Probing the Linguistic Capacity of Pre-Trained Vision-Language Models

Anonymous ACL submission

Abstract

How do recent vision-language pre-trained models compare against language-specific pre-trained models on common linguistic tasks? In this paper, we assess this in a probing setting. Our results suggest that different multimodal pre-training strategies entail distinct strengths. Although pre-trained language models generally fare better, pre-trained vision-language models can obtain higher average scores in certain scenarios (e.g., CLIP is 2% higher than BERT on SST2). We also analyze and illustrate that the different competences in different model layers cause such performance differences. Our work then proposes fine-tuning techniques to improve the abilities of vision-language models on linguistic tasks.

1 Introduction

A number of pre-trained multimodal models have recently come to prominence, e.g. OpenAI's CLIP (Radford et al., 2021) and VisualBERT (Li et al., 2019). The underlying motivation is the intuition that grounded learning from cross-modal pairs of data brings unique advantages (Tan and Bansal, 2020). In particular, such grounding may entail a better acquisition of essential concepts in natural language, such as colors, shapes, and emotions. Additionally, similar to the common paradigm of pre-training large models and then applying them on related downstream tasks (Qiu et al., 2020), a future direction could be to pre-train large general-purpose models on multimodal domains, and then adopt them to facilitate both multimodal and pure unimodal applications.

However, thus far, the general linguistic capabilities of current pretrained vision-language models have not been studied extensively. In this paper, we consider prominent pre-trained vision-language models (Radford et al., 2021; Tan and Bansal, 2019; Li et al., 2019) and shed light on the question of whether the visual-language grounding helps these pre-trained models better understand linguistic concepts and contributes to the performance on language understanding benchmarks. We are also interested in how and why these pre-trained vision-language models (PVLMs) may exhibit different strengths than pre-trained language models (PLMs). Specifically, we fine-tune PVLMs under few-shot settings (e.g., $K = 32$ data points) for each unique label over tasks in the GLUE benchmark (Wang et al., 2018). Our findings suggest that, somewhat unsurprisingly, current PVLMs still tend to underperform in comparison with PLMs, likely due to noise introduced during the domain transfer process. However, we also observe that under certain conditions, the PVLMs exhibit unique strengths compared to language models, e.g., CLIP (Radford et al., 2021) has strong single sentence classification performance (SST2), and VisualBERT (Li et al., 2019) is more proficient in solving sentence relationship tasks than BERT (Devlin et al., 2019) (MRPC, QNLI, QQP), despite both having the same structure and parameter size.

Through experiments and in-depth analysis, we confirm that PVLMs pre-trained weights indeed contribute to their performance on linguistic tasks, but that with ample training data their performance ceiling is lower than for pure language models. We show that the differences between VisualBERT and BERT can be attributed to a competence discrepancy in particular task-specific layers. Motivated by this, we investigate a fine-tuning technique that trains particular task-specific layers, observing an improvement of 8% on SST2.

2 Related Work

Pre-Trained Models. PLMs have brought substantial gains across numerous linguistic tasks (Devlin et al., 2019; Brown et al., 2020; Radford and Narasimhan, 2018). Inspired by the strong results from linguistic pre-training, cross-modal pre-training has been proposed in the multimodal realm. PVLMs, such as VisualBERT (Li et al., 2019), VilBERT (Lu et al., 2019), and LXMERT (Tan and Bansal, 2019) that adopt similar pre-training strategies as language models were proposed, and demonstrated strong capacity over cross-modality tasks for retrieval and captioning, such as on MSCOCO (Lin et al., 2014), Flickr30k (Plummer et al., 2015), and VQA (Antol et al., 2015). Moreover, some multimodal studies in turn demonstrate that learning grounded language from visual information is beneficial for a model's understanding of natural language (Tan and Bansal, 2020; Tang et al., 2021).

Pre-Trained Models Probing. Numerous studies have sought to assess what and how PLMs learn in their text-driven pre-training process (Rogers et al., 2020). Beyond reflecting linguistic structures and semantics (Hewitt and Manning, 2019; Tenney et al., 2019) as well as world knowledge (Li et al., 2021), some studies show that PLMs can generalize to similar tasks (Hendrycks et al., 2020). Besides probing PLMs for linguistic understanding, there are works investigating multimodal models (Cao et al., 2020) on their cross-modal grounding abilities and probing PLMs about visual knowledge (Ilharco et al., 2021). The study most similar to ours is that of Yun et al. (2021), which evaluates PVLMs with regard to lexical grounding.

Rather than assessing the grounding behavior in commonsense tasks, our goal is to shed light on and explain the linguistic understanding capabilities exhibited by PVLMs. We find that differences among models are due to their competency at different layers for different tasks, and accordingly propose a custom fine-tuning technique for PVLMs.

3 Probing Assessment

3.1 Methodology

Our approach to evaluate pre-trained multimodal models will follow the standard probing methodology for language models (Adi et al., 2016; Conneau et al., 2018; Hewitt and Liang, 2019).

Let $\mathbf{h}^x$ represents the representation produced by model for a given input x, and $\mathbf{h}^x_{[\mathrm{CLS}]}$ denotes the class-level representation, typically for the custom class-level token [CLS]. We apply a linear classifier $\mathbf{W} \in \mathbb{R}^{d \times N}$, where d is the dimension of model features and N is the number of class labels, with a Softmax activation function, and maximize the probability of the expected label y by optimizing model parameter θ:

$$\arg\max_{\theta} \ p(y \mid \mathbf{W}\mathbf{h}^x_{[\mathrm{CLS}]}) \qquad (1)$$

Training. We fine-tune and evaluate PVLMs and PLMs over the GLUE benchmark (Wang et al., 2018). Table 4 gives an overview of the models considered in the experiments. To comprehensively evaluate the models, we adopt different fine-tuning strategies, such as fine-tuning all parameters, freezing the pre-trained weights and then tuning the classifier, and adopting BitFit (Ben-Zaken et al., 2021) in Appendix E. Training details are given in Appendix B.

3.2 Main Results

3.2.1 Full Parameter Fine-tuning

Table 1 shows the results of fine-tuning the entire models with few-shot data ($K = 32$ samples). Such few-shot settings prevent the model from absorbing extra knowledge from the probe's training data and thus requires models to rely extensively on the knowledge acquired during pre-training. Overall, PLMs obtain superior results in comparison with PVLMs, and among the considered models, RoBERTa achieves the best results on average.

However, we also observe that VisualBERT shows small improvements over BERT and DistilBERT, despite having the same parameter count and structure as BERT. Specifically, VisualBERT exhibits lower performance than BERT on SST2 and COLA, and better results on most multi-sentence corpora. We hypothesize that VisualBERT's pre-training, which requires inferring the relationship between images and texts, may strengthen its reasoning capacity.

CLIP and LXMERT obtain relatively lower scores, and CLIP performs worse over most sentence pair tasks, but we can observe that CLIP has more robust performance on SST2 even compared with PLMs like BERT and DistilBERT. We conjecture that this could be due to CLIP's pre-training, as the separate text encoder does not require information from images. Thus, the learning capacity can easily transfer to sentence classification tasks, unlike VisualBERT. However, such separate encoder setting also impedes the model's cross-sentence ability because the ability can not be directly learned from image-text pairs. For LXMERT, the results suggest that the model faces more pre-training and fine-tuning mismatch issues, leading to lower scores.

Models	SST2	COLA	MNLI	MRPC	QNLI	QQP	RTE	WNLI	STSB	AVG
					Pre-trained Language Models					
BERT	0.771	**0.181**	0.438	0.569	0.574	0.642	0.517	0.471	0.729	0.544
RoBERTa	**0.848**	0.157	**0.494**	**0.680**	**0.702**	**0.741**	0.531	0.448	**0.732**	**0.593**
DistilBERT	0.761	0.066	0.418	0.578	0.576	0.650	0.523	0.504	0.697	0.530
					Pre-trained Vision-Language Models					
CLIP	0.798	0.047	0.345	0.592	0.543	0.620	0.514	0.471	0.437	0.485
VisualBERT	0.647	0.078	0.430	0.650	0.623	0.696	**0.557**	**0.526**	0.701	0.546
LXMERT	0.569	-0.014	0.348	0.515	0.530	0.534	0.495	0.491	0.161	0.409

Table 1: Results on GLUE in few-shot scenario, reporting average scores over 3 different runs. **Bold** denotes the best results, and underlining emphasizes the second best results.

Models	SST2	QQP	MNLI	QNLI	AVG
BERT	**0.847**	0.746	0.458	0.571	0.656
RoBERTa	0.798	0.719	0.407	0.602	0.632
DistilBERT	0.808	0.739	0.437	0.596	0.645
CLIP	0.826	0.705	0.385	0.531	0.612
VisualBERT	0.773	**0.756**	**0.504**	**0.667**	**0.675**
LXMERT	0.607	0.668	0.324	0.538	0.534

Table 2: Fine-tuning with frozen pre-training weights, for $K = 1000$. **Bold** denotes the best results, and underlining highlights the second best results.

Models	SST2	QQP	MNLI	AVG
BERT	0.899	0.793	0.707	0.800
RoBERTa	**0.924**	**0.829**	**0.820**	**0.858**
DistilBERT	0.897	0.781	0.689	0.789
CLIP	0.893	0.745	0.561	0.733
LXMERT	0.793	0.687	0.455	0.645
VisualBERT	0.877	0.780	0.650	0.769

Table 3: Full training ($K = 2000$). **Bold** / underlining denote best / second best results, respectively.

3.2.2 Parameter Frozen Fine-tuning

Table 2 provides the results of fine-tuning models while freezing pre-trained parameters and only training a classifier at the top of models with training data $K = 1000$ and learning rate 0.001. In this setting, the pre-trained knowledge and layers remain unaltered and the training data can only affect the final classification probe layer. We find that VisualBERT achieves the best average results among all models, including PLMs, with a sizeable gain on SST2 and a noticeable margin on further three tasks. The variance of average scores among different models shrinks compared with the results in Table 1.

4 Discussion and Analysis

4.1 Performance Upper Bound

In previous experiments, we imposed various constraints on the fine-tuning to investigate the pre-trained knowledge. To evaluate the limits of various models, we fine-tune models with a large data size of $K = 2000$ (5 epochs, batch size 16). The results in Table 3 show that PLMs have stronger learning capacity and attain better results. Although VisualBERT has better cross-sentence knowledge under few-shot settings, BERT has the capacity to learn more task-specific knowledge when sufficient training data is provisioned. Moreover, the gap between VisualBERT to CLIP and BERT is reduced. Appendix 9 shows that increased data sizes benefit all models.

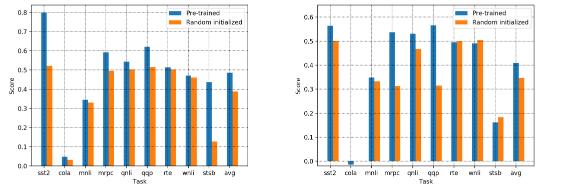


Figure 1: CLIP (left) and LXMERT (right) with different tasks.

4.2 Pre-trained Weights

How much do the pre-trained weights in PVLMs really contribute to the performance? Considering that there is a distribution shift from vision–language data to linguistic tasks, the pre-trained weights may not be sufficiently useful. In the previous experiments, the scores of LXMERT are consistently low, raising the question whether its pre-trained weights provide useful knowledge for linguistic tasks. To answer this question, we compare the pre-trained models with their randomly initialized versions. Figure 1 demonstrates that pre-trained weights from vision-language training indeed contributes to linguistic tasks. However, in a few cases, e.g., LXMERT on STSB, RTE, and WNLI, randomly initialized models obtain better results.

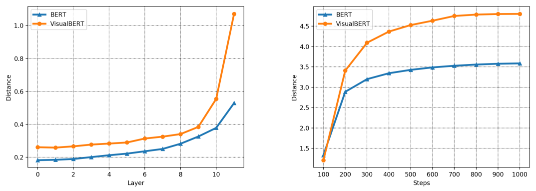


Figure 2: Parameter distance across layers in the best model (left) and across steps in training (right) on SST2.

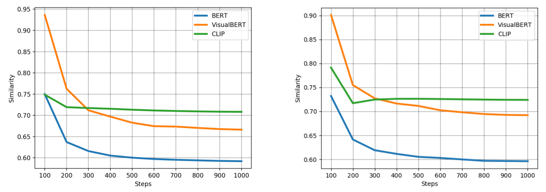


Figure 3: Cosine similarity over fine-tuned and pre-trained representations on SST2 (left) and QQP (right).

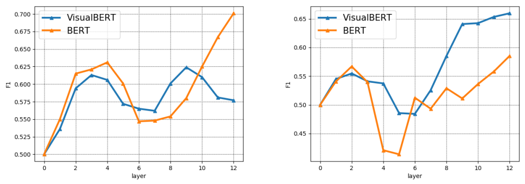


Figure 4: Results on SST2 (left) and MRPC (right) when fine-tuning on each intermediate representation.

4.3 Assessing Parameter Distance

Another scheme we use to compare PVLMs with PLMs is (1) the parameter distance between fine-tuned weights and pre-trained weights by computing $\sum_i |\mathbf{w}_{\text{ft}}^i - \mathbf{w}_{\text{pt}}^i|^2$, where i is the layer, $\mathbf{w}_{\text{ft}}$ denotes fine-tuned weights, $\mathbf{w}_{\text{pt}}$ are pre-trained weights, and (2) the cosine similarity between fine-tuned representations and pre-trained representations. One might assume bigger distances and smaller similarity scores correspond to larger domain gaps, but we find that PVLMs typically have bigger distance yet higher similarity scores. Figure 2 provides an example plotting the distance of each layer in the best model and across steps, while comprehensive results are given in Figure 8. Figure 3 provides the cosine similarity changes. We observe that most parameter changes occur in top layers, and the overall distance tends to enlarge as training proceeds. In Figure 3, VisualBERT has a higher similarity score, then drops drastically after 100 steps. BERT and CLIP initially remain close in terms of the similarity but soon adapt as training continues. VisualBERT experiences more parameter changes in Figure 2, both overall and in individual layers, yet has higher similarity in Figure 3.

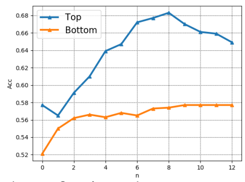


Figure 5: Results of VisualBERT on SST2 when fine-tuning selected layers. Top: freezing layers $< n$ and fine-tuning layers $\geq n$. Bottom: freezing layers $> n$ and fine-tuning layers $\leq n$.

4.4 Layer Representations

To fully compare models, especially what competencies are required for applications, we adopt approaches to model truncation (Merchant et al., 2020). We train classifiers using representations from intermediate layers rather than the final one.

Figure 4 shows that models perform similarly when using initial layers, across different tasks such as SST2 or MRPC, and it decreases when considering intermediate layers in the middle. However, the performance diverges when top layers participate, and models may rely on different layers for different tasks. For example, on SST2 the divergence between BERT and VisualBERT happens in layers 8-12, and on SST2 it occurs in layers 6-9. Models are known to capture different kinds of downstream knowledge in different layers, and thus there is a knowledge discrepancy in different layers. This discrepancy may explain why VisualBERT performs worse on SST2 but better over sentence relationship tasks in Table 1.

Inspired by this observation, we conducted additional experiments by fine-tuning only task-specific layers and freezing other layers. The selection of task-specific layers is based on the empirical investigation in Figure 4. We compare it with results when selecting other layers. Figure 5 reveals that VisualBERT achieves the best SST2 results (8% higher than full parameter fine-tuning) when only tuning layers 8–12, which are the task-specific layers in Figure 4.

5 Conclusion

In this paper, we employ PVLMs on text-only tasks and provide a series of experiments to compare PVLMs with PLMs and analyze their performances. We find that different PVLMs have different performance patterns. But generally, PVLMs tend to have worse performance and lower performance upper ceiling. We conjecture that this is because of the discrepancy at each layer and propose fine-tuning task-related layers to improve the performance.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, abs/1608.04207.

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

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *ArXiv*, abs/2106.10199.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jize Cao, Zhe Gan, Yu Cheng, Licheng Yu, Yen-Chun Chen, and Jingjing Liu. 2020. Behind the scene: Revealing the secrets of pre-trained vision-and-language models. In *ECCV*.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. *ArXiv*, abs/1909.03368.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *NAACL*.

Gabriel Ilharco, Rowan Zellers, Ali Farhadi, and Hannaneh Hajishirzi. 2021. Probing contextual language models for common ground with visual representations. In *NAACL*.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *ACL/IJCNLP*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft coco: Common objects in context. Cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.

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

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2641–2649.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *ArXiv*, abs/2003.08271.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pretraining. *OpenAI Blog*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. 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 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2020. Vokenization: Improving language understanding with contextualized, visual-grounded supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2066–2080, Online. Association for Computational Linguistics.

Zineng Tang, Jaemin Cho, Hao Tan, and Mohit Bansal. 2021. VidlanKD: Improving language understanding via video-distilled knowledge transfer. In *Advances in Neural Information Processing Systems*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *ArXiv*, abs/1905.05950.

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

Tian Yun, Chen Sun, and Ellie Pavlick. 2021. Does vision-and-language pretraining improve lexical grounding? In *EMNLP*.

A Overview of Compared Models

Table 4 provides detailed information of the models considered in the comparison.

B Training Details

Unless stated, the default setting for training is $K = 32$ instances, the validation size is 200, and test size is 500. The batch size is 2, learning rate is 1×10^{-5}, the total number of training steps is 1000, and the number of validation steps is 100. We select the checkpoint with the highest validation scores for testing. We generate fake image representations for VisualBERT and LXMERT.

C Faked Image Features

In this additional experiment, we alter generated image feature settings to evaluate whether these irrelevant image features necessarily bring noise and distribution shift that hurts models on language tasks.

C.1 To Fake or not to Fake

In order to fulfill the input requirements of considered models, we create image features as needed. However, we can use certain settings to avoid involving image features. For VisualBERT, this encompasses deleting visual position embeddings, and for LXMERT, we can take the language encoder outputs as the inputs to the classifier. Hence, the models can avoid unnecessary noise and might be expected to obtain better results. Our findings in Figure 6 show that excluding image features does not always bring an improvement. In the left part, including image features can bring score increases for LXMERT. We believe that this is because the models can identify the artificial noise and can avoid incorporating these signals into further computations.

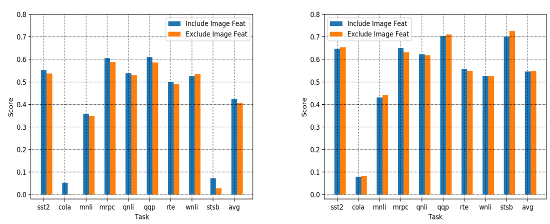


Figure 6: Performance over different tasks when including image features or not. Left: LXMERT, right: VisualBERT.

C.2 To Randomize or not to Randomize

We also wish to know whether randomized image features or constant image features are better for the models to process. In this experiment, we initialize the image presentation with different values and then fine-tune models on SST2 dataset with such image features. The results in Figure 7 show that altering the initialization does not affect the results significantly.

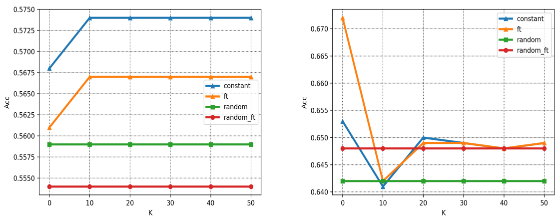


Figure 7: Performance over SST2 with different initialized image feature values. Left: LXMERT, right: VisualBERT.

C.3 To Tune or not to Tune

We next study what happens if we allow models to change the image features during training. Could this make the image features more suitable for the models? In Figure 7, we observe that allowing changes to the features does not bring any benefit to a model's understanding of sentences.

Overall, the studies in Sections C.1, C.2, and C.3 show that incorporating and changing synthesized image features typically does not affect PVLMs significantly. The experiments thus corroborate the feasibility of applying PVLMs on language tasks without facing vast domain adoption challenges and more generally lends further credence to the idea of applying multimodal models on individual modalities.

D Parameter Changes

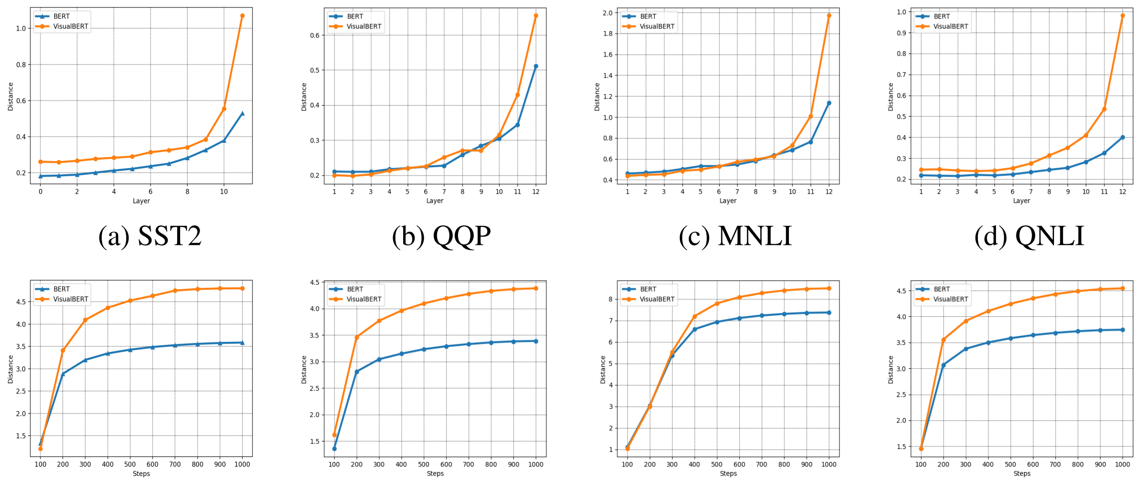
Figure 8 provides comprehensive experiments on parameter changes of VisualBERT and BERT.

E BitFit Tuning

BitFit (Ben-Zaken et al., 2021) is a sample-efficient fine-tuning approach that only trains bias terms. Hence, only 0.08% of parameters are trained to control the use of the pre-trained knowledge, but the pre-trained knowledge itself remains largely unchanged. We can think of this as a manner of probing whether these models directly learned knowledge valuable for downstream tasks.

Category	Models	Layer	Size	Heads	Parameter	Image
PLMs	BERT (Lu et al., 2019)	12	768	12	110M	N
	RoBERTa (Liu et al., 2019)	12	1024	12	125M	N
	DistilBERT (Sanh et al., 2019)	6	768	12	66M	N
PVLMs	CLIP (Radford et al., 2021)	12	512	8	38M	N
	LXMERT (Tan and Bansal, 2019)	14	768	12	123M	Y
	VisualBERT (Li et al., 2019)	12	768	12	110M	Y/N

Table 4: Overview of Models used in experiments. Layer: hidden layers, Hidden Size: representation size, Heads: self-attention heads, Parameter: total parameter, Image: requiring image input or not. N represents no requiring, Y requires images, and Y/N denotes the model can switch from including image inputs or not.



(a) SST2 (b) QQP (c) MNLI (d) QNLI

(e) SST2 (f) QQP (g) MNLI (h) QNLI

Figure 8: Results for parameter and similarity changes across layer (top) and training steps (bottom).

Models	SST2	QQP	MNLI	QNLI	AVG
BERT	0.88	0.766	0.612	0.759	0.754
RoBERTa	**0.9**	**0.794**	**0.761**	**0.800**	**0.814**
DistilBERT	0.859	<u>0.778</u>	0.615	0.737	0.747
CLIP	<u>0.894</u>	0.758	0.506	0.682	0.710
LXMERT	0.690	0.678	0.370	0.559	0.574
VisualBERT	0.861	0.785	<u>0.650</u>	<u>0.765</u>	<u>0.765</u>

Table 5: Bitfit tuning with $K = 1000$. **Bold** denotes the best results, and <u>underline</u> emphasizes the second best results.

Models	SST2	QQP	MNLI	QNLI	AVG
BERT	**0.858**	0.647	<u>0.485</u>	0.547	<u>0.634</u>
RoBERTa	<u>0.852</u>	**0.746**	0.439	<u>0.578</u>	**0.654**
DistilBERT	0.818	<u>0.667</u>	0.415	0.560	0.615
CLIP	0.806	0.659	0.329	0.523	0.579
LXMERT	0.578	0.625	0.339	0.525	0.517
VisualBERT	0.729	<u>0.667</u>	**0.502**	**0.633**	0.633

Table 6: Bitfit tuning with $K = 32$. **Bold** denotes the best results, and <u>underline</u> emphasizes the second best results.

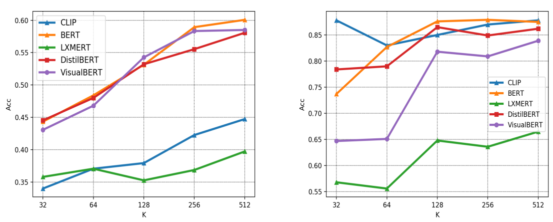


Figure 9: Results of models on MNLI (left) and on SST2 (right) with different K.

In this experiment, we finetuned models with two different K values – 1000 and 32, as we want to investigate the influence of the training data size. Results are given in Tables 5 and 6. Similar to the corresponding results in Section 3.2.1, VisualBERT tends to show a strong sentence relationship reasoning capacity across different K, VisualBERT can always achieve better results over MNLI, QQP, and QNLI in comparison with BERT and DistilBERT. However, RoBERTa can benefits more strongly from large K. CLIP shows strong results on SST2, but still generally underperforms on other tasks.

F The Effect of Training Data Size

In this section, we are interested in what we can find if we gradually increase the training data size, especially observing that there is a boost for VisualBERT on SST2 in Table 1 and Table 3. Thus, in these experiments, we gradually consider larger training data sizes $K \in \{32, 64, 128, 256, 512\}$, as reported in Figure 9.

As expected, an increase in the number data points benefits all models. Apart from drawing the same conclusions as in Section 3.2.1, one interesting additional observation is that there is a jump for VisualBERT on SST2 when the data size increases from 64 to 128. This might indicate that VisualBERT can learn knowledge for solving SST2 given sufficient data, but does not capture enough about this task during pre-training. Thus, on SST2, all models except LXMERT gradually converge, while on MNLI, the gap between CLIP and language models remains constant.