BUILDING COMPACT REPRESENTATIONS FOR IMAGE-LANGUAGE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a method to learn compact vision and language representations, which adaptively and iteratively fuses the multi-modal features. It greatly lowers the FLOPs of the model by effectively combining and reducing the number of tokens used for both text and images. This allows the model to scale without a large increase in FLOPs or memory and leads to a data efficient training. In addition, we propose adaptive pre-training data sampling which further improves the data efficiency. We achieve competitive performance compared to much larger models, and do so with significantly less data and FLOPs. With only 40M training examples and with 39 GFLOPs our model of 350M parameters outperforms all methods that have used less than 1B examples for pre-training. Code will be released.

1 INTRODUCTION

Vision and language learning models have made big strides recently, enabling novel capabilities of natural language interaction with the visual world, such as question-answering, or providing descriptions or reasoning about images. Vision and language models have achieved high performance by scaling of the model architectures, which in turn relies on the availability of very large datasets.

One important component of these models is building the underlying joint visuo-lingual representation which captures the relations between the modalities.

Recent vision and language representation learning approaches share core architecture elements, effectively employing the Transformer model (Vaswani et al., 2017) to learn across modalities (Wang et al., 2021; Li et al., 2021; Dou et al., 2022; Li et al., 2022).

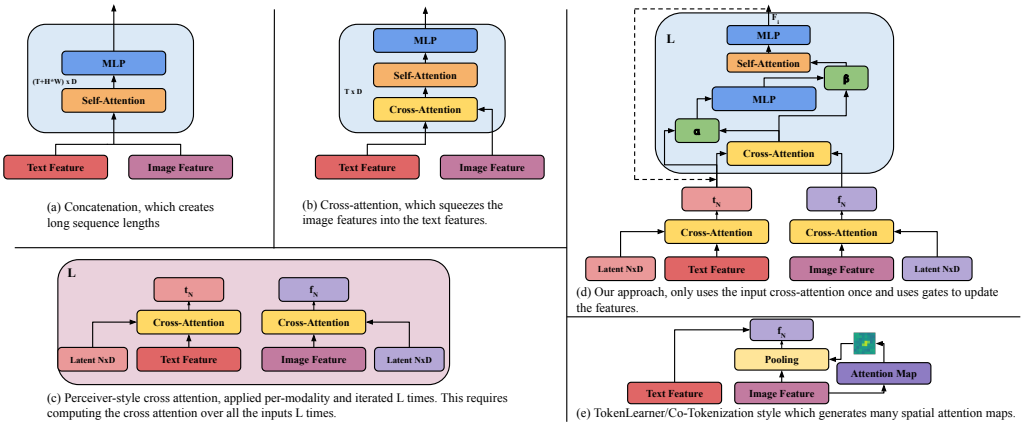

Figure 1: Illustration of our proposed approach (d) and comparisons to others. Unlike Perceiver-style models (c), we only process the input a single time, and unlike TokenLearner approaches, we do not use spatial attention maps. Ours results in more compact features than concatenation (a), while still maintaining more complex features than cross-attention (b).

However, expensive attention mechanisms are applied within Transformers, in which the compute required grows quadratically with the increase of the input sizes; further, these models perform better

with significantly more data (Dosovitskiy et al., 2021) and training steps to learn the joint representations; and lastly, since large data are hard to collect, automatically collected datasets contain large amounts of noise. All this makes these models even more ineffective and expensive to train: scaling the models, combined with the corresponding data scaling required, and training with large amounts of noise, require large amounts of compute. Thus, it is desirable to construct more memory- FLOPs- and data- efficient vision-language representations where one can take advantage of model scale but in a more effective way.

To that end, we propose a vision-language representation learning, the Joint Adaptive Representation, which allows efficient joint image-language learning (Figure 2). This approach first reduces the number of tokens in the input modalities, then adaptively fuses them. This process greatly reduces FLOPs, while maintaining or improving performance. Our approach results in a more compact and efficient representations, obtaining 33% fewer FLOPs than the commonly used concatenation, while improving performance. This leads to more data- and compute- efficient models.

Furthermore, pre-training has been critically important for successful vision and language methods. In many cases, very large datasets and compute intensive training are needed to establish alignment between modalities (Wang et al., 2021). We find that the proposed method reduces the needed pretraining data, due to the reduced representation size. In addition, we also propose an adaptive sampling of the pretraining datasets and tasks, which progressively increases the focus on harder tasks. As a result of the above-mentioned innovations, pre-training is done in a more data-efficient manner, capturing the vision-language features more effectively. Data efficiency, combined with fewer FLOPs, reduces the overall cost of the model.

We evaluate the approach on Visual Question Answering tasks, where understanding jointly the image in the context of the language input is important. Our model performs competitively with respect to the state-of-the-art (SOTA) models, outperforming all models of standard parameter and data scale (Figure 2). Further, the model outperforms other efficient joint vision-language learning methods (Piergiovanni et al., 2022b; Jaegle et al., 2021), surpassing them both in accuracy and in reducing FLOPs. It also scales well with increasing model size and input image size.

The main contributions of our work are: (1) a new image-text fusion method that is more efficient that previous methods; (2) an effective method to mix and pre-train image-text models on smaller datasets.

2 Related work

Vision and language modeling has been of much interest, with many tasks and methods showing great progress. Many approaches have found benefits from scaling the models and data (Radford et al., 2021; Wang et al., 2021). Most commonly, Transformers (Vaswani et al., 2017) are used for multi-modal fusion where the inputs are most often concatenated and the fusion is effectively done by the Transformer itself (Chen et al., 2020; Li et al., 2019; Zhang et al., 2021). Some of these models have also been scaled significantly, which, given the costs of the Transformers, leads to expensive approaches. For example, SimVLM (Wang et al., 2021) relies purely on concatenation and a large Transformer model of 1.5 billion parameters and with about 1.8 billion pretraining samples to achieve strong performance. Co-attention mechanisms within the Transformer or other architectures are popular as well, while still using primarily concatenation of features (Lu et al., 2019; Tan & Bansal, 2019; Nguyen & Okatani, 2018). A recent approach, METER (Dou et al., 2022) uses a co-attention mechanism with concatenation, as well. Other works have studied methods to learning combined vision and language features. For example, ALBEF (Li et al., 2021) and BLIP (Li et al., 2022), use cross-attention to fuse the features, but as we show later, this can be difficult for the model, as it relies on the text length. All of these works have shown benefits from these methods, however, here we propose both better and more efficient fusion mechanisms, which can yield better-scaled models and less costly ones as well.

Other recent works have studied methods for reducing input tokens (Ryoo et al., 2021) and methods to combine multimodal inputs, and reduce their size. For example, Perceiver (Jaegle et al., 2021) iteratively fuses multiple inputs and reduces their sizes and Iterative Co-Tokenization (Piergiovanni et al., 2022b) iteratively selects different vision features with spatial attention. While these approaches were shown to be effective, we find here the compute requirements can further be reduced

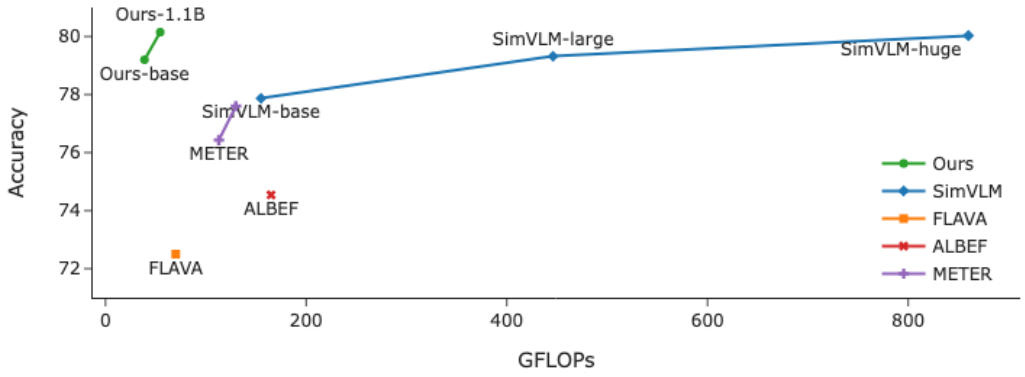

Figure 2: GFLOPs vs. performance for several models. The proposed approach enables much more efficient scaling, and achieves excellent performance for fewer FLOPs. It outperforms SimVLM-huge on VQA2.0 dataset, even though our model is evaluated in the open-vocabulary setting.

by removing the expensive spatial attention maps and iterative updates, while also improving the performance.

Vision-and-language learning approaches have developed powerful pre-training methods following the success of language models; pre-training objectives are either directly borrowed or appropriately adapted to vision-language learning. In these, pre-training is typically done on a single dataset, for example, ViLBERT (Lu et al., 2019) uses the Conceptual Captions 3M dataset (Sharma et al., 2018), VisualBERT (Li et al., 2019) uses COCO Captions (Chen et al., 2015); in some cases very large datasets are used, for example, SimVLM (Wang et al., 2021). Many of the recent vision and language works use a mix of pretraining datasets and tasks, e.g., FLAVA (Singh et al., 2022) uses 70M examples from a variety of sources. ALBEF (Li et al., 2021), BLIP (Li et al., 2022), METER (Dou et al., 2022), VinVL (Zhang et al., 2021), UNITER (Chen et al., 2020), LXMert (Tan & Bansal, 2019) also mix a set of tasks or datasets, common examples are COCO captions (Chen et al., 2015), Conceptual Captions (Sharma et al., 2018), Visual Genome (Krishna et al., 2016), YFCC (Thomee et al., 2015), etc. We follow these works, but propose a new way to sample and utilize small, public datasets to achieve strong performance.

3 Joint Image-Language Representation Learning

The key question we address is how to combine the features from vision and language input modalities. A few basic approaches use either: (1) concatenation or (2) cross-attention. A key issue with concatenation is that it greatly increases the number of tokens by adding $H * W$ to the text length (H, W are the height and width of the image features). Thus, as the image size increases, concatenation greatly increases the FLOPs and memory requirements of the model, e.g., (Wang et al., 2021; Dou et al., 2022; Lu et al., 2019; Su et al., 2019; Li et al., 2019; Lu et al., 2020; Chen et al., 2020). Here, we propose a method to reduce the number of tokens, improving efficiency.

Cross-attention based methods have other issues, mainly that the modality used for the query (usually text, e.g., ALBEF (Li et al., 2021), BLIP (Li et al., 2022)), determines the size of the output representation. Often for vision-language tasks, the visual features have many tokens (for example, the visual tokens are 14x14 = 196 for a modest image input size of 224x224), while text is fairly short, e.g., 10 tokens in VQA2.0. When using cross-attention, the entire visual input must be squeezed into these few text token representations, greatly constraining the amount of visual information that can be used. While this approach has fewer FLOPs than concatenation, it loses information, which can reduce task performance, and puts a dependence on the input text length. Naturally, this cross-modal representation will have even less utility when increasing the input image size.

Instead, we here propose a module that enables better learning of vision-language features by more effectively incorporating the visual information and fusing it with the text information. By adaptively and iteratively tokenizing the inputs, the model is able to refine the feature representation learned from both modalities in the training process, while keeping a reasonable number of FLOPs.

3.1 JOINT ADAPTIVE REPRESENTATION

Our approach is based on several insights. First, we query the image to obtain more informative visual tokens. Previously, this was done using a TokenLearner-like approach (Piergiovanni et al., 2022b; Ryoo et al., 2021). However, this method, while reducing FLOPs, notably for video applications in (Piergiovanni et al., 2022b), still uses quite a few FLOPs to generate and apply the attention maps, and does not scale well with image size. Instead, we utilize a hybrid approach inspired by Perceiver (Jaegle et al., 2021). We generate N tokens independently from each modality as a first step. Secondly, we then use a direct cross-attention mechanism between the new text and compact visual features to produce a better cross-modal representation. This mechanism consists of a cross-attention layer, then a self-attention layer, and a Multi-Layer Perceptron (MLP), similar to a standard Transformer layer (Vaswani et al., 2017), but due to the reduced tokens, is much more lightweight.

Finally, this process is done iteratively, thus refining the current representation based on the set of features from the Transformer. This allows the model to dynamically update and select different visual and text features at each step so it is best able to perform the task, without increasing the compute cost. Our approach is described in detail below.

Let X_{text} and X_{im} be the inputs for text and for images, respectively. More specifically $X_{text} \in \mathbb{R}^{L \times D}$ and $X_{im} \in \mathbb{R}^{H \times W \times C}$, assuming the visual input is of size $W \times H$, the text is of length L. The goal is to produce new, lower dimensional feature representations. This can be done by reducing the representation to a lower number of tokens, which is particularly important for the visual features as they are many more. This is done by first unifying the representation dimensions, more specifically projecting the visual features to the $H * W \times D$ space, where D is the feature dimensions for the text input, $P(X_{im}) = W_1 X_{im}$.

In principle both the visual input and the text input can be projected to a new feature dimension e.g., D', thus not having to be necessarily dependent on the input feature dimension.

As a second step, we proceed to learn a set of new N learnable tokens $X_N \in R^{N \times D}$, which is done in a DETR-style (Carion et al., 2020) feature learning. That is, X_N is a randomly initialized representation that is learned via back-propagation jointly with the other parameters to minimize the loss.

$$f_N = W_2 \Phi(X_N, P(X_{im})). \tag{1}$$

Here $P(X_{im})$ represents the projection of visual features from Equation **??**, X_N is the learned latent features, Φ is the standard multi-head attention operation. This results in f_N, the compact intermediate representations with N features. This can also be viewed as learning N new tokens, which represent the input of M tokens, where $N \ll M$, for the large visual input $M = H * W$. We note that this is similar to the Perceiver architecture (Jaegle et al., 2021), albeit it is done only once here. This process is also done to X_{text}, resulting in N text features (t_N). Thus, unlike prior work (e.g., (Li et al., 2021; 2022)), N is not required to be tied to the input text length; so a richer, but more compact representation is built.

Next, for the two inputs t_N, f_N we learn a new joint feature representation $F(t_N, f_N)$ via cross attention. Importantly, we note here that both of these inputs will influence the subsequent representation to create a cross-modal fused version of text and image features. In the co-tokenization approach (Piergiovanni et al., 2022b), the two modalities are also fused for better representation learning, but here with two key differences: 1) the initial token reduction is not done at each iteration, which is a computationally intensive process; and 2) ours uses a lightweight cross-attention compared to the co-tokenization approach.

This process uses the following components. We first use LayerNorm (Ba et al., 2016) (denoted as Ln) in order to normalize the features. We then compute cross-attention between t_N (text features) and f_N (image features). The idea is that they will help construct a representation which is a combination of these modalities. We then use a standard Transformer layer with self-attention and MLPs to compute the features.

$$\begin{aligned} P_{cross}(t_N, f_N) &= Ln(t_N) + tanh(\alpha)\Phi(Ln(t_N), Ln(f_N)) \\ F(t_N, f_N) &= P_{cross}(t_N, f_N) + tanh(\beta)MLP(P_{cross}(t_N, f_N) \end{aligned} \tag{2}$$

where α and β are learnable parameters that control how the text and vision features are fused (Φ is the standard multi-head attention operation). We note that here, throughout, $P_{cross}(t_N, f_N) \in R^{N \times D}$, i.e., is a compact representation which combines the two modalities. We also add the tanh gating mechanism, which we find to be advantageous in our ablation experiments (Section 6.2).

The resultant representation $F(t_N, f_N) \in R^{N \times D}$ is then fed to the Transformer to produce a transformed intermediate representation of the same dimension $F = \mathcal{T}(F(t_N, f_N)) \in R^{N \times D}$. We use a standard Transformer layer ($\mathcal{T}$) with multi-headed attention (Vaswani et al., 2017).

This new feature representation can be further refined to produce even better cross-modal learning by repeating the same process, but this time taking the already obtained feature as input. The operation is the same as Equation 2 but with continually updated input by replacing t_N with $F + t_N$, which adds in the output of the previous Transformer layer. This lets the model continually refine and fuse the features. Assuming F_i is the current representation and F_{i+1} is the next, this uses the previous equations to iteratively update the features as follows:

$$\begin{aligned} P_{cross}(F_i + t_N, f_N) =& Ln(F_i + t_N) + tanh(\alpha)\Phi(Ln(F_i + t_N), Ln(f_N)) \\ F_{i+1} =& P_{cross}(F_i + t_N, f_N) + tanh(\beta)MLP(P_{cross}(F_i + t_N, f_N) \\ F_{i+1} =& \mathcal{T}(F_{i+1}) \end{aligned} \tag{3}$$

At the first iteration we note the text input is used, whereas subsequently the joint features are.

Of key importance is that during the cross-modal learning process, we use the interaction of both modalities. Specifically, we use attention to determine lower dimensional projections from both modalities which differs both from the Transformer (Vaswani et al., 2017) which preserves the input dimensionalities, and is a more efficient process than the Iterative Co-Tokenization (Piergiovanni et al., 2022b) and Perceiver (Jaegle et al., 2021), also used by Flamingo (Alayrac et al., 2022), as the expensive tokenization step over the whole input is only done once here. Further, different from Flamingo are the iterative updates, Equation 3, where we iteratively combine the features, rather than relying only on cross-attention. The approach is also different from methods like TokenLearner which is only applied on a single input, which can lead to a loss in accuracy if not placed appropriately (Ryoo et al., 2021). It is also different from cross-attention methods (Li et al., 2021; 2022; Dou et al., 2022) due to the initial feature learning and iterative updating of the cross-modal information (Equation 1). It is more efficient than these approaches, as well.

This approach also offers better performance than the concatenation baselines while using at least **33% fewer** FLOPs than them (as seen in ablations, Section 6.2).

4 PRE-TRAINING

To strengthen the cross-modal feature representation learning, we use a number of cross-modal pre-training tasks. While conflicting evidence points to either success with one or two pre-training tasks (Wang et al., 2021; Dou et al., 2022) or with more tasks (Li et al., 2021; Dou et al., 2022; Piergiovanni et al., 2022a), we find that a larger mixture of cross-modal tasks is more beneficial for our vision-language model.

Our pre-training mixture includes tasks spanning captioning, split-captioning, token masking and VQA style questions generated automatically from labeled data (which can be object specific), as proposed by some of the above-mentioned works (Li et al., 2021; Dou et al., 2022; Piergiovanni et al., 2022a). Please see appendix for a full list of tasks and datasets. We train all these tasks with a single loss: per-token cross entropy, as they are all text-generative tasks.

4.1 ADAPTIVE TASK SAMPLING

Inspired by curriculum learning, we adaptively change the mixture ratios between the tasks during pre-training. The idea is that initially in the early stages of training, various pre-training tasks will be sampled roughly in equal proportions. However, as training progresses, it will be beneficial if harder tasks are given larger sampling weights within each training batch. Thus the training will progress from easy to hard examples. Since training batches are created for each training step, this

will allow for the pre-training to seamlessly adapt its weights before sampling, so each iteration will focus on the currently challenging tasks.

To implement this, we assume that tasks with a higher loss at a given training step are harder for the model, and sample them more. Since all tasks are of the type (input: (image, text) and output: text) and share the same loss, they are directly comparable and no cross-loss normalization is needed. Specifically, a tasks sampling weight is:

$$\frac{\mathcal{L}_{\mathcal{S}}}{\sum_s \mathcal{L}_s} \tag{4}$$

where $\mathcal{L}_S$ is the value of the current observed loss of a task S. That is, we take the percentage of a task $\mathcal{S}$ loss over the sum of all tasks' losses, computed on the training set. We find this helps balance the tasks so the model is trained on the more difficult tasks, further improving data efficiency. We enforce a minimum number of samples per task, so this can always be computed for all tasks.

We note that our adaptive sampling strategy is different from the common sampling strategies during training, which are typically agnostic of a task's performance, but rather based on the evolution of the task performance during training. For example common methods are uniform sampling or weighted sampling based on data volume or conversely to compensate for tasks or datasets which are under-represented in training. Our finding is also consistent with other observations in which well-performing losses are shown to be 'myopic', i.e., they work well on the task at hand but not as well on downstream tasks (Kornblith et al., 2021).

In our experimental results, (Section 6.2) we see the benefits of the proposed adaptive sampling and particularly for training in a data-efficient manner and with fewer training steps.

5 Implementation details

Model. In this work we use individual modality encoders. Specifically ViT Base/32 (Dosovitskiy et al., 2021) for images and a T5 Base encoder (Roberts et al., 2022) for text. These representations for each modality are taken from scratch, and are then jointly learned as described in Section 3.

Our base model contains about 350M parameters (where most of the parameters are in the encoders). When the model is scaled, it has about 1.1B. Despite larger parameter count the models are very efficient with only 38.9 and 54.5 GFLOPs, respectively, which is much fewer than methods such as ALBEF (Li et al., 2021) with 165, BLIP (Li et al., 2022) with 250 and METER (Dou et al., 2022) with 130; and significantly better than large models, like SimVLM (Wang et al., 2021) with 890 GFLOPs. To scale the model up, we increase from T5-base to T5-large. A T5 decoder is used to decode the output into generated text. Exact details are provided in the supplemental materials; code implementation will also be provided.

Pre-training mix and datasets. In contrast to several prior works that use hundreds of millions, or 1-2 billion images (Singh et al., 2022), (Yuan et al., 2022), (Wang et al., 2021), we use a mix of public datasets with only at most 40M samples. Specifically, we use the Conceptual Captions 3M dataset (Sharma et al., 2018) (CC3M), the Conceptual Captions 12M dataset (Changpinyo et al., 2021) (CC12M), Visual Genome (VG) (Krishna et al., 2016), Open Images (OI) (Kuznetsova et al., 2020) and Localized Narratives (LN) (Pont-Tuset et al., 2020) dataset for OpenImages. Note that we removed any images in the downstream validation/tests sets from these training sets.

Our vision and text backbones are initialized from scratch and only pre-trained by the 40M image-language mixture described above.

6 Experiments

6.1 Evaluation

We evaluate our approach on three VQA datasets **VQA2.0** (Agrawal et al., 2015), **GQA** (Hudson & Manning, 2019), and visual entailment (**SNLI-VE** (Xie et al., 2019)). For each of these, we follow the standard accuracy metrics which is the exact match accuracy. Our model uses the open-vocabulary of the decoder to generate text and we use this output directly to compare to the ground truth. A handful of approaches use open-vocabulary like us, e.g., ALBEF and BLIP (Li et al., 2021;

Table 1: Comparison to the Perceiver (Jaegle et al., 2021) method and to the Iterative Co-tokenization (Piergiovanni et al., 2022b) approach for image+text fusion. Both are our implementations. *Adapted to VQA (The original Iterative Co-Tokenization method is intended for Video Question Answering with multiple video stream inputs, however we find they are not needed for still images and use a single input stream). Base model.

	GFLOPs	GQA	SNLI-VE
Perceiver	40.3	78.2	77.4
CoTok*	43.8	78.5	77.5
Ours	**38.9**	**79.1**	**77.9**

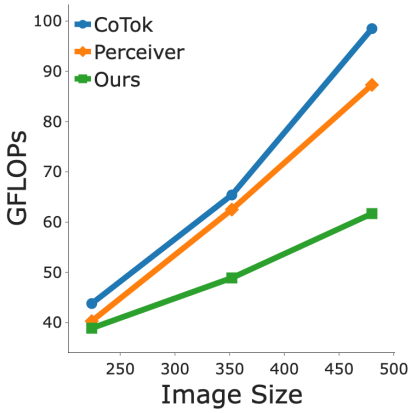

Figure 3: Comparison of the three methods with different image sizes. Since ours does not have the iterative input updates, it scales better to larger inputs.

Brattoli et al., 2020). This is a more challenging scenario than the classification to a fixed vocabulary set. After pretraining of the model, we finetune on each dataset as is also customary in the literature. Please see the appendix for more information.

6.2 Ablation studies

In Table 1, we first directly compare our approach to other joint image-language representation learning methods, specifically the Perceiver (Jaegle et al., 2021) and the Iterative Co-Tokenization (Piergiovanni et al., 2022b). Both approaches are also efficient ones, using fewer FLOPs, compared to other vision-language approaches which tend to use concatenation (Wang et al., 2021; Dou et al., 2022; Lu et al., 2019; Su et al., 2019; Li et al., 2019; Lu et al., 2020; Chen et al., 2020). The Iterative Co-Tokenization (Piergiovanni et al., 2022b) uses spatial attention maps, and both iteratively recompute the visual features based on the raw inputs, which increase FLOPs. As seen, our approach outperforms these already efficient and advanced fusion methods, while using fewer FLOPs (Table 1). It also scales much better than them with an increase of the input image size (Figure 3).

Next, we conduct detailed ablations to study all the pieces of the proposed model and their effect on the performance of the model as well as the FLOPs needed for each method (Table 2). For each experiment, we modify one component of our main approach to verify its independent impact. When comparing to alternatives, our approach is able to produce more accurate results with lower or even FLOPs. More specifically:

In Table 2 (a), we first compare to the concatenation baseline, which is most commonly used in the literature e.g., (Wang et al., 2021; Dou et al., 2022; Lu et al., 2019; Su et al., 2019; Li et al., 2019; Lu et al., 2020; Chen et al., 2020). As seen, the reduction in compute for our method is significant, namely a 33% reduction in compute or in other words using 1.5x fewer FLOPs. This is also in conjunction with improved performance (bottom row). We note that such reduction in FLOPs is quite important for large vision-language models which take days to train, so it can save a lot of compute for such models. We also see that gating is the main contributor, and so we apply our method with gating which does not incur computational costs but brings improvements (Eq. 2).

Table 2 (b) and (c) provide an ablation of the number of tokens learned and iteration steps, respectively, showing a trade-off of spending more compute for higher accuracy, but with mostly diminishing returns. Note that here, the iterations occur after the initial resampling, in contrast to the Perceiver and Co-Tokenization, that iteratively resample the inputs.

Table 2 (d) illustrates that a single, latent cross-attention resampling of our approach gives both better performance and uses fewer FLOPs. This is in contrast to a spatial resampling used in prior works (Ryoo et al., 2021; Piergiovanni et al., 2022b).

Table 2: Ablation studies. These models are trained for 200,000 steps with a BS=512 to reduce the compute used. GF stands for GFLOPs. The highlighted row in each experiment indicates the setting selected and used in the other experiments here and the main experiments of the paper.

(a) **Cross-Attention Method**. Fusing text and image features.

	GF	GQA	SNLI-VE
Concat	58.4	78.9	77.4
Ours add	38.9	78.5	77.2
Ours gated	38.9	79.1	77.9

(b) **Number of Tokens** used in the model.

	GF	GQA	SNLI-VE
16	18.5	76.5	75.8
32	28.4	78.3	76.8
64	38.9	79.1	77.9
128	72.9	79.2	78.1

(c) **Number of Iterations** used to compute tokens.

	GF	GQA	SNLI-VE
1	34.2	78.3	77.1
2	35.5	78.8	77.6
4	38.9	79.1	77.9
8	42.5	79.2	77.6

(d) **Resampling Method** We find a latent cross-attention is better.

	GF	GQA	SNLI-VE
Spatial	42.5	78.9	77.4
Latent	38.9	79.1	77.9

(e) **Iterative Combine**. Combining features after each iteration.

	GF	GQA	SNLI-VE
None	38.9	78.1	76.5
Residual	38.9	78.7	77.6
Weighted	38.9	79.1	77.9

(f) **Number Layers** used in the fusion module.

	GF	GQA	SNLI-VE
8	22.4	76.7	74.2
16	30.5	78.3	75.4
32	38.9	79.1	77.9

Table 3: We find the proposed approach is more data efficient. Here, we compare pre-training with 3M samples (single dataset), and a mix of datasets/tasks with 12M samples.

	Method	GQA	SNLI-VE
3M Data	Concat	77.8	76.5
3M Data	Ours	78.6	77.2
12M Mixture	Concat	78.9	77.4
12M Mixture	Ours	79.1	77.9

Table 4: Comparison of pre-training sampling method. We find the adaptive sampling (our method) improves performance, and even works better with 100k than Uniform with 200k steps.

	PT Steps	GQA	SNLI-VE
Uniform	100k	78.1	76.7
Adaptive	100k	78.8	77.2
Uniform	200k	78.5	77.1
Adaptive	200k	79.1	77.9

Table 2 (e) provides insights regarding using the proposed weighting in the method (Eq. 3), which improves performance for the same FLOPs; (f) ablated the number of layers needed.

We note that in each of the ablations we are running the setting used in the main experiments of the paper, and per each experiment only one component is changed to understand the importance of specific component in a controlled setting. The grey rows indicate the main setting used. The ablations are done with the base-sized model and is trained to fewer steps.

Furthermore, we conduct ablations on the adaptive sampling and on how the adaptive sampling and Joint Adaptive Representation work together, specifically Tables 3 and 4 experiment with our adaptive sampling pre-training approach.

Table 3 evaluates the performance of the base model when a single pre-training dataset is used (here, CC3M) vs 12M mixture of tasks with adaptive sampling (here, CC3M+LN+OI+VG+subset of CC12M). We find that the proposed Joint Adaptive Representation is able to learn strong features, even with small data, showing its data efficiency. With respect to adaptive sampling, we also can see that on 4x smaller dataset it is closer to the performance of larger data without our approach. The last row indicates that both work better together and can better leverage the low data regimes.

Table 4 focuses on adaptive sampling vs uniform sampling of the same dataset and task mixture (here, we use the 12M dataset and tasks). The experiments are conducted until either 100k or 200k training steps. We see first that adaptive sampling is better in both cases. Further, we note that adaptive sampling (which is working in conjunction with Joint Adaptive Representation) is training faster, i.e., obtaining results at 100K steps which are better than uniform sampling at twice the steps.

Table 5: We outperform or perform competitively to the state-of-the-art models (SOTA), despite using very few FLOPs and small amounts of data. In fact with 40M training examples and with 39 GFLOPs our small model (350M params) outperforms all methods that have used less than a Billion examples for pre-training. Our large model further outperforms SimVLM on the popular VQA2.0 benchmark, still using 40M examples, and despite using the more challenging open-vocabulary evaluation. We are outperformed by 80B Flamingo on this benchmark. Models such as ALBEF and BLIP use about 14M sets but use have many more FLOPs. Some models are included not necessarily because they are comparable but because they provide useful context. *Our calculation of FLOPs.

	GFLOPs	Data	GQA	SNLI-VE	VQA2
Large-data Models					
Flamingo-80B (Alayrac et al., 2022)	-	2.3B+	-	-	**82.0**
SimVLM-Huge (Wang et al., 2021)	890*	1.8B	-	**86.21**	80.03
GIT (Wang et al., 2022)	-	800M	-	-	78.81
METER-CLIP-ViT Base** (Dou et al., 2022)	130*	404M	-	80.86	77.68
BLIP-L (Li et al., 2022)	250*	129M	-	-	78.25
Small-data Models					
FLAVA (Singh et al., 2022)	70*	70M	-	78.9	72.5
CFR (Nguyen et al., 2022)	-	-	73.6	-	69.8
VinVL (Zhang et al., 2021)	-	16M	65.05	-	75.95
BLIP (Li et al., 2022)	122*	14M	-	-	77.54
ALBEF (Li et al., 2021)	165*	14M	-	80.14	74.54
12-in-1 (Lu et al., 2020)	-	-	60.5	-	71.3
UNITER (Chen et al., 2020)	-	10M	-	79.39	72.5
LXMERT (Tan & Bansal, 2019)	-	6.5M	60.0	-	69.9
Ours-Base	38.9	40M	**81.9**	**82.1**	**79.20**
Ours	54.5	40M	**83.1**	**84.2**	**80.15**

6.3 Comparison to the State-of-the-Art (SOTA)

Table 5 shows the comparison with the state-of-the-art approaches. We see that our method performs competitively or outperforms prior models. Of note is that both our base and our larger model are actually the lowest FLOPs among contemporary models and outperforming models with many more FLOPs. Our small model (300M params) outperforms all SOTA approaches with the exception of extremely large models, Flamingo and SimVLM, both of which pre-training on very large datasets. Our main model further outperforms SimVLM on VQA2.0. We note that our models evaluate in the open-vocabulary setting which is more challenging. We note that there are many more vision and language learning approaches worth mentioning (Lu et al., 2019; Li et al., 2020; 2019; Huang et al., 2021), but are not included for space limitations. We also include one of the multi-task methods, the 12-in-1 (Lu et al., 2020) as one of the best performing ones. While multi-task training is a different regime, we include it here for context, rather than as a comparison point.

Scaling of our model further indicates promise, first by improving results, and secondly, by doing so at a very modest increase in the number of FLOPs, and still a very low overall FLOPs. Our scaled model outperforms SimVLM-Huge (Wang et al., 2021), a larger model which used extremely large 1.8B training data, on the popular VQA2.0 benchmark, even though we are evaluating in the open-vocabulary setting.

7 Conclusions

In this work we propose compact joint image-text representations, which provide efficient and data-efficient training for pre-trained vision-and-language models. The main idea is to adaptively and iteratively fuse the feature representations from both modalities. It greatly reduces the FLOPs used by the model, at the same time improving the overall accuracy. This also allows the model to scale without a large increase in FLOPs or memory requirements. In addition, we propose adaptive pre-training data sampling which further improves the data efficiency. We achieve new SOTA or competitive performance compared to much larger models, and do so with significantly less data.

8 REPRODUCIBILITY STATEMENT

For better reproducibility, we plan to open source the code with widely permissible license for everybody to use. The main proposed components of this work (Section 3.1 and Section 4.1) are both modular and an easy substitutions to the standard techniques used on prior works i.e., they can be substituted for the common concatenation of features, or for weighted to uniform sampling that are often used, respectively. Model, training and implementation details are provided in Section 5 and Section A in the Appendix. We also used well established and available in the literature backbones such as the ViT Base/32 (Dosovitskiy et al., 2021) for images and a T5 encoder and decoder (Roberts et al., 2022) for text.

9 ETHICS STATEMENT

The proposed approach is an alternative to present vision-language representation learning which is costly and requires large datasets and compute. The scaling of our model (which is at a small increase in FLOPs compared to others), confirmed that higher accuracies are obtained across all datasets. We anticipate that even further scaling is needed in order to outperform extremely large models trained on very large datasets. Despite the success of these methods, they incur very large costs, so it is desirable to combat that.

Vision-and-language models also might have pitfalls with regards to memorizing and/or propagating biases, making unfair statements or classifications and other potential issues. While we have used our models for evaluation purposes and trained and tested on publicly available datasets, such datasets are collected by automatic or semi-automatic processes, so special consideration should be applied before applying the models for other purposes.

REFERENCES

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL `https://arxiv.org/abs/2204.14198`.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *CoRR abs/1607.06450*, 2016.

Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *CVPR*, 2020.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

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

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

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. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021,*, 2021.

Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, et al. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18166–18176, 2022.

Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing out of the box: End-to-end pre-training for vision-language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12976–12985, 2021.

Drew A Hudson and Christopher D Manning. Gqa: a new dataset for compositional question answering over realworld images. In *CVPR*, 2019.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021.

Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? 2021.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. URL `https://arxiv.org/abs/1602.07332`.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *ICCV*, 128(7):1956–1981, 2020.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh D. Gotmare, Shafiq Joty, Caiming Xiong, and Steven C.H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

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

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020.

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

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *CVPR*, 2020.

Binh X. Nguyen, Tuong Do, Huy Tran, Erman Tjiputra, Quang D, and Anh Nguyen Tran. Coarse-to-fine reasoning for visual question answering. In *CVPR MULA Workshop*, 2022.

Duy-Kien Nguyen and Takayuki Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *CVPR*, 2018.

AJ Piergiovanni, Weicheng Kuo, and Anelia Angelova. Pre-training image-language transformers for open-vocabulary tasks. In *T4V: Transformers for Vision Workshop, Conference on Computer Vision and Pattern Recognition*, 2022a.

AJ Piergiovanni, Kairo Morton, Weicheng Kuo, Michael Ryoo, and Anelia Angelova. Video question answering with iterative video-text co-tokenization. *ECCV*, 2022b.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *ECCV*, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. Scaling up models and data with `t5x` and `seqio`, 2022. URL `https://arxiv.org/abs/2203.17189`.

Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. 2021.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami abd Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *arxiv.org/pdf/2112.04482.pdf*, 2022.

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

Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision, 2021. URL `https://arxiv.org/abs/2108.10904`.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. In *https://arxiv.org/abs/1901.06706*, 2019.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. In *arxiv.org/pdf/2110.02095.pdf*, 2022.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5579–5588, 2021.

A IMPLEMENTATION DETAILS

A.1 MODEL DETAILS

We use a ViT-Base/32 backbone (Dosovitskiy et al., 2021) for images and a T5 encoder and decoder (Roberts et al., 2022) for text. We use the standard 32,000-token vocabulary of T5.

Base model: The base model has a ViT-Base/32 backbone and T5 Base encoder and decoder. It has 350M parameters.

1.1B Model: The large model also uses the ViT-Base/32 backbone but a T5 Large encoder and decoder. It has 1.1B parameters.

Both of these models are trained from scratch (i.e., the VIT and T5 backbones are not pre-trained), and are only pre-trained jointly with our approach on the 40M public datasets we use in this paper.

The model is trained to minimize a per-token cross entropy loss over the 32,000 tokens of the vocabulary, both for pre-training and finetuning.

A.2 TRAINING, EVALUATION AND FINE-TUNING DETAILS

During pre-training, the model is trained with a batch size of 1024 for 500,000 steps. The learning rate is set to 1e-4 with 10,000 step linear warmup and cosine decay. The model is trained for 500k steps for the main experiments and for 200,000 steps for ablations.

For finetuning, we set the learning rate to 1e-5, train for 50,000 steps with a batch size of 64. For GQA, we train for 200,000 steps.

The model is trained on 224x224 image resolution during pretraining. In the most general cases we use the same resolution for fine-tuning, as well. One exception is the VQA2.0 dataset, where larger resolutions for fine-tuning, have been shown to be advantageous (Wang et al., 2021). Thus for this dataset, we fine-tune on 384 image size.

Data splits: Training and evaluation splits follow the ones established in the literature.

Evaluation metrics: The datasets considered use exact match accuracy based on the generated text tokens. Furthermore, we use an open-vocabulary generation output, which is more challenging than fixed vocabulary.

A.3 LIST OF PRE-TRAINING TASKS AND DATASETS

We use a relatively small sample of the publicly available image-language datasets, specifically, the Conceptual Captions (3M samples) (Sharma et al., 2018), the Conceptual Captions 12M dataset (12M samples) (Changpinyo et al., 2021), Visual Genome (Krishna et al., 2016), Open Images (Kuznetsova et al., 2020) and Localized Narratives dataset associated with Open Images (Pont-Tuset et al., 2020). The mixture of datasets is of about 40M examples.

We also use broader set of tasks, the benefits of which have been demonstrated in recent work. All tasks are formulated as image and text input and text output, More specifically the tasks are: the Masked Language Modeling (MLM) (Lu et al., 2019; Li et al., 2020; Chen et al., 2020; Tan & Bansal, 2019; Lu et al., 2019) (we mask random 15% of the text), Image-Text Matching (ITM) (Chen et al., 2020; Lu et al., 2019; Tan & Bansal, 2019; Lu et al., 2019), Captioning, and Caption Completion (Sharma et al., 2018), which is a generalized version of the Captioning task, where a portion of the caption is provided as an input and the remaining caption needs to be generated. We also use object-specific tasks for datasets which provide information about object presence, such as Open-Images (Kuznetsova et al., 2020), for example: 'Does object X exist', 'Does object X and Object Y exist', 'Does object X or object Y' 'List all objects'. For these tasks the ground truth answer is 'Yes'/,'No' or a list of class names; no localization information is required in the answer, even though some datasets might have it. Since all tasks are effectively following the sane input and output interface, and they easily share the same loss during training/pre-training.

B FLOPs AND MODEL PARAMETERS CALCULATION

The FLOPs and model parameters of ALBEF, METER, BLIP and FLAVA, etc were computed using their open source code and the THOP library https://github.com/Lyken17/pytorch-OpCounter. For other models we obtained information from the authors.