HOW MUCH CAN CLIP BENEFIT VISION-AND-LANGUAGE TASKS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Most existing Vision-and-Language (V&L) models rely on pre-trained visual encoders, using a relatively small set of manually-annotated data (as compared to web-crawled data), to perceive the visual world. However, it has been observed that large-scale pre-training usually can result in better generalization performance, e.g., CLIP (Contrastive Language-Image Pre-training), trained on a massive amount of image-caption pairs, has shown a strong zero-shot capability on various vision tasks. To further study the advantage brought by CLIP, we propose to use CLIP as the visual encoder in various V&L models in two typical scenarios: 1) plugging CLIP into task-specific fine-tuning; 2) combining CLIP with V&L pre-training and transferring to downstream tasks. We show that CLIP significantly outperforms widely-used visual encoders trained with in-domain annotated data, such as BottomUp-TopDown. We achieve competitive or better results on diverse V&L tasks, while establishing new state-of-the-art results on Visual Question Answering, Visual Entailment, and V&L Navigation tasks.

1 INTRODUCTION

Vision-and-Language (V&L) tasks such as VQA (Antol et al., 2015) test a system's ability to understand and reason about the semantics of the visual world with the help of natural language. Most V&L models rely on a *visual encoder* to perceive the visual world, which translates the raw pixels into vectors from a representation space. Recent work (Anderson et al., 2018a; Jiang et al., 2020; Zhang et al., 2021) observes that the visual representation has become the performance bottleneck of V&L models and stress the importance of learning a powerful visual encoder. These high-performing visual encoders are trained on manually-annotated data with class labels (e.g., ImageNet) (Russakovsky et al., 2015) or bounding boxes (e.g., Visual Genome) (Krishna et al., 2017). However, such detection or image classification data is costly to collect, and the visual representation is limited by the predefined class labels. Thus, there is a need for a visual encoder that is trained on more diverse and large-scale data sources, unbounded by a fixed set of labels, and with generalization ability to unseen objects and concepts.

Recently, CLIP (Radford et al., 2021) has been proposed to learn visual concepts with language supervision. CLIP consists of a visual encoder and a text encoder. It is trained on 400M noisy image-text pairs crawled from the Internet. Thus, the data collection process is scalable and requires little human annotation. CLIP has shown strong *zero-shot* capabilities on benchmarks such as ImageNet classification. We hypothesize that it also bears great potential for the V&L tasks. However, directly applying CLIP as a zero-shot model to V&L tasks proves to be difficult (Section 5 and Kim et al. (2021)), as many V&L tasks require complex multi-modal reasoning. Thus, we propose to integrate CLIP with existing V&L models by replacing their *visual encoder* with CLIP's visual encoder.[1]

To the best of our knowledge, we present the first large-scale empirical study on using CLIP as the visual encoder for diverse V&L tasks. We consider two typical scenarios: 1) we use CLIP in direct task-specific fine-tuning (Section 3); 2) we integrate CLIP with V&L pre-training on image-text pairs

[1]Without confusion, we use the term CLIP to interchangeably refer to both the whole CLIP model (including the text and visual encoder) and just its visual encoder.

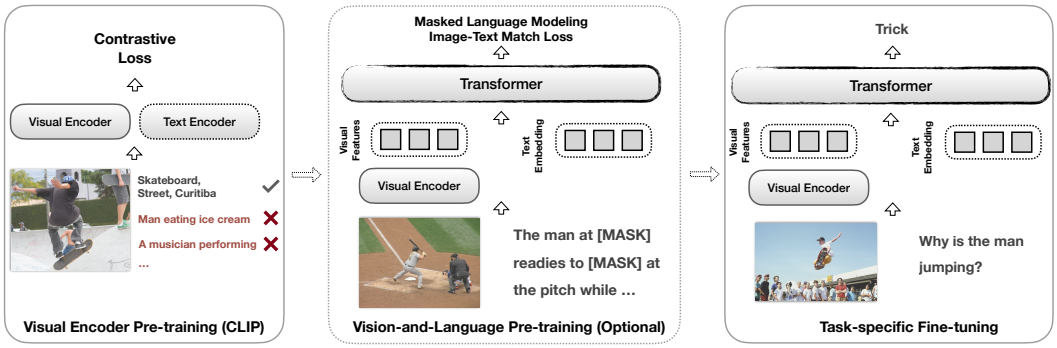


Figure 1: The training process of a V&L model typically consists of three steps: 1) visual encoder pre-training, 2) vision-and-language pre-training (optional), and 3) task-specific fine-tuning. In previous V&L models, visual encoder pre-training requires human annotated vision datasets, which is hard to scale up. Our CLIP-ViL proposes to use CLIP, which is trained on image-text pairs crawled from the Internet, as the visual encoder for V&L models. This reduces the need for human annotated in the pipeline and greatly improves model performance.

and transfer to downstream tasks (Section 4).[2] For clarity, we denote the models used in these two scenarios as **CLIP-ViL** (without V&L pre-training) and **CLIP-ViL$_{\mathbf{p}}$** (with V&L pre-training).

In *direct task-specific fine-tuning*, we consider three widely-adopted tasks: Visual Question Answering (Antol et al., 2015), Image Captioning (Chen et al., 2015), and Vision-and-Language Navigation (Anderson et al., 2018b). On all three tasks, CLIP-ViL brings sizable improvement over strong baselines, 1.4% accuracy on VQA v2.0, 6.5 CIDEr on COCO Captioning, and 4.0% success rate on Room-to-Room navigation.

In *V&L pre-training*, we replace the conventionally used region-based representation (Anderson et al., 2018a) with CLIP. CLIP-ViL$_{\text{p}}$ performs exceptionally well on three benchmarks, including VQA v2.0, SNLI-VE (Xie et al., 2019), and GQA (Hudson and Manning, 2019), setting a new state-of-the-art (SotA) on VQA (76.70% on test-std), and SNLI-VE (80.20% on test). CLIP-ViL$_{\text{p}}$ with CLIP-Res50 outperforms models based on the widely used region-based encoder, BottomUp-TopDown (BUTD) ResNet101 (Anderson et al., 2018a). Moreover, CLIP-ViL$_{\text{p}}$ with CLIP-Res50x4 surpasses VinVL-ResNeXt152 (Zhang et al., 2021), which is an extreme scale-up attempt of the region-based encoder with human-annotated data.

2 Background and Motivation

Vision-and-Language (V&L) models. V&L tasks require a model to understand the visual world and to ground natural language to the visual observations. Prominent tasks include visual question answering (Antol et al., 2015), image captioning (Chen et al., 2015), vision-language navigation (Anderson et al., 2018a), image-text retrieval (Wang et al., 2016) and so on. V&L models designed for these tasks often consist of a visual encoder, a text encoder, and a cross-modal interaction module (Kim et al., 2021).

We illustrate the three typical training stages in Figure 1: 1) the visual encoder is trained on annotated vision datasets (Russakovsky et al., 2015; Krishna et al., 2017) (denoted as *visual encoder pre-training*); 2) (optionally) pre-training on paired image-caption data with a reconstructive objective and an image-text matching objective (denoted as *vision-and-language pre-training*) (Lu et al., 2019); 3) fine-tuning on task-specific data (denoted as *task-specific fine-tuning*).

Visual encoders in V&L models. Different models employ different visual encoders, we illustrate their architectures and pre-training processes in Figure 2. The encoders can be categorized as follows: 1) *region-based* models such as BUTD object detector (Anderson et al., 2018a; Kamath et al., 2021);

[2]We distinguish between *V&L pre-training* and *CLIP pre-training*: V&L pre-training models (Lu et al., 2019) have deep interactions between modalities while CLIP follows a shallow-interaction design (Section 2).

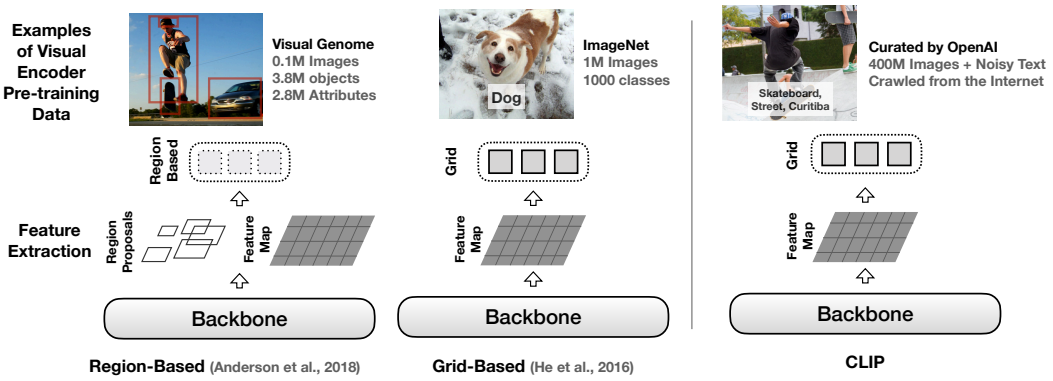


Figure 2: CLIP versus other visual encoders. Region-based methods (Anderson et al., 2018a) are trained on object detection data. For grid-based methods, previous work use either image classification (He et al., 2016) or detection data (Jiang et al., 2020). However, CLIP requires only aligned text.

2) *grid-based* models such as Jiang et al. (2020) that directly extract grid-like feature maps from the visual backbone (He et al., 2016; Dosovitskiy et al., 2020).

The encoder is first pre-trained on human-annotated vision datasets. Region-based encoders are pre-trained with detection data such as Visual Genome (Krishna et al., 2017). Grid-based encoders are pre-trained with image classification data such as ImageNet (Russakovsky et al., 2015) or detection data (Jiang et al., 2020). However, these manually labeled datasets are expensive to construct and hard to scale up. They only provide supervision for a limited number of predetermined visual concepts. This motivates us to use CLIP as the visual encoder.

CLIP. CLIP (Contrastive Language-Image Pre-training) (Radford et al., 2021)[3] falls into the line of research that learns visual representations from natural language supervision (Desai and Johnson, 2020; Sariyildiz et al., 2020; Jia et al., 2021). CLIP follows a "shallow-interaction design", where a visual encoder and a text encoder encode an input image and text independently, and the dot-product between the two encoder's output is used as the similarity score between the input image and text. It is pre-trained with a contrastive loss where the model needs to distinguish aligned pairs from randomly sampled pairs. CLIP leverages an abundantly available source of supervision without human annotation: 400M image-text pairs found across the Internet. As a result, CLIP achieves SotA performance in a range of image classification and image-text retrieval tasks in a zero-shot setting.

2.1 Motivation

Despite the strong zero-shot capability of CLIP on vision tasks, CLIP does not exhibit the same level of performance on certain V&L downstream tasks. For instance, if we cast VQA 2.0 (Goyal et al., 2017) into a zero-shot image-to-text retrieval task, we only observe chance performance (Section 5). Thus, we propose to integrate CLIP's visual encoder with previous V&L models (Figure 1). We consider the following CLIP variants with different visual backbones (He et al., 2016; Dosovitskiy et al., 2020) (CLIP-ResNet denoted as CLIP-Res): CLIP-Res50, CLIP-Res101, CLIP-Res50x4, and CLIP-ViT-B. We next describe our methods in two scenarios: 1) direct task-specific fine-tuning (Section 3) and 2) V&L pre-training (Section 4).

3 CLIP-ViL

In this section, we directly use CLIP as the visual encoder in task-specific models (referred as CLIP-ViL) and fine-tune on three representative tasks including Visual Question Answering (Section 3.1), Image Captioning (Section 3.2), and Vision-Language Navigation (Section 3.3).

[3] https://github.com/openai/CLIP

3.1 VISUAL QUESTION ANSWERING

The task of Visual Question Answering (VQA) (Antol et al., 2015) is to provide the answer given an image and a related question. Various methods have been introduced (Fukui et al., 2016; Yang et al., 2016; Anderson et al., 2018a; Jiang et al., 2018; Gao et al., 2019; Jiang et al., 2020). Here, we select two representative approaches (i.e., Pythia (Jiang et al., 2018) and MCAN (Yu et al., 2019)) to study the impact of the CLIP visual encoders in VQA.

Table 1: Results on VQA v2.0. "VG" denotes that the visual encoder has been further pre-trained on Visual Genome detection. "*" marks results from (Jiang et al., 2020). Subscription "R" denotes the region features, while other methods use grid features.

VQA Model	Visual Encoder	Result	
		Test-dev	Test-std
Pythia	ImageNet-Res50*	63.21	-
	ImageNet-Res50	61.54	61.76
	CLIP-ViT-B	51.38	51.47
	CLIP-Res50	65.55	65.78
	CLIP-Res101	66.76	67.14
	CLIP-Res50x4	**68.37**	**68.68**
MCAN	ImageNet-ResNet50	67.23	67.46
	CLIP-ViT-B	56.18	56.28
	CLIP-Res50	71.49	71.72
	CLIP-Res101	72.77	73.19
	CLIP-Res50x4	**74.01**	**74.17**
$\text{Pythia}_{\text{VG}}$	ImageNet-Res50 *	66.27	-
	ImageNet-ResNeXt-101_{R}*	**68.21**	-
	ImageNet-ResNeXt-101*	67.76	-
	ImageNet-Res50	64.36	64.81
	CLIP-Res50	60.01	60.06
MCAN_{VG}	ImageNet-ResNeXt-101_{R}*	72.01	-
	ImageNet-ResNeXt-101*	**72.59**	-
	ImageNet-Res50	70.13	70.28
	CLIP-Res50	67.41	67.52

Experimental Setup. We evaluate on VQA v2.0 (Goyal et al., 2017) and follow Jiang et al. (2020)[4] for grid feature extraction. Details of Pythia and MCAN as well as full implementation details are included in the Appendix.

Experimental Results. We report results on the VQA v2.0 Test-dev / Test-std set in Table 1. Compared to the visual feature extractors pre-trained on ImageNet classification task, CLIP visual modules demonstrate clear improvement (first two blocks of Table 1). CLIP-Res50 achieves 65.55% with Pythia (4.01% better than ImageNet-Res50) and 71.49% with MCAN (4.26% better than ImageNet-Res50) on Test-dev. With larger models (i.e., CLIP-Res101 and CLIP-Res50x4), the results continue improving and the largest CLIP-ViL CLIP-Res50x4 outperforms ImageNet-ResNeXt-101_{R} (+0.16) with Pythia. CLIP-ViL CLIP-Res50x4 also achieves the best performance of 74.01% on Test-dev and 74.17% on Test-std with MCAN.

We also show results after further detection pre-training on VG (Jiang et al., 2020). We mark these results as $\text{Pythia}_{\text{VG}}$ and MCAN_{VG} in the last two blocks of Table 1. With ImageNet-Res50 encoder, it helps boost the performance by 2.82% on $\text{Pythia}_{\text{VG}}$ (61.54% vs. 64.36%) and 2.90% on MCAN_{VG} (67.23% vs. 70.13%). However, the performance drops dramatically for CLIP-Res50 by 5.54% with $\text{Pythia}_{\text{VG}}$ and 4.08% with MCAN_{VG}. The potential reason is that CLIP-Res50 is trained on different data and with a different method compared to ImageNet counterparts, so following the previous Visual-Genome fine-tuning practice designed for ImageNet models may hurt. We also notice that our best-performing CLIP-ViL CLIP-Res50x4 MCAN (74.01%) still surpasses the best ImageNet-ResNeXt-101 MCAN_{VG} (72.59%) on Test-dev.

3.2 IMAGE CAPTIONING

Image captioning aims at generating a natural language description of an image. Various methods have been delivered for image captioning (Karpathy and Fei-Fei, 2015; Rennie et al., 2017; Anderson et al., 2018a; Luo et al., 2018; Luo, 2020). We investigate the effectiveness of the CLIP model for this popular task with (Luo, 2020) method.

Experimental Setup. For the model architecture, we experiment with the basic Transformer model adapted from Vaswani et al. (2017) in Luo (2020). Grid feature maps are extracted for each image. We evaluate our model on COCO dataset (Chen et al., 2015). We use the standard automatic evaluation metrics including CIDEr (Anderson et al., 2016), BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and SPICE (Anderson et al., 2016). The scores are obtained on Karparthy test split (Karpathy and Fei-Fei, 2015) with beam search of 5 beams. Details are given in Appendix.

[4]https://github.com/facebookresearch/grid-feats-vqa

Table 2: Image Captioning results. B@4, M, C, and S are BLUE-4, METEOR, CIDEr and SPICE metric, respectively. "*" marks results from Luo (2020). CLIP-Res models outperform ImageNet pre-trained alternatives for both ResNet50 and ResNet101, as well as the strong in-domain region-based features from BUTD.

Model	B@4	M	C	S
BUTD (Anderson et al., 2018a)	36.3	27.7	120.1	21.4
VLP (Zhou et al., 2020)	39.5	29.3	129.8	22.4
AoANet (Huang et al., 2019b)	38.9	29.2	129.8	22.4
$\text{Oscar}_{\text{base}}$ (Li et al., 2020)	40.5	29.7	137.6	22.8
$\text{VinVL}_{\text{base}}$ (Zhang et al., 2021)	**40.9**	**30.9**	**140.4**	**25.1**
$\text{BUTD}_{\text{Transformer}}$* (Luo, 2020)	-	-	**127.7**	**22.5**
$\text{ImageNet-Res50}_{\text{Transformer}}$	36.2	27.6	118.8	21.2
$\text{BiT}_{\text{M}}\text{-Res50}_{\text{Transformer}}$	37.4	28.1	122.7	22.1
$\text{BiT}_{\text{M}}\text{-Res101}_{\text{Transformer}}$	37.4	28.1	122.7	22.1
$\text{CLIP-Res50}_{\text{Transformer}}$	38.6	28.8	127.9	22.7
$\text{CLIP-Res101}_{\text{Transformer}}$	39.2	29.1	130.3	23.0
$\text{CLIP-Res50x4}_{\text{Transformer}}$	**40.2**	**29.7**	**134.2**	**23.8**
$\text{CLIP-ViT-B}_{\text{Transformer}}$	21.1	19.4	58.0	12.2

Table 3: Unseen test results for Room-to-Room (R2R) dataset. 'SR' and 'SPL' are Success Rate and Success rate normalized by Path Length. 'Pre-Training' methods are mostly in-domain pre-trained on the Matterport3D (Chang et al., 2017) environments.

Method	Unseen Test	
	SR	SPL
No Pre-Training		
R2R (Anderson et al., 2018b)	20	18
RPA (Wang et al., 2018)	25	23
S-Follower (Fried et al., 2018)	35	28
RCM (Wang et al., 2019)	43	38
SMNA (Ma et al., 2019a)	48	35
Regretful (Ma et al., 2019b)	48	40
FAST-Short (Ke et al., 2019)	54	41
EnvDrop (Tan et al., 2019)	51	47
PRESS (Li et al., 2019b)	49	45
ALTR (Huang et al., 2019a)	48	45
CG (Anderson et al., 2019)	33	30
RelGraph (Hong et al., 2020)	55	52
EnvDrop + CLIP-ViL	**59**	**53**
Pre-Training		
AuxRN (Zhu et al., 2020)	55	50
PREVALENT (Hao et al., 2020)	54	51
VLN-BERT(Hong et al., 2021)+OSCAR	57	53
VLN-BERT(Hong et al., 2021)	63	57

Experimental Results. We report Image Captioning results with different models in Table 2. Using the Transformer architecture from (Luo, 2020), we see that CLIP-Res models outperform ImageNet pre-trained alternatives for both ResNet50 (+9.1 / +1.5 in CIDEr / SPICE) and ResNet101 (+9.2 / +1.5 in CIDEr / SPICE). It even surpasses the strong in-domain region-based feature from BUTD. As the model size grows in CLIP-ViL, the results also improve and the largest CLIP-Res50x4 achieves the best performance, although there still remains a gap to the pre-trained models that have interactive image-text pre-training phase like $\text{Oscar}_{\text{base}}$ and $\text{VinVL}_{\text{base}}$. Again, CLIP-ViT-B variant leads to dramatically worse performance compared to other visual modules, that we will discuss in Section 5.

3.3 Vision-and-Language Navigation

Vision-and-language navigation tests the agent's ability to take action according to human instructions, which recently gains popularity in embodied AI (Anderson et al., 2018b; Chen et al., 2019; Jain et al., 2019; Chen et al., 2019; Qi et al., 2020b; Krantz et al., 2020; Nguyen and Daumé III, 2019; Ku et al., 2020). Specifically, the agent is put at a location in the environment (Chang et al., 2017) and asked to reach a target by following the language instructions. Here, we investigate the impact of the CLIP visual encoder on this new task.

Model Architecture. We experiment with the basic attentive neural agent as in Fried et al. (2018) (please refer to the original paper for implementation details). At each time step, the agent attends to the panoramic views and the instruction to make an action. We replace the pre-trained visual encoder from ImageNet pre-trained ResNet to the pre-trained CLIP visual encoders. Different from the VQA task that uses a feature map to include detailed information, we use a single-vector output for the entire image following previous works (Fried et al., 2018). For CLIP-ViT-B models, we take the output of the [CLS] token. For CLIP-ResNet models, we take the attentive pooled feature (Radford et al., 2021) of the feature map. These features are also linearly projected and L2-normalized as in the CLIP model.

Experimental Setup. We apply our model to two vision-and-language navigation datasets: Room-to-Room (R2R, Anderson et al. (2018b)) and Room-across-Room (RxR, Ku et al. (2020)). R2R is built on the indoor environments from the MatterPort3D dataset (Chang et al., 2017). The environments are split into training, unseen validation, and unseen test. RxR extends the R2R dataset to multiple

Table 4: Results of Room-to-Room (R2R) and Room-across-Room (RxR) datasets with original ResNet features and CLIP feature variants. 'BT-Agent' is the agent trained with back translation (BT). 'SR' is Success Rate. 'SPL' and 'nDTW' are the main metrics for R2R and RxR, respectively. The best results are bold. CLIP-ViL shows clear improvements over the previous ImageNet-trained ResNet model.

Features	Room-to-Room				Room-across-Room							
	Agent		BT-Agent		English		Hindi		Telugu		Average	
	SR	*SPL*	SR	*SPL*	SR	*nDTW*	SR	*nDTW*	SR	*nDTW*	SR	*nDTW*
ImageNet-Res152	48.2	44.4	53.5	48.8	35.3	50.6	37.9	51.9	37.1	52.0	36.8	51.5
CLIP-Res50	52.6	47.4	56.2	49.7	38.8	53.3	44.1	55.7	43.5	55.5	42.1	54.8
CLIP-ViT-B	52.5	47.7	57.4	51.3	40.2	52.5	44.3	55.0	42.1	54.6	42.2	54.0
CLIP-Res101	53.6	47.5	56.7	49.5	**41.0**	54.6	**44.9**	**56.9**	42.2	55.3	**42.7**	55.6
CLIP-Res50x4	**54.7**	**48.7**	**59.2**	**52.9**	40.8	**54.7**	44.5	56.5	**42.4**	**56.0**	42.6	**55.7**

languages and follows the environment split. For R2R dataset, we follow the hyperparameter of the publicly available implementation[5] R2R-EnvDrop (Tan et al., 2019) and replace the input features[6] with the CLIP features. For RxR dataset, we change the path length and instruction length; details are given in Appendix.

Table 5: Unseen test results for Room-across-Room (RxR) dataset under mono-lingual setup. 'SR' and 'nDTW' are Success Rate and normalized Dynamic Time Warping.

Method	**Unseen Test**	
	SR	**nDTW**
Random-Baseline (Ku et al., 2020)	7.5	15.4
Mono-Baseline (Ku et al., 2020)	25.4	41.1
SAA (Li et al., 2021a)	35.4	46.8
EnvDrop + CLIP-ViL	**38.3**	**51.1**

Experimental Results. We show the test-unseen results of our best model (CLIP-Res50x4) and the comparison to the previous methods. On R2R dataset (in Table 3), CLIP-ViL reaches 8% higher in SR (success rate) and 6% higher in SPL (Success Rate normalized by Path Length) than our baseline, EnvDrop. CLIP-ViL outperforms previous non-pre-training agents and shows competitive results to VLN-specific pre-trained models. On RxR dataset (Table 5), CLIP-ViL achieves the best success rate and nDTW (normalized Dynamic Time Warping) under the mono-lingual setup (Ku et al., 2020) and is 4.3% better then the previous results for nDTW.

In Table 4, we compare different CLIP variants with the previous standard ResNet-152 feature extractors. These extractors are pre-trained on ImageNet and use the mean-pooled features as the representation for the image. CLIP-Res50 shows a clear improvement over the IN alternative ('ImageNet-Res152'). With larger models (i.e., 'CLIP-Res101' and 'CLIP-Res50x4'), the agent performance scales well on both R2R and RxR. Lastly, we find that the CLIP ViT model ('CLIP-ViT-B') has similar results as CLIP-Res50 model. ViT also shows a relatively better result when back translation (BT) is applied. The success of ViT model in VLN is possibly due to the use of [CLS] feature instead of the feature map.

4 Vision-and-Language Pre-training

Recently, V&L pre-training has been proposed as an effective technique to improve the performance on various V&L tasks (Lu et al., 2019; Tan and Bansal, 2019; Li et al., 2019a; Su et al., 2019; Chen et al., 2020; Zhou et al., 2020; Huang et al., 2020; Li et al., 2020; Zhang et al., 2021; Li et al., 2021b). Before task-specific fine-tuning, the model is pre-trained on aligned image-text data with a reconstructive objective and an image-text matching objective. We seek to test the potential of combining CLIP pre-training and V&L pre-training. We introduce CLiP-ViL$_{\text{p}}$, a vision-and-language model pre-trained on image-text data with CLIP visual encoder as its visual backbone. In the following, we introduce the model architecture and pre-training process of CLiP-ViL$_{\text{p}}$ in detail.

[5] https://github.com/airsplay/R2R-EnvDrop

[6] https://github.com/peteanderson80/Matterport3DSimulator

Table 6: Evaluation results on three vision-and-language tasks. Our model with CLIP-Res50 outperforms most BUTD-based models. Our model with CLIP-Res50x4 sets a new state-of-the-art on VQA and SNLI-VE. It surpasses VinVL, which is a scaled-up version of BUTD and undergoes more intensive V&L pre-training than ours.

Model	VisualEncoder	V&L Pretrain		VQA		SNLI-VE		GQA	
		Data	Epoch	Test-Dev	Test-Std	Dev	Test-P	Test-Dev	Test-Std
PixelBERT	ImageNet-Res50	5.5M	40	71.35	71.42	-	-	-	-
PixelBERT	ImageNet-ResX152	5.5M	40	74.45	74.55	-	-	-	-
LXMERT	BUTD-Res101	9.2M	20	72.42	72.54	-	-	60.00	60.30
UNITER	BUTD-Res101	6.5M	-	72.70	72.91	78.59	78.28	-	-
Oscar	BUTD-Res101	6.5M	118	73.16	73.44	-	-	61.19	61.23
VinVL	VinVL-ResX152	8.9M	116	75.95	76.12	-	-	**65.05**	**65.65**
CLiP-ViL$_{\mathrm{p}}$	CLIP-Res50	9.2M	20	73.92	74.09	78.64	78.97	59.79	60.55
	CLIP-Res50x4	9.2M	20	**76.48**	**76.70**	**80.61**	**80.20**	61.42	62.93

4.1 CLiP-ViL$_{\mathrm{p}}$

Model Architecture. CLiP-ViL$_{\mathrm{p}}$ assumes a text segment T and an image I as input. As in BERT, the text is tokenized into a sequence of subwords $\{w_1, w_2, ..., w_k\}$. Every subword is embedded as the sum of its token, position, and segment embeddings (Devlin et al., 2019) and thus the text is embedded as a sequence of word embeddings $\{\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_n}\}$. The image is embedded as a set of visual vectors $\{\boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_m}\}$ from the grid-like feature map. The text and visual input are then concatanated into a sequence, $\{\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_n}, \boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_m}\}$, and processed by a single Transformer. In most region-based models, the visual backbone is frozen as fine-tuning the object detector along with the Transformer remains an open problem (Su et al., 2019). In CLiP-ViL$_{\mathrm{p}}$, the CLIP backbone is trained during both V&L pre-training and task-specific fine-tuning (see discussion in Section 5).

Pre-training on Image-Text Data. To learn unified representations for both vision and language, we follow prior work and pre-train the model on image-text pairs. We consider three pre-training objectives from LXMERT (Tan and Bansal, 2019): 1) grounded masked language modeling, where we randomly mask out 15% of words in the input sentence and train the model to reconstruct the masked words; 2) text-image matching, where the model is provided with a mismatched sentence with a probability of 0.5, and is trained to classify whether the text corresponds to the image; 3) visual question answering, where we train the model to predict the correct answer given a question.

4.2 Experiments

Setup. We experiment with two variants of CLIP as the visual encoder, CLIP-Res50 and CLIP-Res50x4. Following LXMERT, we use the same corpora aggregated from MS COCO Captions (Chen et al., 2015), Visual Genome Captions (Krishna et al., 2017), VQA (Antol et al., 2015), GQA (Hudson and Manning, 2019), and VG-QA (Zhu et al., 2016) for pre-training. We follow the same pre-processing procedure and exclude any test data from the pre-training dataset. This results in 9.18M image-text pairs.

For computational efficiency, we use a relatively small resolution for images. We resize the shorter edges of images to 384 and the longer edges to under 640 with preserved aspect ratios. During pre-training, as the number of image patches is large, we randomly sample 100 image patches for every image following PixelBERT (Huang et al., 2020). We pre-train the model for 20 epochs and unfreeze the CLIP backbone during pre-training and fine-tuning. For details see the Appendix.

Tasks. For evaluation, we fine-tune the pre-trained model on three V&L tasks: VQA v2.0 (Goyal et al., 2017), visual entailment SNLI-VE (Xie et al., 2019), and GQA (Hudson and Manning, 2019). We provide more details in the Appendix.

Results. We report the results in Table 6. We include previous best pre-trained V&L models and their V&L pre-training data and epochs. As our model is based on BERT$_{\mathrm{BASE}}$, we compare only with models based on BERT$_{\mathrm{BASE}}$. The models are grouped by their visual encoder type. We first note that

Table 7: Zero-shot performance of CLIP on VQA v2.0 `mini-eval`, "PE" denotes we follow similar prompt engineering as suggested in CLIP paper.

Model	VQA Question Type		
	yes/no	number	other
CLIP-Res50	0.037	0.057	0.0
CLIP-ViT-B_{PE}	0.019	0.0	0.0
CLIP-Res50$_{PE}$	0.055	0.057	0.0
CLIP-Res101$_{PE}$	0.260	0.0	0.0
CLIP-Res50x4$_{PE}$	0.446	0.118	0.034

Table 8: The importance of unfreezing the visual backbone (evaluated on VQA test-dev). CLIP models (unfrozen) get more improvement from V&L pre-training than BUTD (frozen).

Feature	No Pre-train	Pre-train	Diff
CLIP-Res50	64.66	73.92	+9.26
CLIP-Res50x4	69.91	76.48	+6.57
BUTD-Res101	66.70	72.42	+5.72

our two models perform competitively on all metrics. Especially, CLIP-ViL with CLIP-Res50x4 establishes a new SotA on VQA and SNLI-VE.

When comparing with the BUTD visual encoder trained on *in-domain data* (including LXMERT (Tan and Bansal, 2019), UNITER (Chen et al., 2020), and Oscar (Li et al., 2020)), our two models (CLIP-ViL with CLIP-Res50 and CLIP-Res50x4) significantly outperform most BUTD-Res101 based models. We especially note that LXMERT is trained on the same pre-training dataset and for the same number of epochs as our model, yet our CLiP-ViL$_p$ with CLIP-Res50 outperforms LXMERT on VQA by 2.59.

VinVL (Li et al., 2020) is an extreme scale-up of the region-based paradigm, which is pre-trained on multiple object detection datasets, including MS COCO (Lin et al., 2014), OpenImages (Kuznetsova et al., 2020), Object365 (Shao et al., 2019), and Visual Genome (Krishna et al., 2017). Yet, our model with CLIP-Res50x4 outperforms VinVL on VQA, while requiring significantly less steps of V&L pre-training. On GQA, our model under-performs VinVL. The potential reason is that GQA is automatically constructed from object bounding box data, which may give region-based models trained on such object data a significant advantage.

Lastly, we compare to Pixel-BERT (Huang et al., 2020), which takes a similar design as our model, but with an ImageNet initialized ResNet. CLIP initialization clearly holds advantage over ImageNet initialization, as CLIP-Res50 significantly outperforms Pixel-BERT with ImageNet-Res50.

5 Analysis

In this section, we provide detailed analyses on a few interesting phenomena we observe during our experiments, which may help guide future exploration. Quantitative and qualitative analysis are provided to support our findings.

Zero-Shot Performance of CLIP in VQA. In the original paper, CLIP is intended as a zero-shot model and shows strong performance on various vision and image retrieval tasks. We are thus curious if CLIP can also perform well as a zero-shot model on V&L tasks that may require complex reasoning. To conduct zero-shot image classification, CLIP (Radford et al., 2021) uses the names of all classes in the dataset as the set of candidate text and predict the most probable (image, text) pair. We thus experiment with a similar setting on VQA but modify the candidate text to be the concatenation of question and answer pair for each question. Moreover, Radford et al. (2021) find a result improvement from prompt engineering. We follow this design by constructing "question: [question text] answer: [answer text]" as the prompt template. The results on VQA v2.0 `mini-eval` are shown in Table 7. All CLIP variants perform at near-chance level in the zero-shot setting while prompt engineering helps only a little. CLIP models also perform worse when the question becomes harder ("other" vs. "yes/no"). All these results suggest the need of a deep interactive model and additional pre-training/fine-tuning.

Unfreezing the Visual Backbone. Because of technical difficulty in fine-tuning the object detector, most V&L models rely on frozen region-based encoders (Lu et al., 2019). However, we find that unfreezing the visual backbone (Section 4) may bring performance improvement. Specifically, we

test the backbone fine-tuning performance of two CLIP features (i.e., CLIP-Res50, CLIP-Res50x4) on VQA (test-dev) [7] and compare with the frozen BUTD-Res101 features.

Without pre-training, BUTD-Res101 achieves higher performance than CLIP-Res50. However, after V&L pre-training, CLIP-Res50 significantly outperforms BUTD-Res101, because CLIP-Res50 benefits more from pre-training (+9.25) than BUTD-Res101 (+5.72). This suggests that unfreezing the visual backbone during pre-training allows CLIP-Res50 to adapt to the pre-training task. We hope that our finding inspires future work to further explore unfreezing the visual backbone in V&L models when computational budget allows.

(a) Original (b) CLIP-ViT-B (c) CLIP-Res50

Figure 3: Grad-CAM Visualization of CLIP-ViT-B and CLIP-Res50 for the question "What color is the woman's shirt on the left?".

Low Detection Performance of CLIP-ViT-B. As shown in Table 1 and Table 2, CLIP-ViT-B with grid features has a large performance degradation compared with other models. We hypothesize that such decrease is due to the lack of visual localization inside the ViT feature map since different pooling strategies may affect the localization ability (Zhou et al., 2016). We thus follow Jiang et al. (2020) to train a detector on Visual Genome over the CLIP-ViT-B grid feature maps to confirm it. We find that the Average Precision (AP) of CLIP-ViT-B is only 0.03, which is much lower than its ImageNet-Res50 alternative (3.14 as we reproduced).

Qualitative Comparison of CLIP Variants. As we discussed above, we suspect that CLIP-ViT-B lacks certain localization ability. To understand this better, we perform Gradient-Based Localization (Grad-CAM) (Selvaraju et al., 2017) to visualize the salient regions of CLIP models. The qualitative example in Figure 3 clearly shows CLIP-Res50 localizes the sentence "What color is the woman's shirt on the left?" better than CLIP-ViT-B. We provide more qualitative examples in the Appendix.

6 Conclusions

In this paper, we propose to leverage CLIP as the visual encoder for different V&L models across various tasks. We experiment with two approaches: in the first, we directly plug CLIP in task-specific fine-tuning; in the second, we integrate CLIP with V&L pre-training and fine-tune on downstream tasks afterwards. A variety of substantial experiments on different V&L tasks demonstrates that CLIP-ViL and CLIP-ViL$_{\text{p}}$ can achieve competitive or better performance as compared to strong baselines. Analyses from different perspectives explain certain intriguing phenomena and offer new directions for future V&L research.

Reproducibility Statement

We provide the code to reproduce the main results in this paper in the supplementary material, which contains comprehensive instructions to reproduce our results. The code and model checkpoints will be made public.

[7] For these ablated models without pre-training, we find it beneficial to freeze the visual encoder of our models. We also reduce the batch size to 32 to allow for more gradient updates. Other hyper-parameters are the same as in previous VQA experiments.

REFERENCES

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018a. 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*, pages 6077–6086.

Peter Anderson, Ayush Shrivastava, Devi Parikh, Dhruv Batra, and Stefan Lee. 2019. Chasing ghosts: Instruction following as bayesian state tracking. In *NeurIPS*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: Universal image-text representation learning. In *ECCV*. ECCV.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Karan Desai and Justin Johnson. 2020. Virtex: Learning visual representations from textual annotations. *arXiv preprint arXiv:2006.06666*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. In *NeurIPS*.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*.

Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. 2019. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. 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*, pages 6904–6913.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. 2020. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13137–13146.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yicong Hong, Cristian Rodriguez-Opazo, Yuankai Qi, Qi Wu, and Stephen Gould. 2020. Language and visual entity relationship graph for agent navigation. In *NeurIPS*.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 2021. Vlnœ bert: A recurrent vision-and-language bert for navigation. In *CVPR*.

Haoshuo Huang, Vihan Jain, Harsh Mehta, Alexander Ku, Gabriel Magalhaes, Jason Baldridge, and Eugene Ie. 2019a. Transferable representation learning in vision-and-language navigation. In *ICCV*.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019b. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4634–4643.

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

Drew A Hudson and Christopher D Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*.

Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1862–1872.

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

Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. 2020. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. 2021. Mdetr–modulated detection for end-to-end multi-modal understanding. *arXiv preprint arXiv:2104.12763*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. 2019. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6741–6749.

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

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. 2020. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. 2020. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. In *International Journal of Computer Vision (IJCV)*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.

Jialu Li, Hao Tan, and Mohit Bansal. 2021a. Improving cross-modal alignment in vision language navigation via syntactic information. *arXiv preprint arXiv:2104.09580*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019a. VisualBERT: A simple and performant baseline for vision and language. In *Arxiv*.

Liunian Harold Li, Haoxuan You, Zhecan Wang, Alireza Zareian, Shih-Fu Chang, and Kai-Wei Chang. 2021b. Unsupervised vision-and-language pre-training without parallel images and captions. In *NAACL*.

Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah A Smith, and Yejin Choi. 2019b. Robust navigation with language pretraining and stochastic sampling. 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 1494–1499.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Jiasen Lu, Batra Dhruv, Parikh Devi, and Lee Lee. 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*.

Ruotian Luo. 2020. A better variant of self-critical sequence training. *arXiv preprint arXiv:2003.09971*.

Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. 2018. Discriminability objective for training descriptive captions. *arXiv preprint arXiv:1803.04376*.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. 2019a. Self-monitoring navigation agent via auxiliary progress estimation. In *ICLR*.

Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. 2019b. The regretful agent: Heuristic-aided navigation through progress estimation. In *CVPR*.

Khanh Nguyen and Hal Daumé III. 2019. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. 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 684–695.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020a. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020b. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991.

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

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. 2020. Learning visual representations with caption annotations. *arXiv preprint arXiv:2008.01392*.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. 2019. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. VL-BERT: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2610–2621.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *CVPR*.

Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. 2018. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *ECCV*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *CVPR*.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6281–6290.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. *arXiv preprint arXiv:2101.00529*.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. 2020. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10012–10022.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *CVPR*.

A APPENDIX

A.1 VISUAL QUESTION ANSWERING

Model Architecture Pythia encodes the question with an attention-based GRU (Chung et al., 2014) network and fuse the information with a multi-modal factorized bilinear pooling network. MCAN takes a LSTM (Hochreiter and Schmidhuber, 1997) as question encoder and an encoder-decoder based modular co-attention network for fusing multiple representations. Both models employ an output classifier on top of the fused representation to predict the final answer. To integrate CLIP for the VQA models, we extract grid features using CLIP. For CLIP-ViT-B models, we reshape the patch representation from the final layer into grid features. For CLIP-ResNet models, we simply take the grid features from the last layer before the pooling.

Implementation Details We follow (Jiang et al., 2020) to resize all input images to have a maximum shorter side of 600 pixels (longest 1000) when keeping the aspect ratio fixed. For training the detector on the VG dataset, we replace the backbone with CLIP visual module using implementation of Faster R-CNN in Detectron2[8]. For training VQA models, we use hyperparameters of the open-source implementation[9] from (Jiang et al., 2020) for the large version of the MCAN and base version of Pythia.

A.2 Image Captioning

Implementation Details For training, we follow the 'long epoch' hyperparameter of the publicly available implementation [10]. During the self-critical stage, we sample 5 captions for each image as in Luo (2020). For training objective, we experiment with the Self-Critical Sequence Training (SCST) in Rennie et al. (2017), where CIDEr (Vedantam et al., 2015) metric is optimized using REINFORCE algorithm (Williams, 1992).

A.3 Vision-and-Language Navigation

Model For the model architecture, we experiment with the basic attentive neural agent as in Fried et al. (2018).

The agent model (i.e., another LSTM) then attends to the visual features and the language representations to predict the actions. At each time step t, the agent attends to the panoramic views $\{v_{t,i}\}_i$ and the instruction $\{w_j\}$ to make the action. The panoramic view is processed with a pre-trained visual encoder (e.g., ResNet) and the instructions are processed by a language LSTM (Hochreiter and Schmidhuber, 1997), denoted $\mathrm{LSTM_L}$. The agent model, $\mathrm{LSTM_A}$, then attends to the visual features and the language representations to predict the actions.

$$g_{t,i} = \mathrm{ResNet}(v_{t,i}) \tag{1}$$

$$x_1, \ldots, x_l = \mathrm{LSTM_L}(w_1, \ldots, w_l) \tag{2}$$

$$input_t = [\mathrm{Attn}(h_{t-1}, \{g_{t,i}\}), \mathrm{Attn}(h_{t-1}, \{x_j\})] \tag{3}$$

$$h_t, c_t = \mathrm{LSTM_A}(input_t, h_{t-1}, c_{t-1}) \tag{4}$$

where h_t and c_t are the hiddens and states of the action LSTM at time step t, respectively. Please refer to Fried et al. (2018) for the implementation details.

Implementation Details We apply our model to two vision-and-language navigation datasets: Room-to-Room (R2R, Anderson et al. (2018b)) and Room-across-Room (RxR, Ku et al. (2020)). R2R is built on the indoor environments from the MatterPort3D dataset (Chang et al., 2017). The environments are split into training (61 environments), unseen validation (11 environments), and unseen test (18 environments). The agent is trained on the training environments (with 14,025 navigation instructions) and tested on separate sets of environments (2,349 in the unseen-validation and 4,173 in the unseen-test). RxR extends the R2R dataset with multiple languages and follow the environment split. Besides the multilingual nature, RxR is also more diverse in the navigation paths and richer in the present language. For R2R dataset, we follow the hyperparameter (e.g., batch size, learning rate, optimizer) of the publicly available implementation [11] R2R-EnvDrop (Tan et al., 2019) and replace the input features [12] with the CLIP features. To reduce the computational cost, the features are pre-extracted and frozen during the training of the navigational agent. For RxR dataset, we take the processed multilingual data provided in Li et al. (2021a) with Stanza tokenizers (Qi et al., 2020a). Since RxR dataset contains instructions longer than R2R, we change the maximum input length to 160 (from 80) and increase the imitation learning ratio from 0.2 to 0.4 to stabilize the training. Other training hyperparameters of RxR are the same as R2R. The models are trained on one RTX 2080 Ti GPU. It takes 1 days to converge in R2R and about 1.5 days to converge in RxR. We report two significant digits for R2R unseen test results following the leaderboard convention.

[8] https://github.com/facebookresearch/detectron2

[9] https://github.com/facebookresearch/mmf

[10] https://github.com/ruotianluo/self-critical.pytorch

[11] https://github.com/airsplay/R2R-EnvDrop

[12] https://github.com/peteanderson80/Matterport3DSimulator

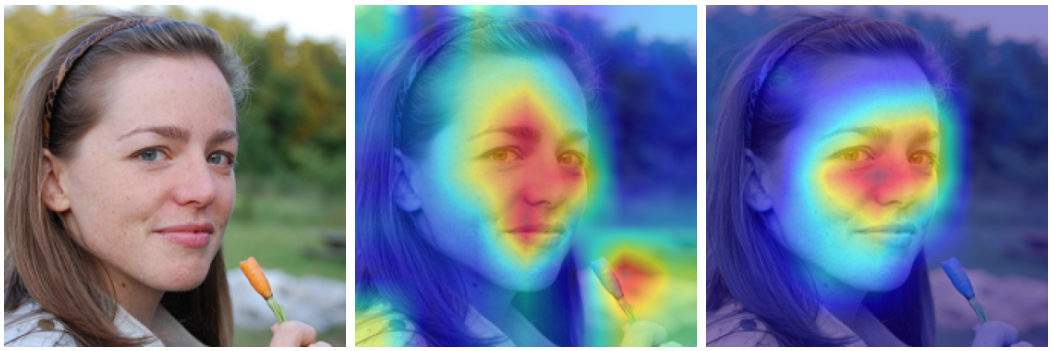
Table 9: Comparison between grid features, CLIP features, and ImageNet-trained features on the R2R dataset. 'SR' and 'SPL' are success rate and success rate weighted by path length.

Feature	Dimension	SR	**SPL**
ImageNet-Res152	2048	48.2	44.4
CLIP-Res50	1024	52.6	47.4
Grid-Res50	2048	47.6	44.7
Grid-ResX101	2048	46.5	43.2
Grid-ResX152	2048	47.8	44.6

Results Comparison to Grid Features In the main paper, we compare the results regarding the ImageNet-pre-trained ResNet-152. We also report the comparison to grid features Jiang et al. (2020) that is trained with detection dataset. Jiang et al. (2020) showed that the results with these features are comparable to the original bottom-up attention with a heavy detection module. The same as the VQA task in Section 3.1, we test the performance of these detection-trained grid features on VLN tasks. Specifically, we use the mean pooling of the feature map as the representation of each view following previous works (Anderson et al., 2018b). As shown in Table 9, under the same ResNet50 backbone [13], we find that the detection-trained grid features are on par with the classification-trained grid features, still showing a gap to the contrastive-trained grid features. We hypothesize that the grid features inject regional knowledge into the dense feature map thus showing good results with grid-based modules (as shown in Section 3.1). However, pooling the feature map into a single feature vector (as in previous VLN works) leads to a loss of this dense information.

A.4 Details of CLIP-ViL$_{\text{P}}$

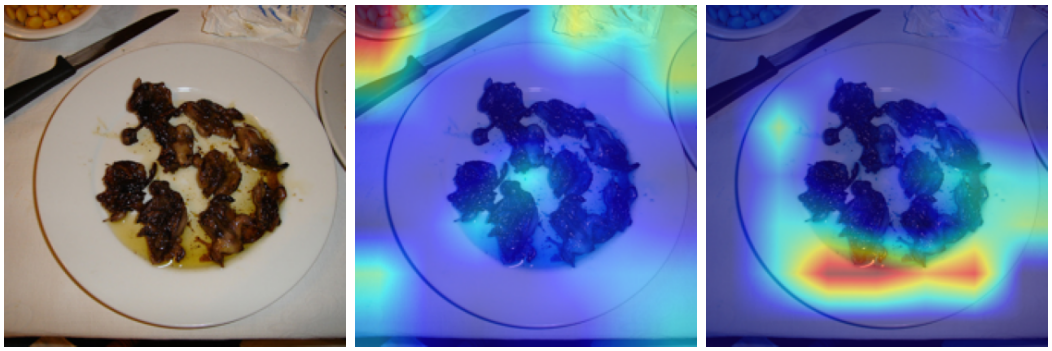
Pre-training We pre-train with a batch size of 512. The Transformer is initialized from BERT$_{\text{BASE}}$ and optimized with an AdamW (Loshchilov and Hutter, 2017) optimizer. We use a linearly-decaying schedule and a peak learning rate of 1×10^{-4} for the model with CLIP-Res50 and 5×10^{-5} for the model with CLIP-Res50x4. The ResNet is initialized from CLIP and we use SGD with a learning rate of 3×10^{-3}. We decay the learning rate of SGD at epochs 12, 17 by a factor of 10. Per the suggestion of Tan and Bansal (2019), we only add the visual question answering loss during the later stage of the pre-training (the last 11 epochs) as the model is prone to overfit to the visual question answering loss. The model is trained on 8 Nvidia A100 GPUs and the pre-training takes around 5 days.



(a) Original (b) CLIP-ViT-B (c) CLIP-Res50

Figure 4: Grad-CAM Visualization of CLIP-ViT-B and CLIP-Res50 for the question "What color are her eyes?".

[13] The CLIP model uses an attention pooling module and makes modifications over the original ResNet (He et al., 2016) backbone.



(a) Original (b) CLIP-ViT-B (c) CLIP-Res50

Figure 5: Grad-CAM Visualization of CLIP-ViT-B and CLIP-Res50 for the question "What is just above the plate?".

Fine-tuning We fine-tune CLIP-ViL$_{\text{p}}$ on three tasks: VQA v2.0, SNLI-VE, and GQA. We introduce the task specifics and fine-tuning hyper-parameters in the following.

Every example in VQA consists of an image and a question, where the task is to predict the correct answer. We use the Karpathy split for training and validation (Karpathy and Fei-Fei, 2015). We fine-tune the model with the binary cross-entropy loss for 5 epoch with a batch size of 256. The Transformer is optimized with AdamW and a peak learning rate of 5×10^{-5}. The ResNet is optimized with SGD and an initial learning rate of 1×10^{-3}. We decay the learning rate of ResNet by a factor of 10 after epoch 3.

SNLI-VE is a three-way classification task, which involves determining the relation between an image and a sentence. The three possible relations include entailment, contradiction, and neutral. We fine-tune the model with the negative log-likelihood loss for 2 epoch with a batch size of 256. The Transformer is optimized with AdamW and a peak learning rate of 5×10^{-5}. The ResNet is optimized with SGD and an initial learning rate of 1×10^{-3}. We decay the learning rate of ResNet by a factor of 10 after epoch 1.

GQA follows the format of VQA but the questions and answers of GQA are automatically generated from ground-truth scene graphs. We use the same hyper-parameters as in VQA.

A.5 More Qualitative Examples

Here we present more qualitative examples using (Grad-CAM) (Selvaraju et al., 2017) to visualize the salient regions of CLIP models. Figure 4 and Figure 5 suggest that CLIP-Res50 localizes the sentence better than CLIP-ViT-B.