Effective End-to-End Vision Language Pretraining with Semantic Visual Loss

Xiaofeng Yang, Fayao Liu, Guosheng Lin

***Abstract*—Current vision language pretraining models are dominated by methods using region visual features extracted from object detectors. Given their good performance, the extract-then-process pipeline significantly restricts the inference speed and therefore limits their real-world use cases. However, training vision language models from raw image pixels is difficult, as the raw image pixels give much less prior knowledge than region features. In this paper, we systematically study how to leverage auxiliary visual pretraining tasks to help training end-to-end vision language models. We introduce three types of visual losses that enable much faster convergence and better finetuning accuracy. Compared with region feature models, our end-to-end models could achieve similar or better performance on downstream tasks and run more than 10 times faster during inference. Compared with other end-to-end models, our proposed method could achieve similar or better performance when pretrained for only 10% of the pretraining GPU hours.**

I. Introduction

Using region visual features [1] in vision language tasks is one of the milestones in developing vision language models. The training process of object detectors brings rich object classes and attributes information to the region features. The use of region features significantly reduces the learning difficulty and improves the performance of various tasks, and it has now become a common practice in VQA [1], [2], visual captioning [1], [3], and vision language pretraining [4], [5]. Given its promising performance, using region visual features has several drawbacks. First, extracting region visual features with an object detector is time-consuming, leading to the difficulty of deploying vision language systems in real-world use-cases. Second, developing an effective feature extraction backbone is tricky and requires careful human engineering.

However, training vision language models from raw image pixels is difficult. For existing region feature based methods, region features are extracted from a trained object detector and they contain high-level semantic information and object instance information. In contrast, when we only use raw pixels as input, the high level semantic information and object instance information are missing. In experiments, we demonstrate that simply replacing the faster-rcnn feature extractor [1] with a grid feature extraction backbone has the problems of slow convergence and low finetuning accuracy. We attribute this observation to the lack of semantic guidance of pixel inputs during pretraining, i.e. telling the model that the image region is semantically a "bird". To this end, we introduce three vision-wise supervision methods to help the training of vision-language models. When used in pretraining, all three losses enable faster convergence and lead to better performance. We show a comparison of validation matching loss during pretraining in Fig 1.

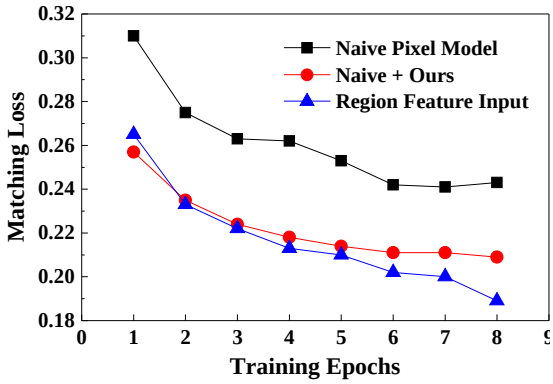

Fig. 1. Comparison of validation loss during pretraining. A naive pixel model is trained by simply replacing the faster rcnn with a grid feature extractor. Results show that the naive pixel model converges slow and has a high final loss. Once our proposed method is used, the pixel input model could achieve a similar convergence speed as a region feature model.

First, we propose to use the **Self-Supervised Loss (SSUL)**. The SSUL trains the vision branch by randomly masking image regions and predicting the mean color (average RGB values) of the masked area. The Self-Supervised Loss does not require additional annotations, hence can be used on any form of image data. Second, we propose to use the **Semantic Segmentation Loss (SEGL)**. Specifically, we use an extra semantic segmentation loss to supervise the training of the visual branch. The advantage of this loss is that it gives pixel-level semantic supervision to the visual branch. We use the annotation in COCO Stuff [6] to calculate this loss. With only half of the vision language pretraining images annotated, we see a great improvement in convergence speed and finetuning accuracy. Finally, we propose the **Knowledge Distillation Based Set Prediction Loss (SPL)**. Knowledge distillation [7] is an important method to extract knowledge from large models and guide the training of small models. For SPL, we first run the pretrained faster-rcnn model [1] to extract object labels from the images. We regard the generated labels as a set

Corresponding author: Guosheng Lin.

Xiaofeng Yang and Guosheng Lin are with School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore 639798 (email: xiaofeng001@e.ntu.edu.sg, gslin@ntu.edu.sg)

Fayao Liu is with Agency for Science, Technology and Research (A*STAR), Singapore 138632 (email: fayaoliu@gmail.com)

of pseudo-labels. The SPL first processes the generated grid feature map with a transformer model to generate a fixed set of region predictions. Then SPL learns to match the generated region predictions with pseudo-labels via optimal matching. The SPL does not require additional human annotations on pretraining images.

For experiments, we show that our proposed methods could achieve similar performance compared with the region feature models. During inference time, our model runs more than 10 times faster than the region feature models. Compared with other pixel input vision language pretraining models [8], our models only require 1/10 of the pre-training GPU hours and get better finetuning accuracy. We also evaluate different vision backbones, e.g. the CNN backbones [9] and Vision Transformer backbones [10].

To summarize our contributions:

- We study the difficulty in training pixel input vision language pretraining models and propose three types of visual losses to tackle the training difficulty and improve pretraining efficiency.
- Compared with region feature based models, our method achieves similar or better performance and inferences 10 times faster.
- Compared with other raw pixel input models, our method trains 10 times faster and achieves similar or better performance.

II. Related Work

A. Vision Language Pretraining Models

Unlike traditional vision language methods [1], [11], [12], [13], [14], [15] that learn vision language mapping based on specific tasks, vision language pretraining models learn robust cross-modal mapping through large scale pretraining. Vision language pretraining models can be categorized into two types: the one-stream models and two-stream models. The one-stream models [5], [16] first embed the languages and images into embeddings. The language embeddings and image embeddings are then concatenated and processed with a shared transformer network. Since the model parameters are shared, the one-stream models usually have a small model size. However, since the self-attention models like transforms have quadratic time complexity, the one-stream models will have slightly larger memory usage. The two-stream models [17], [4], [18] process language embeddings and image embeddings with separate transformer encoders and then combine the two features with cross-attention transformer. Compared with the one-stream models, they usually have a larger model size.

In this paper, we briefly follow the network structure of two-stream models [17], [4], [18]. We change the region feature input to pixel input and change the visual feature encoder to feature extraction backbones.

B. Loss in Vision Language Pretraining

Effective loss functions lie in the heart of visual language pretraining. A region feature based vision language model usually involves three types of losses: the language loss, the visual loss, and the matching loss. Most of the time [16], [5], [17], [4], the language loss is similar to the language modeling loss as in BERT [19], in which the language tokens are masked randomly and the models are trained to predict the masked tokens based on two-directional contexts. The visual loss is to mimic the language loss and is done by masking the visual features and predicting the masked regions' classes or attributes. The matching loss predicts if the language input and the vision input are matched.

When training raw image input based vision language models, the visual inputs (image pixels) don't have specific class annotations. So existing works [8] only use the language loss and the matching loss. In this paper, we show that such training regimes lead to a slow convergence speed and low finetuning performance. With our proposed visual losses, the raw input models could achieve as fast convergence speed as region feature models in early training stages.

C. Efficient Training of Transformer Models

Transformers [20], as the most widely used language models and vision language models, although perform well in terms of accuracy, take a long training time. Training and testing transformer models efficiently is a long-standing goal in both NLP and computer vision. Existing works can be roughly divided into three categories. The first category [21], [22], [23] focuses on improving network structures. They try to optimize the quadratic time complexity of the original transformer model to linear time complexity. These works directly reduce time cost on both training and testing time. The second category [24], [25], [26] attempts to extract effective sub-networks from a trained transformer model through knowledge distillations. The third category [27], [28] focuses on improving the training strategy or proposing new losses. This type of method primarily improves efficiency during training time.

This paper falls in the third category. We are not trying to propose new network structures or extract sub-networks. We improve the training effectiveness by introducing three visual losses in vision language pretraining.

III. Effective Visual Loss for Pixel Input Vision Language Pretraining

In this section, we describe the network structure and the proposed three visual pretraining losses, namely Self-Supervised Loss (SSUL), Semantic Segmentation Loss (SEGL), and Knowledge Distillation Based Set Prediction Loss (SPL).

A. Network Structure and Other Training Losses

The general network architecture is shown in Figure 2. We briefly follow the network structure of two-stream models.

Network Structure. The image and text inputs are first processed by two separate encoders to extract high level features. The visual features and language features are then passed to the cross-modality encoder. An illustration of the cross-modality encoder structure and the cross-attention mechanism can be found in Figure 3. The cross-modality encoder is composed of cross-modality layers. In each cross-modality layer,

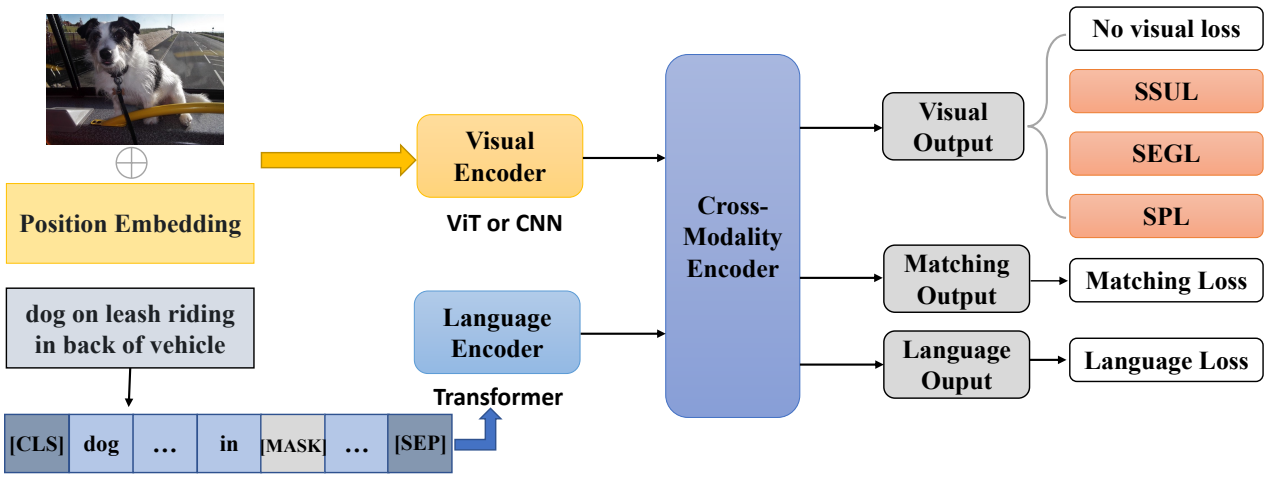

Fig. 2. The general network structure. The network takes a raw pixel image and a sentence as input. The image will pass through a visual encoder to extract features. The language will be tokenized and add a [CLS] and a [SEP] tokens. Word tokens are randomly masked with [MASK] tokens for masked language modeling loss. The visual and language features are merged with a cross-modality encoder. Similar to other methods, we use a matching loss and language modeling loss to train the network. Unlike other pixel input vision language models that use no visual loss, we apply our novel visual losses on the visual output.

the language features and image features are first merged by a cross-attention sub-layer. The cross-attention sub-layer has the same structure as a self-attention layer but uses different inputs as the query and key vectors. For the language branch, the cross-attention sub-layer uses language features from the previous layer as the query vector and image features as the key vector and value vector. After the cross-attention sub-layer, the features are further processed by a self-attention sub-layer and a fully-connected layer. We add short-cut connections between the sub-layers.

The visual input of the network is a raw pixel image. For vision transformer backbone (ViT) [10], we slice the image into patches and add an extra learnable position embedding. For CNN backbone, we first run the feature extraction then combine the extracted features with position embedding.

During pretraining, we follow the common practice to use a language modeling loss and a vision language matching loss.

Masked Language Modeling (MLM). Given an input sentence, the words are randomly masked at a ratio of 0.15. Following BERT [19], at 80% of chance, the masked token is changed to [MASK] token. Otherwise, the token is changed to zero or kept as the original token. The masked language modeling loss predicts the masked tokens given bidirectional contexts.

Image-Text Matching. Given the input sentence and image, the image-text matching loss predicts whether the text could describe the image i.g. if the image and text are matched. Fake sentences and real sentences are sampled at 50%. We use the language feature at [CLS] token to perform the matching task.

For visual loss, existing pixel input vision language pretraining models [8] can be seen as "No Visual Loss". We describe our proposed three losses in the subsection below. Also, we draw the detailed network structure for each loss in Fig 4.

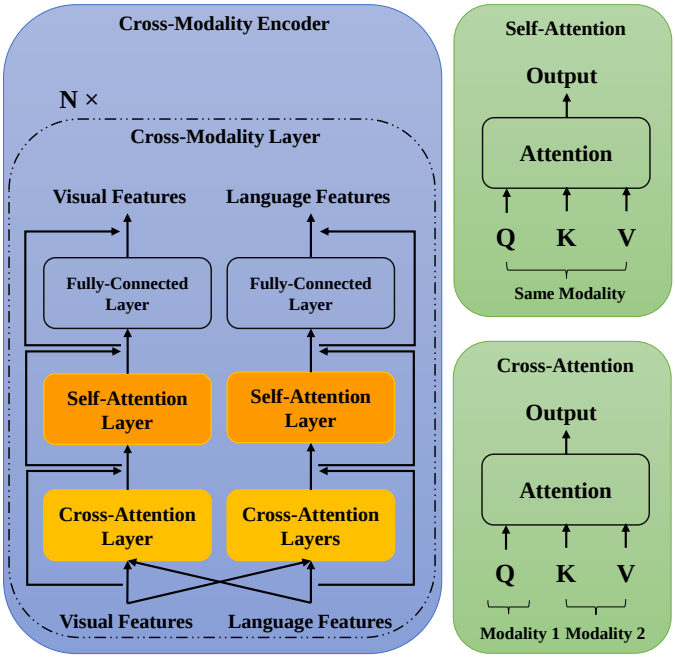

Fig. 3. Illustration of Cross-Modality Encoder structure. The cross-modality encoder is composed of a sequence of cross-modality layers. We show the detailed structure of the cross-modality layer and show the difference between self-attention and cross-attention mechanisms.

B. *Self-Supervised Loss (SSUL)*

Much of the success of BERT in NLP is replying on large scale self-supervised pretraining. Here, we propose to train the vision branch with a similar self-supervised loss. We first divide the input images into $N * N$ patches. Each patch is randomly masked by a ratio p. After that, the self-supervised loss predicts the mean color of the masked regions.

To precisely predict the mean color of masked regions, the model will need to learn to extract useful information from contexts. Note that the self-supervised loss does not require any extra image annotations, hence can be easily scaled to arbitrary size of vision language training images. Formally, we define the SSUL as:

$$SSUL = L2(Mean(\mathbf{m})|\mathbf{w}_{\setminus\mathbf{m}}, M), \tag{1}$$

where $\mathbf{m}$ is the masked areas, $\mathbf{w}_{\setminus\mathbf{m}}$ represents all other areas except for the masked area and M represents the groundtruth mean color value of the masked area.

C. Semantic Segmentation Loss (SEGL)

One of the biggest challenges in vision language pretraining is that languages and images are features of different levels. Languages usually contain high-level semantic information. For example, one language word like "car" means one car object. In contrast, image information is low-level raw information. One image pixel does not have a specific meaning. Objects and semantic information are represented by groups of pixels.

When the vision language models are trained with region feature input, the input vision features are already high-level semantic features, such that the vision and language correspondence can be learned fast. When the vision language models are trained with raw pixel input, the vision branch will be in charge of learning visual concepts.

To this end, we propose to help learning visual branch with an auxiliary Semantic Segmentation Loss (SEGL). Specifically, given vision output $X \in R^{\hat{H}\times\hat{W}}$, and groundtruth annotation $Y \in R^{H\times W}$, where W and H are image weight and height and $\hat{W}$ and $\hat{H}$ are feature map width and height, the SEGL is calculated by downsampling the annotation and calculating the cross-entropy loss between groundtruth and generated feature map:

$$SEGL = \textit{Cross-entropy}(X, downsample(Y)). \tag{2}$$

The SEGL is the same as the commonly used semantic segmentation loss. In normal semantic segmentation task, the networks usually up-sample the feature map and use semantic segmentation loss on the high-resolution feature map, which may include additional learnable parameters during up-sampling. In our task, we use SEGL as an auxiliary training target. Instead of up-sampling the feature map, we down-sample the groundtruth label map. The detailed network structure of Semantic Segmentation Loss is shown in Fig 4 C.

D. Knowledge Distillation Based Set Prediction Loss (SPL)

Although the Semantic Segmentation Loss (SEGL) could provide semantic guidance to the network, it requires additional human annotation data, which is not preferable when training scales up. To overcome this problem, we propose the Knowledge Distillation Based Set Prediction Loss (SPL). SPL first runs a pretrained object detector to get N object labels $y = \{y_i\}_{i=1}^{N}$ from the raw images. Unlike region feature based methods, the object detector is only run once before the pretraining step. The feature extraction process is not required during inference.

For the network part, given vision output $\hat{X} \in R^{\hat{N}\times\hat{M}}$, the network applies transformer decoder layers [19] to map the feature to $X \in R^{N}$, where $N << \hat{N}\times\hat{M}$. The N dimensional feature represents N predicted objects. Then the network applies a Softmax layer and generates N object predictions possibilities $\hat{p} = \{\hat{p}_i\}_{i=1}^{N}$. The detailed network structure is shown in Fig 4 D.

SPL supervises the network training by matching the set of predictions $\hat{p}$ with the set of distillated labels y. The optimal bipartite matching problem is widely studied in multi-object tracking [29], [30] and is recently also extended to object detection [31]. The calculation of SPL is done in two steps: **find optimal matching between the two sets** and then **calculate object label cross-entropy loss**.

Optimal Matching. The optimal matching between two sets is calculated by the Hungarian algorithm. Formally, given two sets $y = \{y_i\}_{i=1}^{N}$ and $\hat{p} = \{\hat{p}_i\}_{i=1}^{N}$, The Hungarian algorithm finds bipartite matching by minimizing the cost of assigning correct labels $-\hat{p}_{\sigma(i)}(c_i)$. Formally, the optimal matching $\hat{\sigma}$ is calculated by:

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{i}^{N} [-\hat{p}_{\sigma(i)}(c_i)], \tag{3}$$

where c_i is the category of object index i and σ is a permutation of the prediction $\hat{p}$ and $\hat{\sigma}$ represents the optimal matching.

Object Label Classification Loss. After the optimal bipartite matching is found, we perturb the generated labels with calculated optimal bipartite matching $\hat{\sigma}$ and continue to calculate the SPL loss as an object label cross-entropy loss between the matched prediction and distillated labels:

$$SPL = \textit{Cross-entropy}(\hat{p}_{\hat{\sigma}}, y). \tag{4}$$

To summarize and compare our proposed methods, we list extra computation cost, extra data, scalability, and inference speed of our methods vs baseline in Table I.

IV. Experiment

A. Implementation Details

1) Network Structure: We follow the network structure of 2-stream models ViLBERT [4] and LXMERT [17], remove the visual feature encoder branch, and replace it with a grid feature extractor. To prove that the training difficulty is consistent for all types of visual backbones. We do experiments for both CNN model ResNet and Vision Transformer model. For CNN model, we use ResNet50 [9]. For Vision Transformer model [10], we use a nine layers transformer model with patch size 32. For both of the backbones, we resize the input image size to 384x384, which is only 40% of the image size compared with Pixel-BERT [8]. For language input, we follow the original BERT [19] to tokenize the sentence into tokens with wordpiece tokenizer and append a [CLS] token and a [SEP] token before and after the sentence. The Cross-Modality

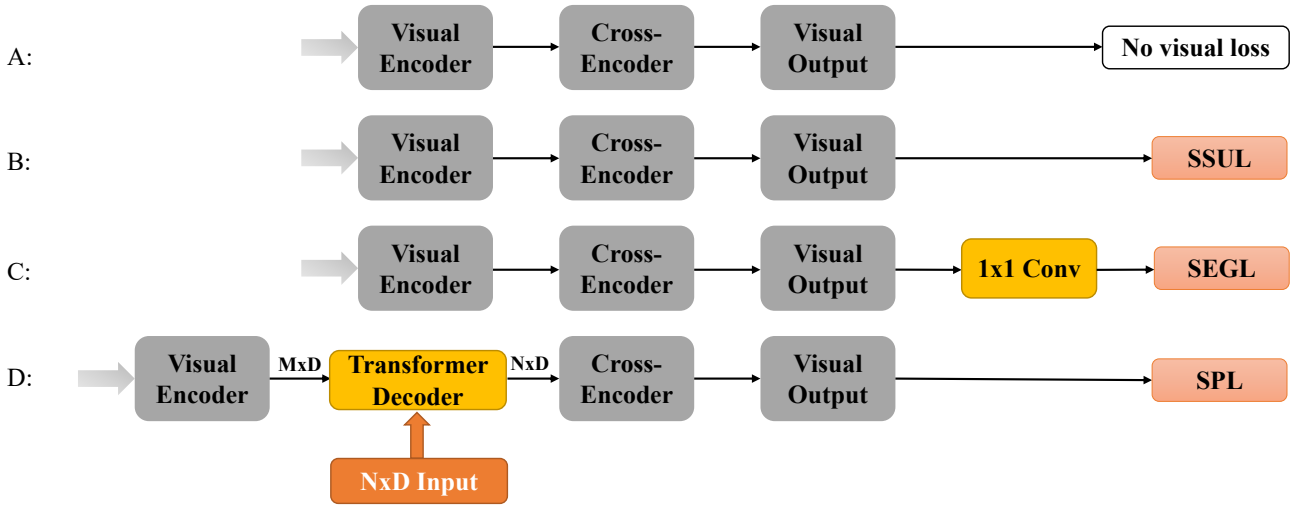

Fig. 4. The detailed network structure for our proposed methods. A: No Visual Loss. B: Self-Supervised Loss. C: Semantic Segmentation Loss adds a one-layer convolution over the baseline method to map the feature to the category dimension. D: Set Prediction Loss. Given a large dimension visual feature, the set prediction loss takes a trainable N-dimensional input and uses a transformer decoder to map the feature dimension to N. Further cross encoder will be performed on the N-dimensional features.

Method	Extra Comp Cost	Extra Data	Scalability	Inference Speed
No Visual Loss	No	No	Yes	-
SSUL	No	No	Yes	Same
SEGL	1 layer 1x1 Conv	Segmentation Mask	Partial	Same
SPL	1 layer transformer decoder	Pretrained detector	Yes	Faster

TABLE I
COMPARISON OF THE THREE PROPOSED METHODS. WE SHOW THAT THE SELF-SUPERVISED LOSS IS THE MOST GENERAL METHOD THAT REQUIRES NO EXTRA COMPUTATIONAL COST AND NO EXTRA TRAINING DATA, HENCE CAN BE EASILY SCALED TO ARBITRARY TRAINING IMAGE SIZE. SEMANTIC SEGMENTATION LOSS REQUIRES ONLY A ONE-LAYER CONVOLUTION DURING PRETRAINING. THE SET PREDICTION LOSS USES AN EXTRA ONE-LAYER TRANSFORMER DECODER TO REDUCE FEATURE DIMENSION AND IS FASTER DURING INFERENCE.

Encode contains 5 layers of cross-modality layers. We load pretrained weights for language branch and vision backbones.

2) Training Data: We only use image-sentence pairs in COCO [34], Flickr [35] and Visual Genome [36], which contains 230k images and 5 million image sentence pairs. We compare our training data with other models in Table IV. We strictly follow the data splitting process in UNITER [5] to make sure that there's no overlap between the training set and testing set. For Semantic Segmentation Loss (SEGL), we use the annotations provided in COCO Stuff [6]. The semantic segmentation groundtruth contains 172 classes of pixel labels on 100k of our training images, which covers around 50% of pretraining image data and only 10% of pretraining image-sentence pairs. For Knowledge Distillation Based Set Prediction Loss (SPL), we use the faster-rcnn model provided in Bottom-Up attention [1] to extract object labels.

3) Data Augmentation: Pixel input models also enable online data augmentation during pretraining. We propose a novel **Description aware data augmentation method**. Horizontal flip can not be applied to all images, especially those images with compositional descriptions. To solve this problem, we build an empirical dictionary with compositional keywords (e.g. left, right, etc.). If a description does not contain keywords in the dictionary, we perform horizontal flip with a chance of 50%. The data augmentation is only used in pretraining. No data augmentation is used during finetuning. **Random Resize and Crop.** We randomly resize the input image between 0.8-1.2 and crop a 384x384 patch in the resized image.

4) Implementation Details of SSUL, SEGL and SPL: **1) SSUL:** For self-supervised loss, we divide the input image size into 32x32 patches. Each patch is randomly masked at a ratio of $p = 0.15$, which is the same as language words are masked. **2) SEGL:** For semantic segmentation loss, to save GPU memory, we resize the groundtruth annotations to the same dimension as the feature map and not vice versa. For vision transformer, we resize the groundtruth annotations to 12x12. For ResNet 50, we resize the feature map to 24x24. **3) SPL:** We try to extract both 36 objects and 100 objects. Note that a typical vision transformer model produces a feature dimension of 144, so even the 100 object version requires less memory than the naive baseline models. For cross-attention modules, the network takes 36 or 100 object features as input. This design also makes sure the memory usage of cross-attention modules is consistent for whatever image input size.

For all of the proposed methods and baseline models, we

Method	Inference Time (ms)	VQA(%)
Baseline (ViT)	40	68.95
ResNet	40	68.52
Baseline + SSUL	40	70.26
Baseline + SEGL	40	**70.92**
ResNet + SEGL	40	70.53
Baseline + SPL(36)	34	69.41
Baseline + SPL(100)	37	70.20

TABLE II
ABLATION EXPERIMENTS OF OUR PROPOSED METHODS ON VQA TASKS. WE DO EXPERIMENTS ON VQAV2. OUR PROPOSED METHOD COULD IMPROVE PERFORMANCE ON BOTH BACKBONES.

Method	Inference Time (ms)	TR (COCO 5k)	IR (COCO 5k)	TR (COCO 1k)	IR (COCO 1k)
Baseline (ViT)	40	0.25	0.12	0.40	0.33
ResNet	40	0.27	0.11	0.38	0.32
Baseline + SSUL	40	0.48	0.33	0.62	0.51
Baseline + SEGL	40	**0.50**	**0.37**	**0.67**	**0.54**
ResNet + SEGL	40	**0.50**	**0.37**	0.65	0.535
Baseline + SPL(36)	34	0.42	0.25	0.55	0.44
Baseline + SPL(100)	37	0.42	0.26	0.61	0.495

TABLE III
ABLATION EXPERIMENTS OF OUR PROPOSED METHODS ON RETRIEVAL TASKS. WE REPORT THE ACCURACY AT R@1. WE DO EXPERIMENTS ON ZERO-SHOT TEXT RETRIEVAL AND ZERO-SHOT IMAGE RETRIEVAL. WE SHOW THAT ALL OF OUR PROPOSED METHODS GREATLY IMPROVE THE PERFORMANCE ON DOWN-STREAM TASKS. AMONG ALL OF THEM, THE SEMANTIC SEGMENTATION LOSS GIVES THE BEST PERFORMANCE. WE ALSO SHOW THAT THE TRAINING DIFFICULTY OF PIXEL INPUT MODELS EXISTS FOR ALL TYPES OF VISUAL BACKBONES (RESNET AND VIT).

Models	Images	Pretrain	Inference	VQAv2		NLVR2		VCR		
		GPU hrs	Time (ms)	test-dev	test-std	val	test-P	Q $\rightarrow$ A	QA $\rightarrow$ R	Q $\rightarrow$ AR
ViLBERT [4]	3m	-	$\sim$900	70.55	70.92	-	-	-	-	-
LXMERT [17]	180k	$\sim$800	$\sim$900	72.42	72.5	74.9	74.5	-	-	-
UNITER-base [5]	4.2m	882	$\sim$900	**72.70**	**72.91**	**75.85**	75.80	74.56	77.03	57.76
Pixel-BERT (R50) [8]	200k	$\sim$4000	$\sim$60	71.35	71.42	71.7	72.4	-	-	-
VilT [32]	4.1m	$\sim$4600	$\sim$15	71.26	-	75.70	**76.13**	-	-	-
Ours (SEGL)	230k	**470**	$\sim$40	**71.69**	**72.03**	72.3	72.9	**75.37**	**77.86**	**58.33**

TABLE IV
COMPARISON WITH OTHER VISION-LANGUAGE PRE-TRAINING MODELS ON VQAV2, NLVR2, AND VCR. THE PRETRAIN GPU HOURS OF PIXEL-BERT ARE APPROXIMATED BASED ON PRETRAIN IMAGE SIZE AND EPOCHS. ALL PRETRAIN GPU HOURS ARE ON V100 GPUS. INFERENCE TIME IS BASED ON P40 GPUS. WE SHOW THAT OUR METHOD COULD ACHIEVE COMPETITIVE PERFORMANCE ON THE TWO DATASETS AND RUN MORE THAN 10X FASTER COMPARED WITH OTHER REGION FEATURE MODELS. COMPARED WITH RAW PIXEL INPUT MODELS, OUR METHOD TRAINS 10 TIMES FASTER AND ACHIEVES BETTER ACCURACY.

pretrain the models for a maximum of 15 epochs and choose the checkpoint with the lowest validation loss for finetuning.

B. Finetuning Setting

In this section, we describe the finetuning settings on downstream tasks.

1) VQAv2: The task of Visual Question Answering is to answer a given question based on the content of an image. The VQA [37] task is usually formatted as a classification problem on all possible answers. During finetuning, we add a two-layer MLP on top of the language features of the [CLS] position and finetune the MLP together with the backbone models.

2) NLVR2: The Natural Language for Visual Reasoning for Real dataset [38] is designed to examine the reasoning ability of the networks. Specifically, given two images, the network will be asked to tell the statement is true or false by comparing the two images. To finetune the NLVR2 task, we use the easiest solution that processes image-language pairs separately and concatenates the [CLS] features. After concatenation, we use a binary classifier to classify if the description is true.

3) VCR: Visual Commonsense Reasoning (VCR) [39] is a dataset that asks commonsense multiple-choice questions based on an image. There are two sub-tasks: the question-answering task (Q $\rightarrow$ A) and the justification task (QA $\rightarrow$ R). The combined task is represented as Q $\rightarrow$ AR. We follow the same finetuning protocol as UNITER [5]. We use each question-answer pair as the input to the language branch and concatenate the [CLS] features of the pairs. A classifier is used to select the best answer based on the concatenated features.

Model	Images	Flickr 30k (1k image)						MSCOCO (1k image)					
		Image Retrieval			Text Retrieval			Image Retrieval			Text Retrieval		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
ViLBERT [4]	3m	31.83	61.12	72.80	-	-	-	-	-	-	-	-	-
Unicoder-VL [16]	3.8m	48.40	76.00	85.20	64.30	85.80	92.30	-	-	-	-	-	-
ImageBERT [33]	10m	54.3	79.6	87.5	70.7	90.2	94.0	53.6	83.2	91.7	67.0	90.3	96.1
UNITER-base [5]	4.2m	**66.16**	**88.40**	**92.94**	80.70	95.70	**98.00**	-	-	-	-	-	-
Ours (SEGL)	**230k**	62.46	86.71	92.80	**82.70**	**96.50**	**98.00**	**54**	**86**	**94.8**	**67.1**	**92.2**	**97.1**

TABLE V
EXPERIMENTS OF ZERO-SHOT IMAGE AND LANGUAGE RETRIEVAL ON FLICKER 30K DATASET AND COCO 1K DATASET. WE ACHIEVE COMPETITIVE RESULTS ON FLICKER AND COCO DATASETS WITH FAR FEWER TRAINING IMAGES.

Model	Images	MSCOCO (5k image)					
		Image Retrieval			Text Retrieval		
		R@1	R@5	R@10	R@1	R@5	R@10
ViLBERT [4]	3m	31.86	61.12	72.80	-	-	-
ImageBERT [33]	10m	32.3	59.0	70.2	44.0	71.2	80.4
VilT [32]	4.1m	**40.4**	**70.0**	**81.1**	**56.5**	**82.6**	**89.6**
Ours (SEGL)	**230k**	37.2	63.3	74.4	49.8	75.9	83.1

TABLE VI
ADDITIONAL EXPERIMENTS ON ZERO-SHOT IMAGE AND LANGUAGE RETRIEVAL ON COCO 5K IMAGE. WE STILL ACHIEVE COMPETITIVE RESULTS WITH FEWER TRAINING IMAGES.

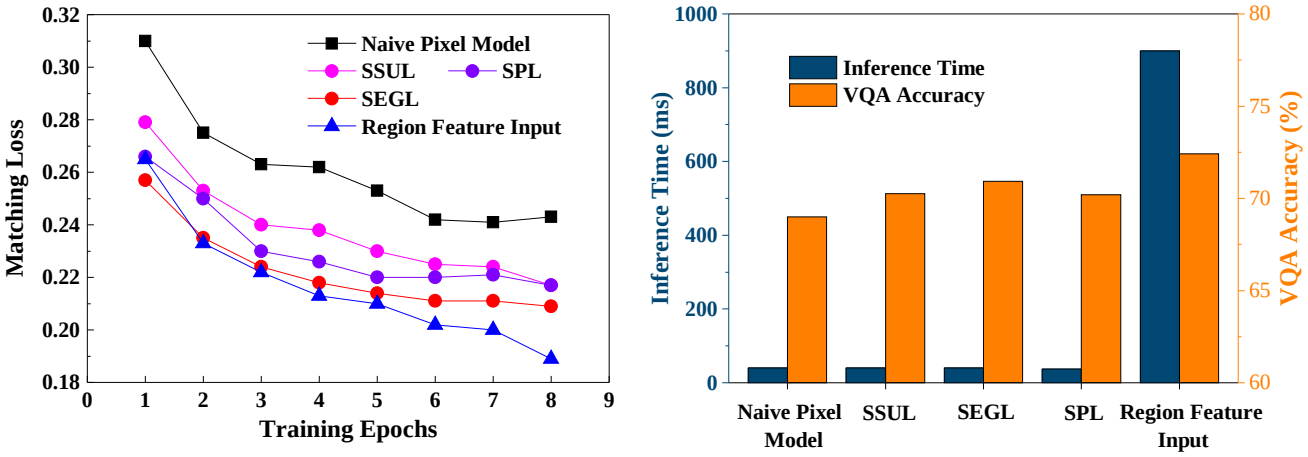

Fig. 5. Left: The convergence speed (reducing speed of matching loss) on validation set. Right: Inference time and VQA Accuracy. We show that when applying all our three losses on the baseline model, we could witness a great boost up in convergence speed. Among all of them, the SEGL gives the best convergence speed, which is almost the same as a region feature model. As for inference speed, our models achieve the same or faster inference speed as a naive baseline model and more than 10 times faster than a region feature model. Accuracy-wise, our method achieves similar performance as the region feature model and is better than the naive baseline.

4) Zero-Shot Image and Text Retrieval: The zero-shot image (text) retrieval task is to find the best matched image (text) in the database given a text (image) without extra finetuning. The retrieval technique is commonly used in modern image-language search engines. To do this, we run inference using pretrained weights on all image and language pairs in the database and rank their matching scores. The experiments are carried out on both Flickr 30k [35] and COCO dataset [34].

C. Ablation Study

We start by building a naive version of pixel input vision language pretraining model using vision transformer [10] backbone. Same as Pixel-BERT [8], the naive version uses only a matching loss and the language modeling loss. We compare our proposed three losses with the baseline version on three tasks. A detailed ablation study is shown in Table II and Table III. For all ablation experiments, we use finetuning input size 600. Following [32], the inference time is calculated by the running speed on Nvidia P40 GPU.

We show that all of our proposed methods surpass the baseline model with a large margin. Within the proposed methods, the Semantic Segmentation Loss achieves the best performance in general. It proves that additional human annotations do help to improve pretraining vision language models.

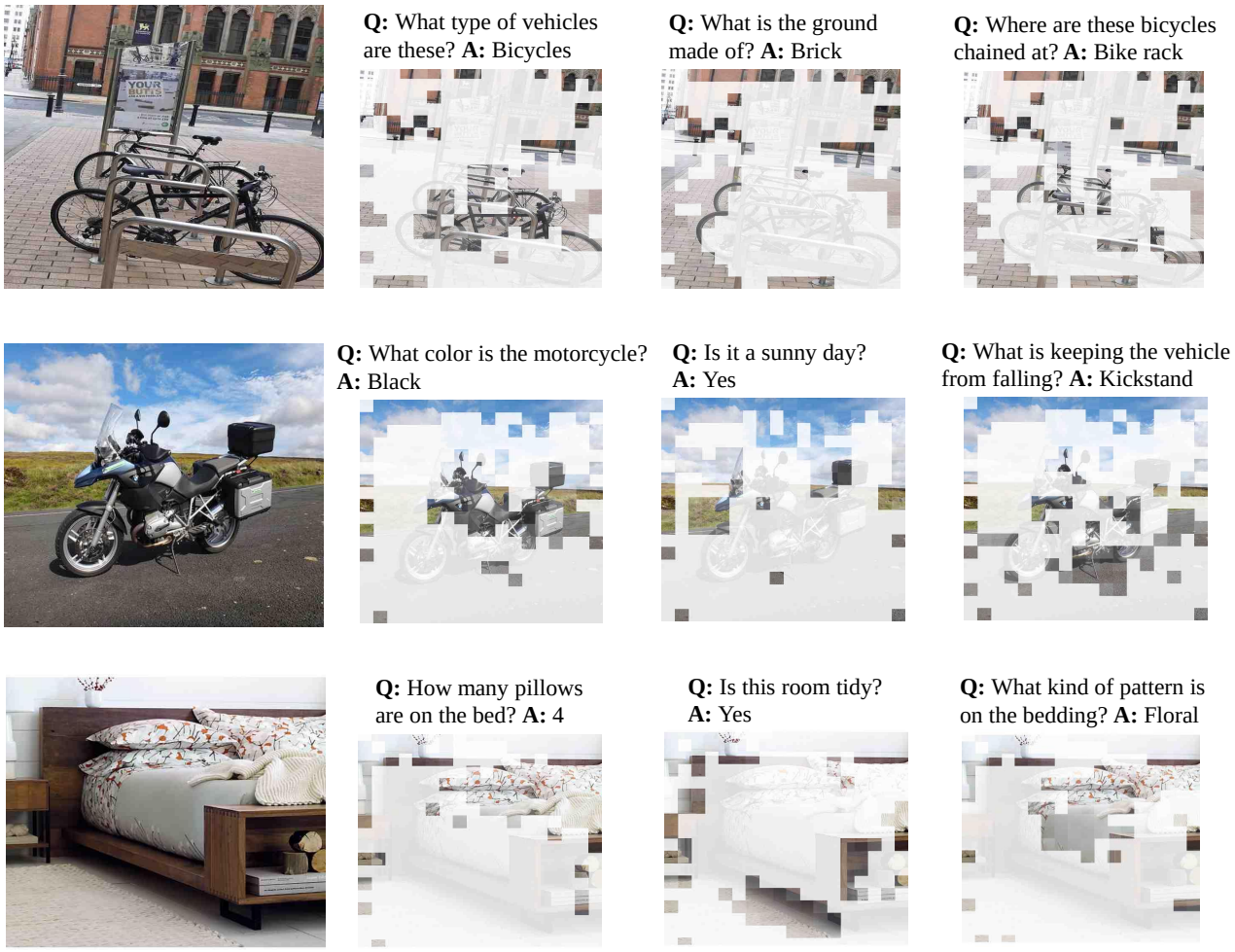

Fig. 6. Visualization of attention maps. First row: the end-to-end model could easily identify no-object items, for example, the ground. Second row: the end-to-end model could attend to small objects, for example, the kickstand. Third row: The grid feature model is weak at counting problems.

Self-supervised loss and set prediction loss also produce good results on down-stream tasks – less than 1 percent lower than semantic segmentation loss on VQA. Among all of them, the set prediction loss has a relatively faster process speed and lower memory usage. For the cross-modality module, the set prediction loss uses 36 or 100 visual features, while other vision transformer models use 324 (18*18) visual features.

By comparing ResNet [9] models and vision transformer models [10], we conclude that the pretraining difficulty exists regardless of the backbone choice and the backbones won't affect the performance of our proposed losses.

D. Comparison with Other Methods

We compare our best result (Vision Transformer + SEGL) with region feature models and pixel input models in Table IV, Table V and Table VI. For all experiments in this section, we continue to use image size 600x600 except for VQA. For VQA we use image size 800x800 to further improve performance, same as Pixel-BERT [8].

Compared with region feature models, when trained with fewer GPU hours, our method could achieve almost the same performance on various datasets. During inference, an end-to-end model could run more than 10x faster than any region feature based model.

Compared with other end-to-end models, our model is trained for only 1/10 of the pretraining GPU hours and gets competitive or even better finetuning performance.

E. Visualization

1) Visualization of Convergence Speed and Finetuning Accuracy: We plot the pretraining matching loss and finetuning performance in Fig 5. We show that our proposed methods greatly improve convergence speed during pretraining. Among all of them, the semantic segmentation loss converges at the fastest speed.

2) Visualization of Attention Map: To further prove an end-to-end model could learn coherent vision language representations, we plot the attention map of the last layer of cross-

attention modules in Fig 6. The attention map shown is the average value of all attention heads. Specifically, we use the finetuned model on VQAv2 dataset as an example. We show the attention map of one image given different questions.

First, we show that the pixel input model could attend to both objects and no-object items, for example, the ground. Answering no-object questions is difficult for a region feature based model.

In the second example, we show an example that the pixel input model could attend to small items like the kickstand. Detecting such small items is challenging for an object detector, therefore it's difficult for the region feature based models to answer questions related to small items.

Third, grid features by default don't understand numbers. When asked about counting problems, although our end-to-end model could learn to attend to the right area, it does not give a correct answer (3).

V. Conclusion and Future Works

In this paper, we have targeted the difficulty in end-to-end vision language pretraining i.g. slow convergence speed and low finetuning accuracy. We propose three losses. All of them could surpass the baseline naive model with a large margin. We show that our proposed methods could achieve similar performance when trained with the same GPU hours as region feature models. During inference time, our model runs 10 times faster than the region feature models. Compared with other pixel input vision language pretraining models, our models only require 1/10 of the pre-training GPU hours. We also evaluate different vision back-bones, e.g. the CNN backbones and Vision Transformer backbones.

Future works can be done by increasing the scale of pre-training. Our proposed self-supervised loss and set-prediction loss enable pain-free scalability. By using more training images from Conceptual Caption [40] and other datasets, a pixel input end-to-end vision language model could be further improved.

More works could also be done by exploring more complicated data augmentations. As mentioned earlier in the paper, an end-to-end vision language model enables online data augmentation. In this paper, we only use simple augmentation methods like random cropping and flip in pretraining. No data augmentation is used in finetuning. Methods like Randaugment [41] could be explored to further improve performance.

Jointly using the proposed losses could be further explored. Currently, we haven't witnessed improvements by jointly using more than one proposed losses. Future works can be done by exploring the proper weights of the proposed methods.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-003), the MOE AcRF Tier-1 research grants: RG95/20, and the OPPO research grant.

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086. 1, 2, 5

[2] D. A. Hudson and C. D. Manning, "Gqa: a new dataset for compositional question answering over real-world images," *arXiv preprint arXiv:1902.09506*, vol. 3, no. 8, 2019. 1

[3] R. Luo, B. Price, S. Cohen, and G. Shakhnarovich, "Discriminability objective for training descriptive captions," *arXiv preprint arXiv:1803.04376*, 2018. 1

[4] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Advances in Neural Information Processing Systems*, 2019, pp. 13–23. 1, 2, 4, 6, 7

[5] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Learning universal image-text representations," *arXiv preprint arXiv:1909.11740*, 2019. 1, 2, 5, 6, 7

[6] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018. 1, 5

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015. 1

[8] Z. Huang, Z. Zeng, B. Liu, D. Fu, and J. Fu, "Pixel-bert: Aligning image pixels with text by deep multi-modal transformers," *arXiv preprint arXiv:2004.00849*, 2020. 2, 3, 4, 6, 7, 8

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 2, 4, 8

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. 2, 3, 4, 7, 8

[11] Z. Zhang, Q. Wu, Y. Wang, and F. Chen, "High-quality image captioning with fine-grained and semantic-guided visual attention," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1681–1693, 2018. 2

[12] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai, "Stat: Spatial-temporal attention mechanism for video captioning," *IEEE transactions on multimedia*, vol. 22, no. 1, pp. 229–241, 2019. 2

[13] J. Hu, S. Qian, Q. Fang, and C. Xu, "Heterogeneous community question answering via social-aware multi-modal co-attention convolutional matching," *IEEE Transactions on Multimedia*, vol. 23, pp. 2321–2334, 2020. 2

[14] Z. Yuan, S. Sun, L. Duan, C. Li, X. Wu, and C. Xu, "Adversarial multimodal network for movie story question answering," *IEEE Transactions on Multimedia*, vol. 23, pp. 1744–1756, 2020. 2

[15] J. Yu, W. Zhang, Y. Lu, Z. Qin, Y. Hu, J. Tan, and Q. Wu, "Reasoning on the relation: Enhancing visual representation for visual question answering and cross-modal retrieval," *IEEE Transactions on Multimedia*, vol. 22, no. 12, pp. 3196–3209, 2020. 2

[16] G. Li, N. Duan, Y. Fang, M. Gong, D. Jiang, and M. Zhou, "Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training." in *AAAI*, 2020, pp. 11 336–11 344. 2, 7

[17] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," *arXiv preprint arXiv:1908.07490*, 2019. 2, 4, 6

[18] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee, "12-in-1: Multi-task vision and language representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 437–10 446. 2

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. 2, 3, 4

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. 2

[21] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," *arXiv preprint arXiv:2001.04451*, 2020. 2

[22] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, "Rethinking pre-training and self-training," *arXiv preprint arXiv:2006.06882*, 2020. 2

[23] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020. 2

[24] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019. 2

[25] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019. 2

[26] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for bert model compression," *arXiv preprint arXiv:1908.09355*, 2019. 2

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019. 2

[28] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," *arXiv preprint arXiv:2011.01403*, 2020. 2

[29] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *European Conference on Computer Vision*. Springer, 2008, pp. 788–801. 4

[30] B. Sahbani and W. Adiprawita, "Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system," in *2016 6th International Conference on System Engineering and Technology (ICSET)*. IEEE, 2016, pp. 109–115. 4

[31] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229. 4

[32] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5583–5594. 6, 7

[33] D. Qi, L. Su, J. Song, E. Cui, T. Bharti, and A. Sacheti, "Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data," *arXiv preprint arXiv:2001.07966*, 2020. 7

[34] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015. 5, 7

[35] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," *IJCV*, vol. 123, no. 1, pp. 74–93, 2017. 5, 7

[36] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017. 5

[37] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6904–6913. 6

[38] A. Suhr, M. Lewis, J. Yeh, and Y. Artzi, "A corpus of natural language for visual reasoning," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 217–223. 6

[39] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From recognition to cognition: Visual commonsense reasoning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6720–6731. 6

[40] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2556–2565. 9

[41] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703. 9