-	72.70
OSCAR-Base	900	-	73.16
VinVL-Base	650	-	75.95
PixelBERT-X152	160	-	74.45
CLIP-ViL (ResNet50x4)	-	57	76.70
ViLT	15	26	71.26
ALBEF (14M)	-	52	76.04
METER-Swin$_{\text{BASE}}$	-	59	76.42
METER-CLIP-ViT$_{\text{BASE}}$	-	53	77.64

Table 2. Inference time (ms) of different models. We report the inference time measured by [8] and in our setting. We also list the model performance on the VQAv2 test-std set.

B. Inference Time

We measure the inference time of different models as in Table 2. First, as shown in [8], their ViT-based model is much faster than previous region-feature-based VLP models. In our setting, we measure the average inference time of processing 1 VQA instance on 1 NVIDIA V100 GPU. We find that while our model can be slower than the ViLT model, it is still significantly faster than region-feature-based models and comparable to other ViT-based ones. In addition, we can achieve much stronger performance on downstream tasks than other models.

C. Image Captioning

While in this paper we mainly focus on finetuning our models for discriminative downstream tasks such as visual question answering, here we investigate if our models can also be applied to generative tasks. Specifically, we finetune our models on the COCO image captioning task.

We finetune our METER-CLIP-ViT$_{\text{BASE}}$ model for 5 epochs using the standard maximum likelihood estimation

Model (#Pre-training Images)	BLEU	METEOR	Cider	SPICE
$\text{OSCAR}_{\text{BASE}}$ (4M)	36.5	30.3	123.7	23.1
$\text{VinVL}_{\text{BASE}}$ (5.6M)	38.2	30.3	129.3	23.6
$\text{SimVLM}_{\text{BASE}}$ (1.8B)	39.0	32.9	134.8	24.0
$\text{METER-CLIP-ViT}_{\text{BASE}}$ (4M)	38.8	30.0	128.2	23.0

Table 3. Image captioning results of different models trained with maximum likelihood estimation on COCO.

objective. At each decoding step, instead of using the causal attention mechanism, the input image and all the text tokens can attend to all the generated text tokens so as to minimize the discrepancy between pre-training and finetuning. We use beam search with the beam size set to 5.

As shown in Table 3, we can achieve reasonable performance on image captioning even though our model employs an encoder-only architecture. We expect that an encoder-decoder model would be more suitable for generative tasks, which we leave as future work.

D. Multi-scale Feature Fusion

For the pre-trained text and visual encoders, different layers can contain different types of information. For example, [7] finds that the intermediate layers of BERT encode a rich hierarchy of linguistic information, with surface features at the bottom, syntactic features in the middle and semantic features at the top. Aggregating the features at different layers has demonstrated to be helpful in both vision [6, 17] and language [2, 5]. Therefore, in this part, we investigate if we can use feature fusion techniques to better utilize the information embedded at different layers of the pre-trained encoders.

Method. Based on some preliminary explorations, here we adopt a simple fusion strategy and only fuse the representations of the text and image encoders but not the cross-modal layers on the top. Specifically, given a text token or image patch x_i, we first feed it into a text or image encoder on the bottom of our model (*e.g.*, BERT), and get its representations $\{h(x_i^j)\}_{j=0}^{N}$ at different layers, where N is the number of layers of the encoder. Then, we compute a gate value for each layer and perform a weighted sum to get the final representation of x_i:

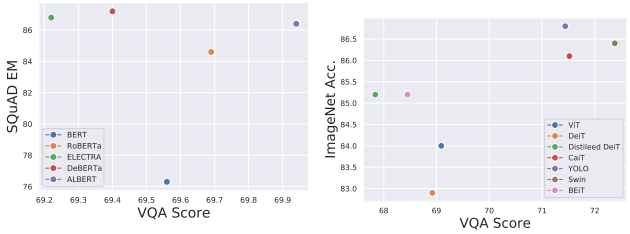
$$o(x_i) = h(x_i^N) + \sum_{j=0}^{N-1} g(h(x_i^j))h(x_i^j), \qquad (1)$$

where g is a linear transformation function. We then feed $o(x_i)$ to the top cross-modal layers. Note that the fusion can be done in both the text and visual encoders.

Results. We pre-train the models using the co-attention model with RoBERTa as the text encoder and Swin Transformer as the visual encoder. We evaluate the models both with and without VLP following the default settings. Because Swin Transformers have different numbers of image representations at different layers, we perform an average pooling so that each layer has 12×12 patch representations. As shown in Tab. 4, while the fusion strategy can improve the model performance by a small margin without VLP, it can degrade the model performance after pre-training. We hypothesize that this is because after the pre-training, the VLP model can learn how to well utilize the representations in the pre-trained encoders and layer fusion is not necessary.

Model	VQAv2 test-dev	Flickr-ZS IR	Flickr-ZS TR
w/o pre-training			
$\text{METER-Swin}_{\text{BASE}}$ w/o fusion	72.38	-	-
$\text{METER-Swin}_{\text{BASE}}$ w/ fusion	72.91	-	-
$\text{METER-CLIP-ViT}_{\text{BASE}}$ w/o fusion	71.75	-	-
$\text{METER-CLIP-ViT}_{\text{BASE}}$ w/ fusion	72.92	-	-
with pre-training			
$\text{METER-Swin}_{\text{BASE}}$ w/o fusion	76.43	71.68	85.30
$\text{METER-Swin}_{\text{BASE}}$ w/ fusion	76.31	70.58	83.70
$\text{METER-CLIP-ViT}_{\text{BASE}}$ w/o fusion	77.19	76.64	89.60
$\text{METER-CLIP-ViT}_{\text{BASE}}$ w/ fusion	77.06	76.26	88.00

Table 4. The fusion strategy improves the model performance without vision-and-language pre-training but can degrade the model performance after pre-training.



(a) Vision-and-Language vs. Language **(b)** Vision-and-Language vs. Vision

Figure 1. Correlation between model performance on *vision-and-language* tasks and pure *vision* or *language* tasks.

E. Correlation between *Vision-and-Language* Tasks and *Vision* or *Language* Tasks

In this section, we perform a quantitative analysis of the correlation between model performance on *vision-and-language* tasks and pure *vision* or *language* tasks. We vary different text and vision encoders, and plot the model performance on the VQAv2 test-dev set and SQuAD or ImageNet datasets as in Figure 1. We also compute the Pearson correlations in both cases. We find that the Pearson correlation coefficients and p-values are -0.09 and 0.88 in the VL vs. L setting, and 0.41 and 0.36 in the VL vs. V setting, indicating that there exists little to none correlations between the model performance on VL tasks and V or L tasks.

Text Encoder	QQP	MNLI	QNLI	SST2	CoLA	MRPC	STSB	RTE
Before VLP	91.31±0.15	87.53±0.24	92.61±0.34	94.38±0.20	58.72±0.73	91.03±0.59	90.15±0.18	71.24±3.07
After VLP	91.34±0.08	87.38±0.18	92.67±0.06	93.92±0.50	57.88±0.79	90.57±0.78	89.93±0.46	70.28±2.00

Table 5. Performance of text encoders (RoBERTa-base) on GLUE dev sets before and after VLP. The image encoder during VLP is CLIP-ViT-224/16. We report average scores and standard deviations over three runs of different random seeds.

Image Encoder	Before VLP		After VLP	
	CF10	CF100	CF10	CF100
Swin-Base-384/32	97.00	89.15	97.99	90.26
CLIP-ViT-224/16	95.85	82.60	94.92	81.90

Table 6. Linear probe performance on CIFAR-10 and CIFAR-100. The text encoder during VLP is RoBERTa-base.

Pre-training Datasets	VQAv2	Flickr-ZS	
		IR	TR
COCO	72.95	46.38	60.20
CC	73.05	39.84	55.50
SBU	70.14	21.52	35.90
VG	73.54	39.24	49.30
COCO+CC+SBU+VG	74.98	66.08	78.10

Table 7. Results of models pre-trained with different datasets.

F. Unimodal Tasks

We also investigate the model performance on unimodal tasks after VLP. For text-only tasks, we finetune the bottom text encoders on GLUE tasks; for image-only tasks, we fit a linear classifier on the learned representations of image encoders on CIFAR-10 and CIFAR-100 [9].

We report results in Table 5 and 6. As shown in the tables, our text encoder gets slightly worse performance on the GLUE tasks on average; for image-only tasks, VLP seems to improve the model performance for Swin Transformer but not for CLIP-ViT, possibly because of domain issues. Note that in both sets of the experiments, we only use our text or image encoder and discard the rest of the networks, and how to utilize multi-modal encoder to improve uni-modal performance is an open problem and we leave it as a future direction.

G. Analysis on Pre-training Datasets

We also perform analysis on our pre-training datasets. We pre-train our model on each of the pre-training datasets. We choose CLIP-ViT-224/32 as the image encoder and BERT-base-uncased as the text encoder, and employ the co-attention fusion module. We pre-train the model for 50k steps on each dataset and report the evaluation results on VQAv2 and Flickr30k zero-shot retrieval tasks.

As we can see from Table 7, both data size and domain similarity contribute to the downstream task performance. CC and VG are the largest datasets and COCO most matches the downstream task domains, thus models pre-trained on the three datasets obtain the highest scores.

H. Visualization

In this section, we use Grad-CAM [14] to visualize our models. Specifically, we visualize the cross-attention maps of the pre-trained models corresponding to individual words when performing masked language modeling. As shown in Figure 2 and 3, both our Swin Transformer-based and CLIP-ViT-based models can correctly attend to the corresponding regions given different words, suggesting that our models can learn visual grounding implicitly during pre-training.

I. Limitations

While we have demonstrated the effectiveness of our models across different tasks, our models still have several limitations:

Generative Tasks. We mainly focus on discriminative tasks such as visual question answering and visual reasoning in this paper, while generative tasks such as image captioning are under-investigated. We perform experiments on the COCO image captioning data in Appendix, and will investigate more on this in the future.

Scalability. In our current settings, we pre-train the models with 4M or 14M images, thus it is unclear how the model performance would be if we pre-train the models with larger datasets and we are actively experimenting in this direction.

English Data. So far, we only experiment on the English data, and it is worth investigating if our models can generalize to other languages as well, which we leave as a future direction.

References

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

[2] Ankur Bapna, Mia Xu Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. Training deeper neural machine translation models with transparent attention. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[3] 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 *European Conference on Computer Vision (ECCV)*, 2020.

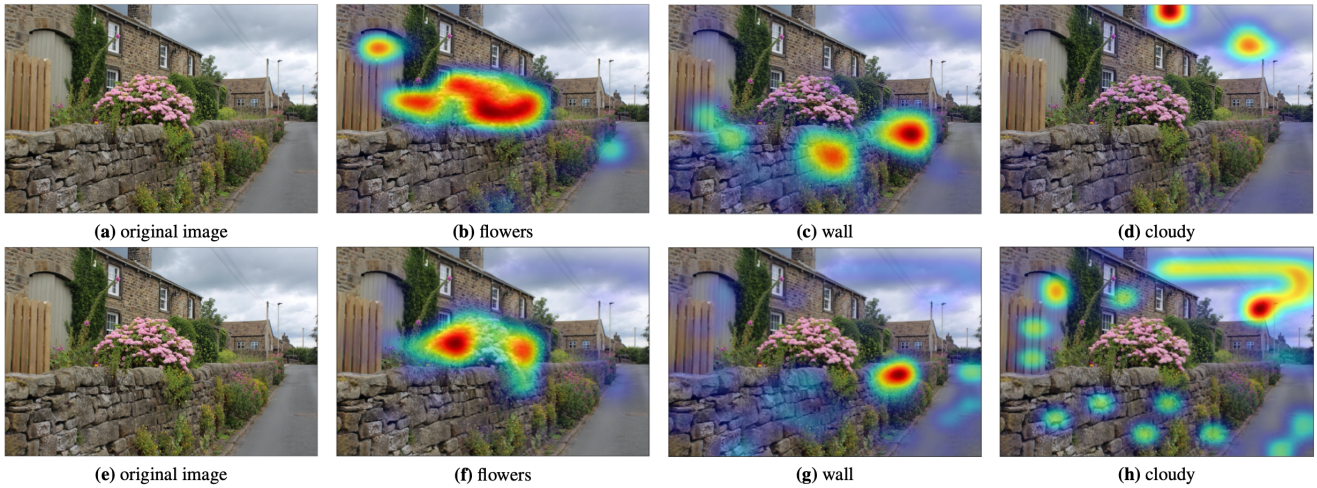


Figure 2. Visualization of the attention maps of text tokens in the caption "a display of *flowers* growing out and over the retaining *wall* in front of cottages on a *cloudy* day." The first and second rows correspond to the results of METER-Swin$_{\text{BASE}}$ and METER-CLIP-ViT$_{\text{BASE}}$, respectively.

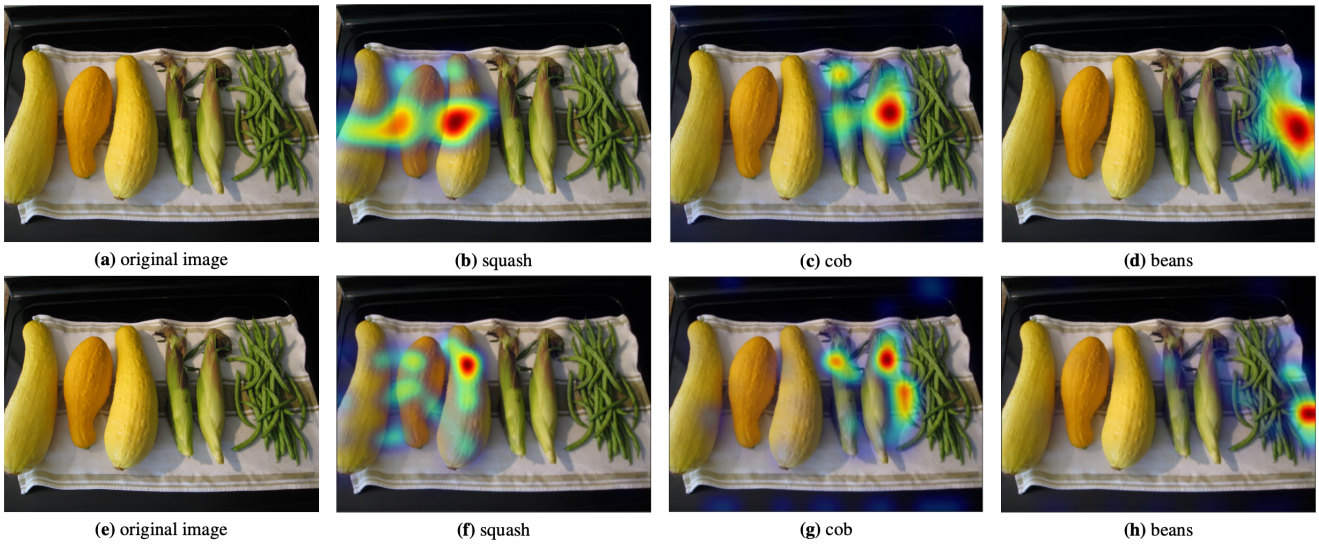


Figure 3. Visualization of the attention maps of text tokens in the caption "yellow *squash*, corn on the *cob* and green *beans* laid out on a white cloth." The first and second rows correspond to the results of METER-Swin$_{\text{BASE}}$ and METER-CLIP-ViT$_{\text{BASE}}$, respectively.

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

[5] Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. Exploiting deep representations for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[6] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[8] Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[10] Junnan Li, Ramprasaath R Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align be-

fore fuse: Vision and language representation learning with momentum distillation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[11] 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 *European Conference on Computer Vision (ECCV)*, 2014.

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

[13] 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 *International Conference on Computer Vision (ICCV)*, 2015.

[14] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision (ICCV)*, 2017.

[15] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

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

[17] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.