
Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, Desmond Elliott

University of Copenhagen, University of Cambridge, ETH Zürich, Tokyo Institute of Technology
emanuele@di.ku.dk, rcotterell@inf.ethz.ch, okazaki@c.titech.ac.jp, de@di.ku.dk

Abstract

Large-scale pretraining and task-specific fine-tuning is now the standard methodology for many tasks in computer vision and natural language processing. Recently, a multitude of methods have been proposed for pretraining vision and language BERTs to tackle challenges at the intersection of these two key areas of AI. These models can be categorized into either single-stream or dual-stream encoders. We study the differences between these two categories, and show how they can be unified under a single theoretical framework. We then conduct controlled experiments to discern the empirical differences between five vision and language BERTs. Our experiments show that training data and hyperparameters are responsible for most of the differences between the reported results, but they also reveal that the embedding layer plays a crucial role in these massive models.

1 Introduction

Learning generic multimodal representations from images paired with sentences is a fundamental step towards a single interface for vision and language (V&L) tasks. In pursuit of this goal, many pre-trained V&L models have been proposed in the last year, inspired by the success of pretraining in both computer vision (Sharif Razavian et al., 2014) and natural language processing (Devlin et al., 2019). All of these V&L models extend BERT (Devlin et al., 2019) to learn representations grounded in both modalities. They can either be classified as (i) single-stream, where images and text are jointly processed by a single encoder (e.g., Zhou et al., 2020), or (ii) dual-stream, where the inputs are encoded separately before being jointly modelled (e.g., Tan and Bansal, 2019).

The differences in downstream performance between single- and dual-stream models are currently unclear, with some papers claiming the superiority of one family over the other (Lu et al., 2019; Chen et al., 2020), while others arguing that it is hard to draw any conclusion (Qi et al., 2020).

The first goal of this paper is to understand the mathematical differences between single- and dual-stream models. Our analysis leads to a unified framework in which currently proposed architectures, both single- and dual-stream, are particular instances. We then implement several of the proposed encoders within this framework to empirically measure their differences in a controlled environment. We believe this comparative analysis is crucial to better understand and guide future research of massive models in this vibrant area of AI, ensuring progress is not blurred by confounds.

In fact, there are many differences in the protocols used to train V&L BERTs. In order to better understand these models, we conduct a series of controlled studies to investigate whether differences in downstream performance is explained by: (i) the amount of pretraining data and the pretraining objectives (e.g., Figure 2); (ii) the hyperparameters used to control the learning process; (iii) the variance caused by random initialization when pretraining (e.g., Figure 1); (iv) the variance due to fine-tuning multiple times on a downstream task; (v) being single- or dual-stream architectures; or (vi) the choice of the embedding layer.

In summary, our contributions in this paper are:

- We introduce a unified mathematical framework in which currently proposed V&L BERTs are only a subset of the possibilities.
- We release code for VOLTA (Vision-linguistic Transformer architectures), a PyTorch implementation of this framework in order to speed up research in multimodal pretraining.

\[\text{https://github.com/e-bug/volta.}\]
We conduct a series of controlled studies\footnote{https://github.com/e-bug/mpre-unmasked.} finding that several models perform similarly when trained under the same conditions.

While we find that single- and dual-stream families perform equally well, performance can differ significantly between two models and the embedding layer plays a key role.

However, these V&L BERTs are sensitive to weight initialization and state-of-the-art claims should not be made from single runs.

2 Vision-and-Language BERTs

Given a sequence of tokens \( \{w_1, \ldots, w_T\} \) and a set of visual features \( \{v_1, \ldots, v_K\} \), a shared goal of V&L BERT models is to produce cross-modal representations that are useful for downstream tasks grounded in both modalities.

In this section, we first review how these models embed their inputs to the feature space. Next, we discuss the main differences in the encoders and, finally, highlight a variety of confounds that might affect the performance achieved by these models.

2.1 Input Embeddings

Language Input All V&L BERTs adopt the approach of BERT: The input sequence is first tokenized into sub-word units (Wu et al., 2016; Sennrich et al., 2016) and two special tokens [CLS] and [SEP] are added to generate the text sequence \( \{[CLS], w_1, \ldots, w_T, [SEP]\} \). The embedding of each token is then given by the sum of three learnable vectors, corresponding to its form, position in the sequence, and segment (Devlin et al., 2019). In addition, VL-BERT (Su et al., 2020) also adds the visual feature of the entire image to each token.

Vision Input Typically, visual inputs are also very similar across all V&L BERTs. For a given image, a pretrained object detector is used to extract regions of interest, representing salient image regions. For each region, in addition to its feature vector, the object detector also returns the spatial location of its bounding box, which most V&L BERTs encode in different ways, analogously to the word position in the language modality. While most approaches present very similar ways to embed spatial locations, VL-BERT relies on a more complex geometry embedding and they are, instead, missing in VISUALBERT (Li et al., 2019). Some models also include a special feature [IMG] that denotes the representation of the entire image (e.g., a mean-pooled visual feature with a spatial encoding corresponding to the full image). Finally, PIXEL-BERT (Huang et al., 2020) does not rely on an object detector but directly extracts a set of visual embeddings from the raw image.

2.2 Encoders

Single-stream Encoders The majority of V&L BERTs follow the single-stream paradigm (Su et al., 2020; Li et al., 2019; Chen et al., 2020; Li et al., 2020a; Zhou et al., 2020; Lin et al., 2020; Li et al., 2020b). Here, a standard BERT architecture is given the concatenation of the visual and linguistic features of an image–text pair as input (Figure 3a). This design allows for an early and unconstrained fusion of cross-modal information.

Dual-stream Encoders ViLBERT (Lu et al., 2019), LXMERT (Tan and Bansal, 2019), and ERNIE-ViL (Yu et al., 2021)\footnote{ERNIE-ViL uses the dual-stream ViLBERT encoder.} are based on a
dual-stream paradigm. Here, the visual and linguistic features are first processed by two independent stacks of Transformer layers. The resulting representations are then fed into cross-modal Transformer layers where intra-modal interactions are alternated with inter-modal interactions (see Figure 3b and c). Interestingly, both ViLBERT and LXMERT modeled inter-modal interactions in the same way: Each stream first computes its query, key, and value matrices, before passing the keys and values to the other modality. By doing so, these models explicitly constrain interactions between modalities at each layer, inhibiting some of the interactions that are possible in a single-stream encoder while increasing their expressive power by separate sets of learnable parameters.

2.3 Pretraining Objectives

V&L BERTs are pretrained by jointly optimizing multiple different self-supervised objectives over tokens and image regions through (weighted) scalarization: \( \mathcal{L}(\theta) = \sum_{\alpha} \lambda_{\alpha} \mathcal{L}_{\alpha}(\theta) \). Here, \( \theta \) denotes a model’s parameters, \( \mathcal{L}_{\alpha} \) is the \( \alpha \)-th objective, and \( \lambda_{\alpha} \) is its corresponding weight. Commonly adopted objectives are of three types: language, vision, and cross-modal predictions.

For language prediction, BERT’s denoising masked language modeling (MLM) objective is typically used. MLM replaces some tokens with a \([\text{MASK}]\) symbol, which are then predicted by using bidirectional text context and image regions.

The MLM objective has been extended to image regions via masked region modeling objectives. These typically take the form of either object classification or feature regression, with some papers showing benefits when modeling both (e.g., Chen et al., 2020). Some models, such as LXMERT, are also optimized over objects’ attributes prediction.

Finally, interactions between the two modalities are explicitly enforced by means of cross-modal objectives. The typical task here is that of image–text matching (ITM; e.g., Chen et al., 2020), which extends BERT’s next sentence prediction objective to V&L inputs: Given a sequence of tokens and a set of image regions, the model is tasked to predict whether the tokens describe the image.

4In practice, ViLBERT directly feeds the image representations obtained from the object detector, while LXMERT further processes them through \( L_V \) layers.

2.4 Further Distinctions

So far, we have given an overview of the core components in V&L BERTs. However, there are several implementation differences between them.

For instance, LXMERT presents two main variations to the above description of dual-stream models. First, in its inter-modal layer, the parameters of the attention sub-layer are shared between the two streams. This results in the model learning a single function to contextualize image and text inputs, regardless of which modality plays the role of query or context. Second, its intra-modal layer only consists of the multi-head attention block. Moreover, a wider range of choices can affect the performance of these models. From the object detector used (and whether it is also fine-tuned during pretraining), to the number of image regions and the maximum text sequence length, to the number of layers and their hidden sizes, to pooling methods and fine-tuning MLP sizes, to the use of text-only data, to optimization hyperparameters (such as the number of pretraining epochs).

Another important distinction is the size and type of pretraining data, which can affect task performance (Figure 2). The size of pretraining datasets ranges from 3M–10M image–text pairs, over a range of pretraining tasks. The literature distinguishes between “in-domain” and “out-of-domain” data, each of which may consist of multiple datasets. An in-domain dataset overlaps with common downstream tasks, for example, using VQAv2 (Goyal et al., 2017) as both a pretraining task and a downstream task, while out-of-domain datasets have no expected overlap, for example, Conceptual Captions (Sharma et al., 2018).
3 A Unified Framework

In this section, we unify the recently proposed single-stream and dual-stream architectures under the same mathematical framework. We start by reviewing the Transformer layer, which forms the core of these architectures, then we explain how this layer has been adapted to encode multimodal data in V&L BERTs, and introduce a gated bi-modal Transformer layer that implements all of the architecture variants as special cases.

3.1 Transformer Layers

Transformer-based architectures consist of a stack of Transformer layers (Vaswani et al., 2017), each typically having a multi-head attention block (MAB) and a feed-forward block (FFB).

Multi-head Attention Block Given $N_q$ query vectors, each of dimension $d_q$, $Q \in \mathbb{R}^{N_q \times d_q}$, and $N_v$ key–value pairs $K \in \mathbb{R}^{N_v \times d_v}$, $V \in \mathbb{R}^{N_v \times d_v}$, an attention function $\text{Att}(Q, K, V)$ maps queries to output vectors with a scaled dot-product:

\[
\text{Att}(Q, K, V) = \omega(QKV^T)V
\]

where $\omega$ denotes a row-wise, scaled softmax: $\omega_i(\cdot) = \text{softmax}(\cdot / \sqrt{d_q})$. Here, $S = QK^T \in \mathbb{R}^{N_q \times N_v}$ is a score matrix that measures the similarity between each pair of query and key vectors. The output of Eq. (1) is a weighted sum of $V$, in which a value gets higher weight if its corresponding key has a larger dot product with the query.

Multi-head attention (MHA) extends this function by first projecting $Q, K, V$ into $H$ different matrices and computing the attention of each projection (Eq. (1)). These $H$ different output vectors are concatenated together (\{\}) and the concatenation is projected with a linear transformation $W^O$:

\[
\text{MHA}(Q, K, V) = [O_1 ; \ldots ; O_H]W^O,
\]

where $O_h = \text{Att} \left( QW_h^Q, KW_h^K, QW_h^K \right)$. (2)

Here, $\{W_h^Q, W_h^K, W_h^V\}^H_{h=1}$ and $W^O$ are learned parameters. Usually, $d_q = d_v = d$, $W^O \in \mathbb{R}^{d \times d}$, and $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{d \times d}$, where $d_a = d/H$.

Finally, given inputs $X, Y \in \mathbb{R}^{N \times d}$, a multi-head attention block is defined as:

\[
\text{MAB}(X, Y) = \text{LN}(X + \text{MHA}(X, Y, Y)),
\]

where LN is layer normalization (Ba et al., 2016).

Feed-forward Block For an input matrix $M \in \mathbb{R}^{N \times d}$, the feed-forward block is given by:

\[
\text{FFB}(M) = \text{LN}(M + \text{ReLU}(MW_1W_2)),
\]

where $W_1, W_2 \in \mathbb{R}^{d \times d_f}$ are learnable matrices.

Standard Transformer Layer Let $X \in \mathbb{R}^{N \times d}$ be an embedded input sequence, a standard Transformer layer performing self-attention is a parameterized function $f_{\theta}: \mathbb{R}^{N \times d} \rightarrow \mathbb{R}^{N \times d}$ such that:

\[
f_{\theta}(X) = \text{FFB}(\text{MAB}(X, X)).
\]

A stack of $L$ Transformer layers that encodes an input $X$, such as BERT, is then seen as a sequence of $L$ Transformer layers, each parametrized by $\theta_l$:

\[
\text{Encoder}(X) = f_{\theta_L} \circ \cdots \circ f_{\theta_1}(X).
\]

3.2 Single-stream Multimodal Transformers

Single-stream V&L BERTs extend BERT by concatenating the embedded visual inputs $X_V \in \mathbb{R}^{N_V \times d}$ and the embedded textual inputs $X_L \in \mathbb{R}^{N_L \times d}$ as a single input, hence the name ‘single-stream’ (Figure 3a). Specifically, $X = [X_L || X_V] \in \mathbb{R}^{N \times d}$, where $N = N_L + N_V$, and the attention is over both modalities (Figure 4a).

Hence, all single-stream models are of the type defined in the previous section: Encoder($X$). The various approaches only differ in the initial V&L embeddings, the pretraining tasks, and the training data.

3.3 Dual-Stream Multimodal Transformers

Both ViLBERT and LXMERT concurrently introduced inter-modal and intra-modal layers.

Inter-modal Transformer Layer The inter-modal layer explicitly models cross-modal interaction via a cross-modal attention module. Specifically, let $\mathcal{M} \in \{C, V\}$ denote either the linguistic ($\mathcal{L}$) or the visual ($\mathcal{V}$) modality, and \( \setminus \mathcal{M} \) its complementary one. The inter-modal multi-head attention for modality $\mathcal{M}$ is given by (Figure 3c):

\[
\mathbf{M}_{\setminus \mathcal{M}} = \text{MAB}(X_{\mathcal{M}}, X_{\setminus \mathcal{M}}).
\]
Figure 3: Visualization of the (a) single-stream, (b) dual-stream intra-modal, and (c) dual-stream inter-modal Transformer layers. (d) shows our gated bimodal layer. The inter-modal layer attends across modalities, while the intra-modal layer attends within each modality. Ours can attend to either or both.

Figure 4: Visualization of the score matrix for (a) single-stream, (b) text–text, (c) vision–vision, (d) text–vision, and (e) vision–text interactions. Shades of green denote the text modality, while purple ones denote the vision modality. Dual-stream scores are sub-matrices of the single-stream scores matrix.

Note that the second input to the multi-head attention block (Eq. (3)) is taken from the complementary modality, which means the keys $K$ and values $V$ in scaled dot-product attention (Eq. (1)) operate across modalities (see Figure 4d and e). The remainder of this layer follows as from Eq. (4).

**Intra-modal Transformer Layer** The intra-modal layer, on the other hand, is a Transformer layer computing the attention of each modality independently (see Figure 3b). For a modality $M$:

$$M_{\text{intra}} = \text{MAB}(\mathbf{X}_M, \mathbf{X}_M).$$

(8)

The rest of the layer follows as in Eq. (4) for ViLBERT, while there is no FFB block in LXMERT.

### 3.4 Dual-stream Attentions as Restricted Single-stream Attention

Recall that in single-stream models the input to a Transformer layer is the concatenation of both modalities, $\mathbf{X} = [\mathbf{X}_L \parallel \mathbf{X}_V]$. Therefore, in each single-stream attention head, the query representation is given by:

$$Q = \mathbf{X}W^Q = \begin{pmatrix} \mathbf{X}_L \\ \mathbf{X}_V \end{pmatrix} W^Q = \begin{pmatrix} Q_L \\ Q_V \end{pmatrix}$$

(9)

where $\begin{pmatrix} \cdot_L \\ \cdot_V \end{pmatrix}$ are the language and visual sub-matrices of the input and the resulting output. A similar expression also holds for the keys $K$ and values $V$. We note that the score matrix $S$ can be defined in terms of four sub-matrices (Figure 4a):

$$S = QK^T = \begin{pmatrix} Q_L & Q_V \\ K_L^T & K_V^T \end{pmatrix}$$

$$= \begin{pmatrix} Q_LK_L^T & Q_LK_V^T \\ Q_VK_L^T & Q_VK_V^T \end{pmatrix}$$

$$= \begin{pmatrix} S_{LL} & S_{LV} \\ S_{VL} & S_{VV} \end{pmatrix}$$

(10)

Recall from Eq. (1) that the attention matrix is a normalised score matrix $S$, so each single-stream layer computes both intra-modal (diagonal of $S$) and inter-modal attention (anti-diagonal of $S$). In other words, the dual-stream inter-modal and intra-modal attention functions act as restricted versions of the attention function in any single-stream layer (see Figure 4). As a result, by interleaving inter- and intra-modal layers, dual-stream models introduce an *inductive bias* towards which interactions the model enforces in each layer.

Note that for this to be exact, the learnable parameters of the MHA function need to be shared between modalities (as done, for example, by LXMERT in its inter-modal blocks).
3.5 Gated Bimodal Transformer Layers

In the previous section, we showed that single-stream attention blocks capture both the inter-modal and intra-modal interactions, separately modeled by dual-stream architectures. We now introduce a general gated bimodal Transformer layer (Figure 3d), in which both single- and dual-stream layers are special cases. By doing so, we can define existing V&L BERTs within a single architecture, which allows us to implement and evaluate several of these models in a controlled environment (see next sections). In addition to textual $X_L$ and visual embeddings $X_V$, this layer takes a set of fixed binary variables $\{\gamma, \tau\}$ as part of its input: $\gamma = \{\gamma_{LV}, \gamma_{VL}, \gamma_{LL}, \gamma_{VV}\}$, and $\tau = \{\tau_{MHA}, \tau_{LN}, \tau_{FF}\}$. The $\gamma$ values act as gates that regulate the cross-modal interactions within a layer, while the $\tau$ values control whether the parameters are tied between modalities.

The main difference in our gated layer is in its attention functions, originally defined in Eq. (1) as gates that regulate the cross-modal interactions, allowing us to define existing architectures within a unified mathematical framework. We can recover an inter-modal block (Eq. (7)) by setting $\gamma_{LV} = \gamma_{VL} = 0$ and $\gamma_{LL} = \gamma_{VV} = 1$. Similarly, the single-stream block (Eq. (3)) can be recovered by setting $\gamma = 0$ and tying the learnable parameters ($\tau = 1$) between the two streams (e.g., $W_L = W_L^Q = W_V^Q$ in each attention head).

Furthermore, the gated bimodal Transformer layer allows us to model a superset of the few combinations considered thus far for cross-modal fusion by multimodal transformer encoders. One may explore asymmetric streams in which the two modalities interact differently with the bimodal inputs, or explore different ways of interleaving conventional single- and dual-stream blocks, or even different levels of parameter sharing. For example, asymmetric vision-and-language layers might be beneficial for navigation (e.g., Hill et al., 2021) or language-conditioned image generation (e.g., Cho et al., 2020). An exploration of these possibilities is left for future work.

4 Experimental Setup

In this section, we present the experimental setup for our controlled studies on V&L encoders. **VOLTA** In order to facilitate research and development of V&L pretraining, we release VOLTA (Visiolinguistic Transformer architectures), an implementation of our unified framework in PyTorch (Paszke et al., 2019). Our code is built on top of the ViLBERT-MT repository,\(^7\) based on PyTorch-Transformers, due to its support to a wide range of V&L tasks. We stress that it is important, for this study, to have a unified implementation that allows us to remove possible confounds due

\(^6\)In practice, our implementation is efficient and does not evaluate sub-matrices whose corresponding gate is set to 1.

\(^7\)https://github.com/facebookresearch/vilbert-multi-task/.
to implementation details and effectively measure differences given by the proposed architectures.

**Implementation Details**  V&L BERTs typically extract image features using a Faster R-CNN (Ren et al., 2015) trained on the Visual Genome dataset (VG; Krishna et al., 2017), either with a ResNet-101 (He et al., 2016) or a ResNeXT-152 backbone (Xie et al., 2017). The number of features varies from 10 to 100. Our models are trained with 36 regions of interest extracted by a Faster R-CNN with a ResNet-101 backbone (Anderson et al., 2018). Each model is initialized with the parameters of BERT, following the approaches described in the original papers. Randomly initialized weights are initialized following the standard approach in PyTorch-Transformers (on which these models are built): Fully-connected and embedding layers are initialized from a normal distribution with mean 0.0 and standard deviation 0.02, bias vectors are initially set to 0.0, and the Layer Normalization weight vector to 1.0. We train all models on 4 NVIDIA P100 GPUs and rely on gradient accumulation to obtain larger batches when needed. The parameter sets giving the best validation performance based on the pretraining objective are used for downstream tasks.

**Pretraining**  As discussed in §2.4, V&L BERTs have been pretrained on datasets of varying size and type. In this paper, we pretrain all of our models on the Conceptual Captions dataset (CC; Sharma et al., 2018), which consists of 3.3M images with weakly associated captions automatically collected from billions of Web pages. This stands in contrast to other datasets, for example, COCO (Lin et al., 2014) or VQA (Antol et al., 2015), where the images are strongly associated with crowdsourced captions or question–answer pairs. The CC dataset is a good candidate for learning generic multimodal representations because of its size, that it was scraped from the Web, and that it has a broad coverage of subject matter. Note that due to broken links, and a subsequent pruning phase, where images also found in the test sets of common V&L tasks are removed, we pretrain all our models on 2.77M image–caption pairs from Conceptual Captions.

**Downstream Evaluation Tasks**  We consider the most common tasks used to evaluate V&L BERTs, spanning four groups: vocab-based VQA (Goyal et al., 2017; Hudson and Manning, 2019), image–text retrieval (Lin et al., 2014; Plummer et al., 2015), referring expression (Kazemzadeh et al., 2014; Mao et al., 2016), and multimodal verification (Suhr et al., 2019; Xie et al., 2019). See Table 1 for details. For each model, the parameter set giving the best performance in the validation set was used for test.

**5 Results**  We perform carefully controlled experiments to investigate the possible reasons for the reported difference in performance between V&L BERTs.

**5.1 Unified Data and Reimplementation**  We start by examining the performance of V&L BERTs pretrained on the same 2.7M CC dataset. Recall from Figure 2 that V&L BERTs have been pretrained on different combinations of datasets, which may explain most of the claimed differences in downstream task performance. Here, we evaluate three models with official released code: Vl.BERT, LXMERT, and VL-BERT.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image Source</th>
<th>Train</th>
<th>Test</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQAv2</td>
<td>COCO</td>
<td>655K</td>
<td>448K</td>
<td>VQA-score</td>
</tr>
<tr>
<td>GQA</td>
<td>COCO+Flickr</td>
<td>1.1M</td>
<td>12.6K</td>
<td>Accuracy</td>
</tr>
<tr>
<td>RefCOCO+</td>
<td>COCO</td>
<td>120K</td>
<td>10.6K</td>
<td>Accuracy</td>
</tr>
<tr>
<td>RefCOCOg</td>
<td>COCO</td>
<td>80K</td>
<td>9.6K</td>
<td>Accuracy</td>
</tr>
<tr>
<td>NLVR2</td>
<td>Web Crawled</td>
<td>86K</td>
<td>7K</td>
<td>Accuracy</td>
</tr>
<tr>
<td>SNLI-VE</td>
<td>Flickr</td>
<td>529K</td>
<td>17.9K</td>
<td>Accuracy</td>
</tr>
<tr>
<td>COCO</td>
<td>COCO</td>
<td>567K</td>
<td>1K</td>
<td>Recall@1</td>
</tr>
<tr>
<td>Flickr30k</td>
<td>Flickr</td>
<td>145K</td>
<td>1K</td>
<td>Recall@1</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the downstream V&L tasks.

---

8 Only Tan and Bansal (2019) reported slightly better performance when pretraining from scratch but they relied on large corpora of in-domain, human-annotated data.
9 VL-BERT also adds text-only data to avoid overfitting on short and simple sentences typical of V&L datasets.
10 We also expect this type of dataset will be easier to collect for low-resource languages in the future.
11 The datasets listed in Table 1, Visual 7W (Zhu et al., 2016), RefCOCO (Kazemzadeh et al., 2014), GuessWhat (de Vries et al., 2017), and VCR (Zellers et al., 2019).
12 Following previous work, accuracy in referring expression is evaluated on the region proposals of Yu et al. (2018).
13 Vl.BERT was trained as described in Lu et al. (2020).
Figure 5: Unified data and reimplementation results. Performance of selected V&L BERTs on multiple tasks from the original papers (♦), and when pretrained on 2.7M Conceptual Captions with their official code (□) or in VOLTA (◦).

Same Data, Similar Performance  Figure 5 shows the results of controlling the pretraining data and pretraining tasks. The results from the papers are reported (♦), alongside our training of these models using the official code (□). There is a drop in performance for the models we trained on the VQAv2, NLVR2, and image retrieval tasks, compared to the performance reported in the papers. This is not surprising given that the □ models were pretrained on less data than the papers. In particular, given that ViLBERT was also pretrained on CC but with more image–text pairs, our results corroborate previous studies showing diminishing returns with pretraining data size (e.g., Lu et al., 2019; Li et al., 2020a). However, the claimed performance gaps between these models narrows when they are pretrained on the same data. For instance, according to the literature, LXMERT was clearly the best model in VQA tasks, which is likely due to its use of large, in-domain data and a VQA pretraining objective.14

VOLTA Implementation  We also implemented these models in VOLTA and trained them using their official procedures and hyperparameters. Figure 5 shows that the performance of each of these models (◦) closely follows the official implementations in these downstream tasks, confirming the correctness of our framework. There are, however, some larger differences for some of the tasks: In VQAv2, we now see that ViLBERT performs slightly worse than the other models (contrarily to what we obtained with the official code), and in GQA, LXMERT closes the gap with ViLBERT. ViLBERT’s performance on NLVR2 and COCO image retrieval increases by 2–3 points in the VOLTA framework. As VOLTA is based on the ViLBERT code base, these differences might be due to weight initialization, an hypothesis that we test in later sections.

With this first study, we have seen that the performance of these V&L BERTs is similar when they are trained on the same data. Moreover, we demonstrated the correctness of our implementations in VOLTA, in which these models are built following the unified framework introduced in §3. Nevertheless, there are still many possible confounds in the training procedures adopted by these models that might interfere with a fair comparison of these architectures. In the next section, we control these variables to unmask the true gains introduced by a number of multimodal encoders.

5.2 Controlled Setup

We define a fixed set of hyperparameters to evaluate ViLBERT, LXMERT, VL-BERT, VISUAL-BERT, and UNITER on four downstream tasks: VQAv2, RefCOCO+, NLVR2, and Flickr30K.

- **Inputs:** Each model used a different maximum number of tokens and LXMERT did not have an overall [IMG] feature. We fix the same maximum number of tokens and add the [IMG] feature to each architecture.

- **Encoders:** We noticed that ViLBERT used higher dimensional representations for the visual stream. We fix the same dimension as in the linguistic stream for a comparison that is fairer comparison against LXMERT, and more intuitive with the single-stream models.

- **Pooling:** While VL-BERT is the only architecture that does not have a pooling layer, other V&L BERTs use it for the image–text matching objective. We fix the models to use multiplicative pooling (Lu et al., 2019) for all the models in order to separately learn
Table 2: Results with our controlled setup. Each model is pretrained using the VOLTA framework with the same fixed hyperparameters on the 2.7M CC dataset, and fine-tuned on downstream tasks.

<table>
<thead>
<tr>
<th>Model</th>
<th>VQA v2</th>
<th>RefCOCO+</th>
<th>NLVR2</th>
<th>Flickr30k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>test-dev</td>
<td>test dev</td>
<td>test P</td>
<td>test IR</td>
</tr>
<tr>
<td>ViLBERTBASE</td>
<td>68.7</td>
<td>71.4</td>
<td>72.4</td>
<td>59.8</td>
</tr>
<tr>
<td>LXMERT</td>
<td>67.1</td>
<td>68.8</td>
<td>69.1</td>
<td>50.4</td>
</tr>
<tr>
<td>VL-BERTBASE</td>
<td>68.3</td>
<td>71.1</td>
<td>72.6</td>
<td>57.9</td>
</tr>
<tr>
<td>VisualBERT</td>
<td>68.2</td>
<td>69.7</td>
<td>71.3</td>
<td>61.1</td>
</tr>
<tr>
<td>UNITERBASE</td>
<td>68.8</td>
<td>71.9</td>
<td>72.9</td>
<td>60.9</td>
</tr>
</tbody>
</table>

Results Table 2 shows the results of our controlled study. First, we note that the performance of ViLBERT and VL-BERT is similar compared to training with their original hyperparameters. In fact, VQA v2 performance improves for ViLBERT, showing that dual-stream models do not require different sizes in the two streams. VL-BERT also performs similarly to its official setup, showing the additional ITM pretraining objective in our controlled setup does not hurt downstream task performance (contrarily to the results reported in their paper). We do, however, note that LXMERT performs worse on NLVR2 and VQA v2 in our controlled setup than with its original hyperparameters, suggesting that LXMERT may require more pretraining steps to converge. Overall, the results show that most of the examined models perform similarly in our controlled setup, compared to the official setups.

5.3 Fine-tuning Variance

We now turn our attention to the effect of fine-tuning variance on task performance. It has been observed that the fine-tuning of BERT is sensitive to randomness in initialization and data ordering (Dodge et al., 2020). Here, we investigate the sensitivity of the five models used in the controlled study. We fine-tune each model 10 times on the RefCOCO+ and NLVR2 tasks by varying the seed. This changes training data order and the weight initialization of the classification layer. Figure 7 shows violin plots of the distribution of results, in which the dots represent the experimental observations. We also report an average standard deviation of 0.3 points for these models across both tasks. However, the minimum and the maximum scores of a given model often differ by 1 or more points, showing how a single fine-tuning run of these models can lead to incorrect conclusions.

5.4 Pretraining Variance

In the previous section, we found substantial variance in the performance of V&L BERTs across 10 fine-tuning runs. We now investigate if the pretraining phase is similarly affected by different runs. Here, each model in our controlled setup is pretrained 10 times and fine-tuned once on four tasks: VQA v2, RefCOCO+, NLVR2, and Flickr30K image–text retrieval. By varying the seed, we modify training data order as well as all the layers that are not initialized from BERT (e.g., the visual embeddings, the masked object classification head and the ITM head in single-stream models). Figure 6 shows violin plots for each task. We start by noting that our first pretraining run (Table 2) of LXMERT was the worst one (its text retrieval recall on Flickr30K is 10 points lower than its mean). We also confirm that LXMERT has slower convergence rate, with its task performance after 10 epochs showing the largest variance among the V&L BERTs we tested. On the other hand, we find that some of these architectures are less prone to variance caused by pretraining.
Figure 6: Pretraining variance of V&L BERTs. Each model is pretrained 10 times and fine-tuned once.

Figure 7: Fine-tuning variance of V&L BERTs on RefCOCO+ and NLVR2. Each model is pretrained once and fine-tuned 10 times on each task.

Figure 8: Variance of V&L BERTs on the Constrastive Set of NLVR2, when each model is pretrained 10 times and fine-tuned once (a), or pretrained once and fine-tuned 10 times (b).

5.5 Evaluating Local Decision Boundaries

Previous work has shown that state-of-the-art systems can exploit systematic gaps in the data to learn simple decision rules that let them achieve high performance on test data (Gururangan et al., 2018; Geva et al., 2019; Ribeiro et al., 2019). In an effort to more accurately estimate model performance, Gardner et al. (2020) proposed contrast sets: datasets in which existing test instances have small but label-changing modifications in order to characterize the correct decision boundary near them. Figure 8 shows the performance of our analyzed models on the NLVR2 contrast set. Similar to Gardner et al. (2020), we see that LXMERT loses around 15 points when evaluated on perturbed samples. Furthermore, models that performed much better on the standard test set now achieve comparable performance to LXMERT, showing that they exploited systematic gaps. That is, all of these V&L BERTs would perform similarly when evaluated on out-of-distribution data.

5.6 Single- or Dual-stream Architectures

One of the key design choices that distinguishes V&L BERTs is the number of “streams” used by the encoder to process visual and linguistic inputs. Lu et al. (2019) showed how their single-stream baseline performed worse than their dual-stream ViLBERT architecture, while Chen et al. (2020) claimed single-stream UNITER outperformed ViLBERT. Our controlled study across several tasks and different pretraining initializations allows us to provide an answer grounded with statistical tests. To do so, we split the models in dual- and single-stream architectures\(^\text{17}\) and run a one-way ANOVA (Table 3). After Bonferroni correction, we only find statistical difference at \(p < 0.005\) (Benjamin et al., 2018) between these two groups for the Flickr30K text retrieval task.

\(^\text{17}\)We only consider ViLBERT for dual-stream encoders due to LXMERT’s sub-optimal performance in our setup.
Table 3: ANOVA between single- and dual-stream architectures (left) and between all the tested V&L BERTs (right). * denotes significant results at $p < 0.005$ after Bonferroni correction.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Single/Dual Stream</th>
<th>V&amp;L BERTs</th>
<th>V&amp;L BERTs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F-test</td>
<td>p-value</td>
<td>F-test</td>
</tr>
<tr>
<td>VQAv2</td>
<td>11.40</td>
<td>1.7e-03</td>
<td>12.75</td>
</tr>
<tr>
<td>RefCOCO+</td>
<td>0.10</td>
<td>7.6e-01</td>
<td>111.61</td>
</tr>
<tr>
<td>NLVR2</td>
<td>8.28</td>
<td>6.5e-03</td>
<td>13.41</td>
</tr>
<tr>
<td>Flickr30k IR</td>
<td>9.64</td>
<td>3.0e-03</td>
<td>13.27</td>
</tr>
<tr>
<td>Flickr30k TR</td>
<td>31.14</td>
<td>2.0e-06*</td>
<td>29.74</td>
</tr>
</tbody>
</table>

5.8 The Importance of the Embeddings

Finally, our controlled setup leads us to an interesting finding: The embedding layer (§2.1) plays a crucial role in the final performance of V&L BERTs. In fact, the only difference among VILBERT, VISUALBERT, and UNITER in our setup is their embedding layer. Figure 6 and Figure 7 show that this can have a drastic impact on the downstream performance, although the literature has given little attention to this detail. For instance, Chen et al. (2020) claim that the main contribution of UNITER is the set of pretraining tasks, while our results, wherein all the models are trained on the same pretraining tasks, highlight that their embedding layer is an important confound on final performance. Interestingly, VISUALBERT is the only model that does not encode the locations of regions of interest in its embeddings. This leads it to considerably lower performance on RefCOCO+, showing that this information is extremely useful for this task.

Given this result, we conduct one additional experiment to see whether the embedding layer biased our conclusion for dual- and single-stream performance. To test this, we swap the embedding layers of VILBERT (best dual-stream) and UNITER (overall better single-stream) with each other, which we pretrain and fine-tune once (Figure 10). Similar to our previous results, embeddings are especially important for the tasks of referring expression and retrieval. However, no single embedding layer performs better, corroborating that dual- and single-stream architectures perform on par and showing that different embedding strategies are necessary to maximise performance in these two families of V&L BERTs.

5.8 Limitations

All the experiments in this paper are limited to models that use a specific type of pretrained and frozen visual encoder. While most V&L BERTs follow this paradigm, some studies find beneficial to jointly learn the visual encoder with language (Su et al., 2020; Huang et al., 2020; Radford et al., 2021; Kim et al., 2021). In addition, we only consider base architecture variants (initialized with BERT$_{BASE}$) and pretrained on CC. Studying the effects of visual encoders, pretraining data and larger models is left as future work.

Although we expect longer pretraining would be beneficial for every model, in our controlled setup, we pretrain each model for 10 epochs to reduce resource consumption. Here, we also constrain our hyperparameter search over a small grid of values that have been used in the literature. Finally, we leave a thorough, controlled study of the various pretraining objectives to future work.

6 Reproducibility and the Environment

From the perspective of reproducible research, there are several advantages to using the VOLTA framework for V&L encoders. First, VOLTA reduces confounds due to differences in implementations, while also enabling fair comparisons with related work. Second, visual and textual data only need to be preprocessed once instead of creating model-specific formats for every V&L BERT.

From a financial perspective, the costs involved in pretraining hampers contributions from many academic institutions and deters the evaluation of multiple trained models, which we showed
7 Conclusion

We introduced and implemented a unified mathematical framework, under which recently proposed V&L BERTs can be specified as special cases. We conducted a series of controlled studies within this framework to better understand the differences between several models. We found that the performance of the considered models varies significantly due to random initialization, in both pretraining and fine-tuning. We also found that these models achieve similar performance when trained with the same hyperparameters and data. Notably, some models outperform others but we found that (a) single- and dual-stream model families are on par, and (b) embedding layers play a crucial role towards a model’s final performance.

Our fast-paced field rewards the contribution of new methods and state-of-the-art results (Rogers and Augenstein, 2020), which often contrasts with controlled comparisons and training multiple models for variance estimation. In this paper, we showed that several methods for vision-and-language representation learning do not significantly differ when compared in a controlled setting. This finding echoes similar studies of variants of LSTMs (Greff et al., 2017) and Transformers (Narang et al., 2021) that are not significantly better than the original models. Looking to the future, we recommend that new V&L BERTs are pretrained on similar datasets, and that researchers report fine-tuning variance, in addition to their best performing model. We hope that our findings will encourage more controlled evaluations of newly proposed architectures for vision-and-language and beyond.

Acknowledgments

We are grateful to the action editor Jacob Eisenstein and the anonymous reviewers at TACL for their constructive comments and discussions. This project has received funding from the...
European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 801199 and by “Research and Development of Deep Learning Technology for Advanced Multilingual Speech Translation,” the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan.

References


Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. 2020. ImageBERT: Cross-modal pre-training with large-scale...


