

# ManagerTower: Aggregating the Insights of Uni-Modal Experts for Vision-Language Representation Learning

Xiao Xu<sup>1,3\*</sup>, Bei Li<sup>2,3</sup>, Chenfei Wu<sup>3</sup>, Shao-Yen Tseng<sup>4</sup>, Anahita Bhiwandiwalla<sup>4</sup>,  
Shachar Rosenman<sup>4</sup>, Vasudev Lal<sup>4</sup>, Wanxiang Che<sup>1†</sup>, Nan Duan<sup>3†</sup>

<sup>1</sup>Harbin Institute of Technology, Harbin, China, <sup>2</sup>Northeastern University, Shenyang, China

<sup>3</sup>Microsoft Research Asia, <sup>4</sup>Intel Labs, Cognitive Computing Research

{xxu, car}@ir.hit.edu.cn, libei\_neu@outlook.com

{chenfei.wu, nanduan}@microsoft.com, shao-yen.tseng@intel.com

{anahita.bhiwandiwalla, shachar.rosenman, vasudev.lal}@intel.com

## Abstract

Two-Tower Vision-Language (VL) models have shown promising improvements on various downstream VL tasks. Although the most advanced work improves performance by building bridges between encoders, it suffers from ineffective layer-by-layer utilization of uni-modal representations and cannot flexibly exploit different levels of uni-modal semantic knowledge. In this work, we propose ManagerTower, a novel VL model architecture that gathers and combines the insights of pre-trained uni-modal experts at different levels. The managers introduced in each cross-modal layer can adaptively aggregate uni-modal semantic knowledge to facilitate more comprehensive cross-modal alignment and fusion. ManagerTower outperforms previous strong baselines both with and without Vision-Language Pre-training (VLP). With only 4M VLP data, ManagerTower achieves superior performances on various downstream VL tasks, especially 79.15% accuracy on VQAv2 Test-Std, 86.56% IR@1 and 95.64% TR@1 on Flickr30K. Code and checkpoints are available at <https://github.com/LooperXX/ManagerTower>.

## 1 Introduction

In recent years, there has been a growing interest in the field of Vision-Language (VL) representation learning due to the development of Vision-Language Pre-training (VLP) techniques. VLP aims to learn transferable multi-modal knowledge from large-scale image-text pairs, which can further improve the performance of various downstream VL tasks, such as visual question answering (Goyal et al., 2017), visual entailment (Xie et al., 2019), visual reasoning (Suhr et al., 2019), and image-text retrieval (Young et al., 2014).

Visual and textual modalities in VL models are typically processed by uni-modal encoders and subsequently fused in a cross-modal encoder. This

\*Contribution during internship at Microsoft.

†Contact Person

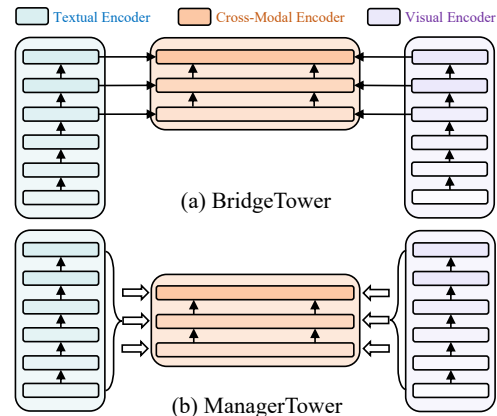


Figure 1: Brief illustrations of BridgeTower and ManagerTower. Hollow arrows indicate the transmission of multi-layer uni-modal representations in ManagerTower instead of layer-by-layer transmission in BridgeTower.

general architecture can be referred to as the Two-Tower architecture. METER (Dou et al., 2022) and BridgeTower (Xu et al., 2022) are two representative Two-Tower VL models. METER uses CLIP-ViT (Radford et al., 2021) and RoBERTa (Liu et al., 2019b) as pre-trained uni-modal encoders, but it ignores different levels of uni-modal semantic knowledge in them and only feeds the last-layer outputs of each uni-modal encoder into the cross-modal encoder. In an effort to address this issue, as illustrated in Figure 1(a), BridgeTower connects multiple top uni-modal layers with each cross-modal layer in a layer-by-layer fashion to exploit uni-modal semantic knowledge at different levels.

In this work, we build upon the research of BridgeTower and advance it in two aspects. Specifically, we address the limitations of BridgeTower: (i) its layer-by-layer utilization of different uni-modal layer representations is ineffective. Each cross-modal layer can only utilize an artificially-connected uni-modal layer representation, thus restricting the exploitation of different levels of uni-modal semantic knowledge. (ii) the number of cross-modal layers is tied to the number of uni-

modal layer representations it used, thus limiting its scalability and capability. For example, increasing the number of uni-modal layer representations used requires a corresponding increase in the number of cross-modal layers. This leads to an increase in the number of parameters and computation cost, while does not always result in performance improvements as demonstrated by Xu et al. (2022).

As shown in Figure 1(b), we propose a novel VL model architecture, ManagerTower, that aggregates multi-layer uni-modal representations via managers in each cross-modal layer. Each manager takes multi-layer uni-modal representations as the **insights** of pre-trained uni-modal **experts** at different levels, and then **adaptively** aggregates them to facilitate more comprehensive cross-modal alignment and fusion. More concretely, inspired by the linear combination of layers (Wang et al., 2019) method, we adapt it as the Static Aggregation of Experts (SAE) manager and then remove redundant information to design the Static Aggregation of Uni-modal Experts (SAUE) manager, which focuses on aggregating uni-modal semantic knowledge. We further propose the Adaptive Aggregation of Uni-modal Experts (AAUE) manager to adaptively aggregate multi-layer uni-modal representations for each token in different cross-modal layers. Moreover, in principle, managers can be easily integrated into any cross-modal encoders and work well with any uni-modal encoders, making ManagerTower scalable and flexible.

We first explore the feasibility of various designs of managers by evaluating and analyzing the performance on VQAv2 and Flickr30K datasets. Then, we pre-train ManagerTower with commonly used 4M VLP data and evaluate it on various downstream VL tasks. With the same pre-training and fine-tuning settings and uni-modal backbones as previous strong baselines such as METER and BridgeTower, ManagerTower achieves superior performances on various downstream VL tasks, especially 79.15% accuracy on VQAv2 Test-Std, 86.56% IR@1 and 95.64% TR@1 on Flickr30K. It outperforms not only many base-size models pre-trained on 4M data but also some models pre-trained on more data and/or with larger size.

## 2 Preliminary

In this work, for a fair comparison with METER and BridgeTower, we use the same cross-modal encoder and pre-trained uni-modal encoders.

### 2.1 Visual Encoder

CLIP-ViT, the visual encoder of CLIP (Radford et al., 2021), has been widely used in VL models (Shen et al., 2021; Dou et al., 2022). It reshapes each input image into a flattened patch sequence and prepends a [class] token to the sequence. After a linear projection, position embeddings are added to the sequence to get the input visual representation  $\mathbf{V}_0$ . The  $\ell^{\text{th}}$  visual layer representation can be computed as:  $\mathbf{V}_\ell = \text{Encoder}_\ell^{\text{V}}(\mathbf{V}_{\ell-1})$ ,  $\ell = 1 \dots L_V$ , where  $\ell$  is the layer index and  $L_V$  is the number of layers of the visual encoder.

### 2.2 Textual Encoder

RoBERTa (Liu et al., 2019b) is widely used in the field of VL (Dou et al., 2022; Li et al., 2022b) due to its robust performance. It tokenizes the input text with the byte-level Byte-Pair Encoding (BPE) (Sennrich et al., 2016; Radford et al., 2019) and adds [ <s> ] and [ </s> ] tokens to the start and end of the sequence, respectively. Then, it applies word embeddings and positional embeddings to the tokenized sequence to get the input textual representation  $\mathbf{T}_0$ . Similarly, the  $\ell^{\text{th}}$  textual layer representation can be computed as:  $\mathbf{T}_\ell = \text{Encoder}_\ell^{\text{T}}(\mathbf{T}_{\ell-1})$ ,  $\ell = 1 \dots L_T$ , where  $L_T$  is the number of layers of the textual encoder.

### 2.3 Cross-Modal Encoder

We adopt the transformer encoder (Vaswani et al., 2017) with the co-attention mechanism as the cross-modal encoder (Lu et al., 2019). For each cross-modal layer, each modality has a multi-head self-attention (MSA) block, a multi-head cross-attention (MCA) block, and a feed-forward (FFN) block. The MCA block allows the visual part of the cross-modal encoder to attend to the textual part and vice versa. Each cross-modal layer is denoted as  $\text{Encoder}_\ell^{\text{C}}$ ,  $\ell = 1 \dots L_C$ , where  $L_C$  is the number of cross-modal layers. For brevity, the  $\ell^{\text{th}}$  cross-modal layer computes as:

$$\tilde{\mathbf{C}}_\ell^{\text{V}} = \mathbf{C}_{\ell-1}^{\text{V}}, \quad (1)$$

$$\tilde{\mathbf{C}}_\ell^{\text{T}} = \mathbf{C}_{\ell-1}^{\text{T}}, \quad (2)$$

$$\mathbf{C}_\ell^{\text{V}}, \mathbf{C}_\ell^{\text{T}} = \text{Encoder}_\ell^{\text{C}}(\tilde{\mathbf{C}}_\ell^{\text{V}}, \tilde{\mathbf{C}}_\ell^{\text{T}}), \quad (3)$$

where  $\mathbf{C}_\ell^{\text{V}}, \mathbf{C}_\ell^{\text{T}}$  are the output representations of the visual and textual part at the  $\ell^{\text{th}}$  layer,  $\tilde{\mathbf{C}}_\ell^{\text{V}}, \tilde{\mathbf{C}}_\ell^{\text{T}}$  are inputs of each part.  $\mathbf{C}_0^{\text{V}}, \mathbf{C}_0^{\text{T}}$  are initialized with the last-layer representations from uni-modal encoders:  $\mathbf{C}_0^{\text{V}} = \mathbf{V}_{L_V} \mathbf{W}_V$ ,  $\mathbf{C}_0^{\text{T}} = \mathbf{T}_{L_T} \mathbf{W}_T$ , where  $\mathbf{W}_V, \mathbf{W}_T$

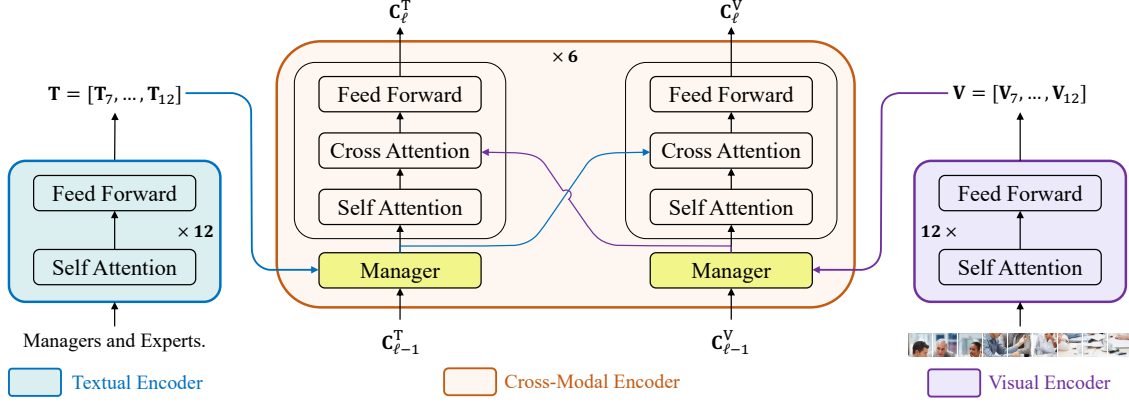


Figure 2: An illustration of ManagerTower, a textual manager and a visual manager are introduced in each cross-modal layer. Top  $N=6$  uni-modal layer representations  $\mathbf{T}, \mathbf{V} \in \mathbb{R}^{N \times L \times D}$  and output representations of the previous cross-modal layer  $\mathbf{C}_{\ell-1}^T, \mathbf{C}_{\ell-1}^V, \ell = 1 \dots 6$  are fed into the textual manager  $\mathcal{M}_{\ell}^T$  and visual manager  $\mathcal{M}_{\ell}^V$ , respectively.  $N$  is the number of pre-trained uni-modal experts we used,  $L$  is the length of the input sequence.

are linear cross-modal projections. In this work, we use the same default setting as BridgeTower for a fair comparison:  $L_V = L_T = 12, L_C = 6$ , and only top  $N = 6$  uni-modal layer representations are used.

## 2.4 Utilization of Uni-Modal Experts

Different layers of uni-modal encoders encoding different levels of semantic information are well demonstrated in vision (Dosovitskiy et al., 2020; Raghu et al., 2021; Naseer et al., 2021) and language (Peters et al., 2018b; Liu et al., 2019a; Jawahar et al., 2019). According to Dosovitskiy et al. (2020) and Raghu et al. (2021), lower layers of ViT tend to attend both locally and globally, while higher layers primarily focus on global information. Similarly, Jawahar et al. (2019) found that the intermediate layers of BERT (Devlin et al., 2019) encode a hierarchy of linguistic information, with surface features at the bottom, syntactic features in the middle, and semantic features at the top.

In the field of VL, some works have explored the usage of pre-trained multi-layer uni-modal representations (Dou et al., 2022; Xu et al., 2022). They simply feed the weighted sum of uni-modal layer representations into the first cross-modal layer, or layer-by-layer exploit multiple top uni-modal layer representations in each cross-modal layer. In this work, we take each layer of the pre-trained uni-modal encoder as a uni-modal **expert**, and the output representation of each layer as the **insight** of the uni-modal expert into the current input.

## 3 Manager Design

Figure 2 depicts the overall framework of ManagerTower. It introduces managers in each cross-modal

layer to adaptively aggregate the insights of pre-trained uni-modal experts at different levels. In the subsequent subsections, we will elaborate on the detailed design schema for the three types of managers, and conclude with the cross-modal encoder with our well-designed managers.<sup>1</sup>

### 3.1 Static Aggregation of Experts

The effectiveness of layer fusion in learning comprehensive representations has been well demonstrated in machine translation (Wang et al., 2018, 2019; Wei et al., 2020). Motivated by this, we decide to apply this technique in the context of VL. As a preliminary approach, we choose to utilize the linear combination of layers method (Wang et al., 2019), which is a simple yet effective way to aggregate the representations of previous layers through the use of learned weights in each encoder layer.

A natural idea is to adapt it to aggregate uni-modal and cross-modal output representations of all previous layers. We name it the Static Aggregation of Experts (SAE) manager. The calculation of the  $\ell^{\text{th}}$  visual manager is:

$$\mathcal{M}_{\ell}^V(\mathbf{V}_7, \dots, \mathbf{V}_{12}, \mathbf{C}_1^V, \dots, \mathbf{C}_{\ell-1}^V) = \sum_{i=1}^{\ell-1} \mathbf{W}_{i+6}^{V,\ell} \odot \text{LN}(\mathbf{C}_i^V) + \sum_{i=1}^6 \mathbf{W}_i^{V,\ell} \odot \text{LN}(\mathbf{V}_{i+6}), \quad (4)$$

where  $\mathcal{M}_{\ell}^V$  denotes the manager for the visual part of the  $\ell^{\text{th}}$  cross-modal layer,  $\mathbf{W}^{V,\ell} \in \mathbb{R}^{(6+\ell-1) \times D}$  is a learnable parameter matrix,  $\odot$  denotes the element-wise product operation and  $\text{LN}(\cdot)$  denotes Layer Normalization (Ba et al., 2016). The

<sup>1</sup>More details on pre-training objectives and downstream fine-tuning are described in Appendix A.

softmax with a learnable temperature is used to normalize  $\mathbf{W}^{V,\ell}$ . We then omit the superscript  $V,\ell$  of  $\mathbf{W}$  for brevity. The learned aggregation weight  $\mathbf{W}$  is initialized with  $\frac{1}{6+\ell-1}$  on average in order to assign equal weights to the output representation of all previous layers.

However, directly applying SAE to VL models is non-trivial, since it does not bring a desired performance improvement compared to BridgeTower but led to a significant performance decrease. We posit that this decrease may be due to the average initialization of  $\mathbf{W}$  not being suitable for cross-modal and pre-trained uni-modal output representations as they have different scales. To investigate this hypothesis, we propose dividing the parameter matrix  $\mathbf{W}$  into uni-modal and cross-modal parts and initializing them with  $\frac{1}{6}$  and  $\frac{1}{\ell-1}$ , respectively,<sup>2</sup> and also learn the softmax temperature separately. The experimental result yield a significant improvement compared to the direct application of SAE, but a limited improvement compared to BridgeTower. These observations provide a compelling argument for re-examining how to aggregate multi-layer pre-trained uni-modal representations.

### 3.2 Static Aggregation of Uni-Modal Experts

Since Equation (4) can be divided into uni-modal and cross-modal parts, by computing the cosine similarity of aggregated uni-modal/cross-modal representations between every two consecutive textual/visual managers, we further analyze the insights aggregated by different SAE managers.

As shown in Figure 3, for SAE managers, the uni-modal similarity is always similar to 1, while the cross-modal similarity increases with depth and gets closer to 1. This indicates that, the uni-modal representations aggregated by different SAE managers are almost identical, and the aggregated cross-modal representations get similar with depth.

We hypothesize that, since different SAE managers provide similar aggregated uni-modal representations to each cross-modal layer, output representation of more preceding cross-modal layers may bring redundant information to confuse the managers. This leads to aggregated cross-modal representations converging to indistinguishable vectors as the depth increases.

Hence, we propose focusing on aggregating the insights of pre-trained uni-modal experts and keep-

<sup>2</sup>We also try some different initialization methods: one, progressive, exponential moving average, BridgeTower-like, etc., but the results are similar to or lower than the average.

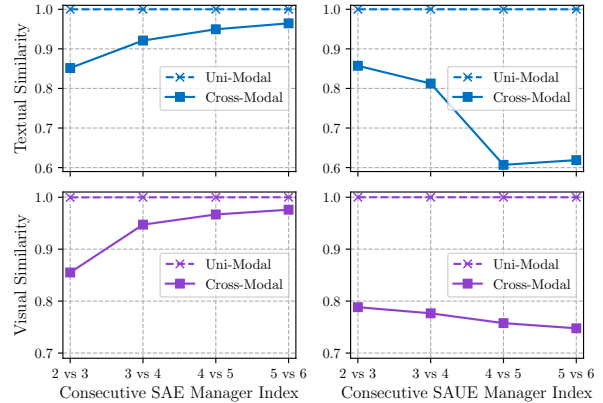


Figure 3: Cosine similarity of aggregated uni-modal/cross-modal representations between every two consecutive textual/visual managers.

ing only the output representation of the previous cross-modal layer. We name it the Static Aggregation of Uni-modal Experts (SAUE) manager. The calculation of the  $\ell^{\text{th}}$  visual manager becomes:

$$\mathcal{M}_{\ell}^V(\mathbf{V}_7, \dots, \mathbf{V}_{12}, \mathbf{C}_{\ell-1}^V) = \mathbf{W}_C \odot \text{LN}(\mathbf{C}_{\ell-1}^V) + \sum_{i=1}^6 \mathbf{W}_i \odot \text{LN}(\mathbf{V}_{i+6}), \quad (5)$$

where  $\mathbf{W} \in \mathbb{R}^{6 \times D}$  and  $\mathbf{W}_C \in \mathbb{R}^{1 \times D}$  are learnable parameter matrices and initialized with  $\frac{1}{6}$  and 1 on average, respectively. The softmax with a learnable temperature only normalizes  $\mathbf{W}$ .

The significant improvement compared to BridgeTower empirically support our hypothesis. Moreover, in Figure 3, the cross-modal similarity of SAUE decreases with depth, which indicates that comprehensive and distinguishable cross-modal representations are learned as depth increases.

### 3.3 Adaptive Aggregation of Uni-Modal Experts

Although the SAUE manager achieves a significant performance improvement, it still has two limitations: (i)  $\mathbf{W}$ , the learned aggregation weight of uni-modal expert insights, is almost identical between managers in different cross-modal layers, as shown in Figure 3 & 7, which is inconsistent with the intuition that the need for uni-modal semantic knowledge varies among cross-modal layers; (ii) in the inference phase, managers in different cross-modal layers use the same aggregation weight of uni-modal expert insights for all tokens in different samples, which does not match the intuition that the need for uni-modal semantic knowledge varies among tokens and samples.



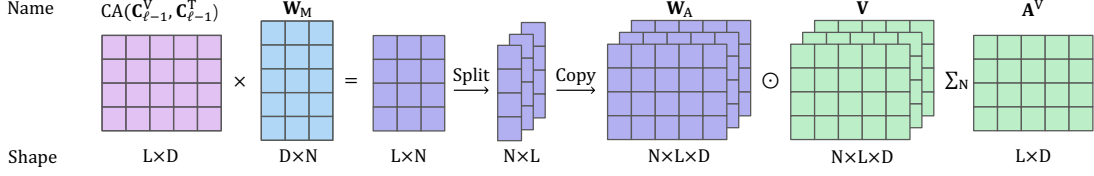


Figure 4: An illustration of the calculation of aggregated uni-modal representations  $\mathbf{A}^V \in \mathbb{R}^{L \times D}$  in the visual AAUE manager. CA denotes the cross-attention mechanism.  $N=6$ . We omit LN and softmax for brevity.

To address the above limitations, we propose the Adaptive Aggregation of Uni-Modal Experts (AAUE) manager. During training and inference phases, AAUE managers can adaptively exploit different levels of uni-modal semantic knowledge from pre-trained uni-modal experts, for different tokens in different samples. Take the visual AAUE manager for example, the calculation of the  $\ell^{\text{th}}$  visual manager becomes:

$$\mathcal{M}_\ell^V(\mathbf{V}_7, \dots, \mathbf{V}_{12}, \mathbf{C}_{\ell-1}^V) = \mathbf{W}_C \odot \text{LN}(\mathbf{C}_{\ell-1}^V) + \sum_{i=1}^6 \mathbf{W}_{A,i} \odot \text{LN}(\mathbf{V}_{i+6}), \quad (6)$$

$$\mathbf{W}_A = \text{softmax}(\text{LN}(\mathbf{C}_{\ell-1}^V) \times \mathbf{W}_M + \epsilon), \quad (7)$$

where  $\mathbf{W}_M \in \mathbb{R}^{D \times 6}$  is a linear projection layer. The generated aggregation weights  $\mathbf{W}_A \in \mathbb{R}^{6 \times L \times D}$  can adaptively aggregate uni-modal representations of each token from different levels of pre-trained uni-modal experts. The softmax has a learnable temperature and  $\epsilon \sim \mathcal{N}(0, \frac{1}{6^2})$  is a Gaussian noise for exploration of aggregation (Xue et al., 2022).

Furthermore, to better help managers to exploit uni-modal semantic knowledge for the current cross-modal layer, we propose replacing the visual query  $\mathbf{C}_{\ell-1}^V$  in Equation (7) with the cross-modal fused query  $\text{CA}(\mathbf{C}_{\ell-1}^V, \mathbf{C}_{\ell-1}^T)$  to further improve performance, where CA is a cross-attention mechanism. We visualize  $\mathbf{W}_A$  in Section 4.4.

### 3.4 Cross-Modal Encoder with Managers

Since the 1<sup>st</sup> cross-modal layer lacks the output representations of the previous cross-modal layer as the query, we introduce the SAUE managers in the 1<sup>st</sup> cross-modal layer and the AAUE managers in the subsequent cross-modal layers. Hence, Equation (1) & (2) of the 1<sup>st</sup> cross-modal layer with SAUE managers becomes:

$$\tilde{\mathbf{C}}_1^V = \mathcal{M}_1^V(\mathbf{V}_7, \dots, \mathbf{V}_{12}), \quad (8)$$

$$\tilde{\mathbf{C}}_1^T = \mathcal{M}_1^T(\mathbf{T}_7, \dots, \mathbf{T}_{12}). \quad (9)$$

For the 2<sup>nd</sup> and subsequent cross-modal layers with AAUE managers:

$$\tilde{\mathbf{C}}_\ell^V = \mathcal{M}_\ell^V(\mathbf{V}_7, \dots, \mathbf{V}_{12}, \mathbf{C}_{\ell-1}^V, \mathbf{C}_{\ell-1}^T), \quad (10)$$

$$\tilde{\mathbf{C}}_\ell^T = \mathcal{M}_\ell^T(\mathbf{T}_7, \dots, \mathbf{T}_{12}, \mathbf{C}_{\ell-1}^T, \mathbf{C}_{\ell-1}^V), \quad (11)$$

where we omit the modality type and layer index embeddings added to uni-modal layer representations  $\mathbf{V}, \mathbf{T}$  in the above equations for simplicity.

Figure 4 shows adaptive aggregation of the insights of pre-trained visual experts in AAUE managers, which is the uni-modal (right) part of Equation (6). As for SAUE managers, they directly broadcast the learned weights  $\mathbf{W} \in \mathbb{R}^{6 \times D}$  to  $\mathbf{W}_A$  and then aggregate the insights.

## 4 Experiments

### 4.1 Implementation Details

ManagerTower consists of a pre-trained textual encoder, RoBERTa<sub>BASE</sub> with 124M parameters, a pre-trained visual encoder, CLIP-ViT B-224/16 with 86M parameters, and a randomly-initialized 6-layer cross-modal encoder with managers which has 113M+12M parameters. The detailed setting of the cross-modal encoder is the same as BridgeTower. The maximum length of the text sequence is set to 50, and the image patch size is  $16 \times 16$ . We use an image resolution of  $384 \times 384$  for Flickr30K and  $576 \times 576$  for VQAv2 for a fair comparison with BridgeTower. AdamW (Loshchilov and Hutter, 2019) optimizer with a base learning rate of  $2e^{-5}$  and warmup ratio of 0.1 is used.

### 4.2 Investigation and Analysis

In this section, we investigate various designs of managers and evaluate the performance by directly fine-tuning on VQAv2 and Flickr30K without VLP. Experimental settings are the same as BridgeTower for a fair comparison. Note that uni-modal encoders are initialized with their pre-trained weights.

#### 4.2.1 Type of Manager

We first investigate the performance of different types of managers and different queries. Take the

Type	Visual Query	Weight	Test-Dev	$R_{\text{MEAN}}$
BT	-	$N \times 1$	75.91	93.33
SAE	-	$N \times 1$	76.19	93.57
		$N \times D$	76.18	93.73
SAUE	-	$N \times 1$	76.38	93.75
		$N \times D$	76.55	93.82
AAUE	$C_{\ell-1}^V$	$N \times L$	76.52	93.84
	$C_{\ell-1}^V, C_{\ell-1}^T$	$N \times L$	<b>76.65</b>	<b>93.97</b>
Concat-Attention	$V, C_{\ell-1}^V$	$N \times L \times D$	76.38	93.78
	$V, C_{\ell-1}^V, C_{\ell-1}^T$	$N \times L \times D$	76.43	93.83
Cross-Attention	$C_{\ell-1}^V$	$N \times L$	76.41	92.15
	$C_{\ell-1}^V, C_{\ell-1}^T$	$N \times L$	76.45	92.61

Table 1: Performance of different types of managers and different queries on VQAv2 and Flickr30K.  $R_{\text{MEAN}}$  indicates the mean recall metrics for image-text retrieval.

visual manager for example, based on the top  $N=6$  visual layer representations  $V \in \mathbb{R}^{N \times L \times D}$  from CLIP-ViT, different managers provide the aggregation weights that can be broadcast to  $W_A$  for aggregating the insights of pre-trained visual experts. From the perspective of aggregation weights  $W_A$ , the SAE and SAUE managers are **static** sentence-level managers that share the same aggregation weights for all tokens in different samples. Correspondingly, the AAUE manager is an **adaptive** token-level manager that adaptively **generates** different aggregation weights for different tokens in different samples. Besides, we also implement Equation (7) with commonly used cross-attention and concat-attention mechanisms for comparison.

Results are shown in Table 1. By focusing on aggregating the insights of pre-trained uni-modal experts, the SAUE manager outperforms the SAE manager on both datasets. Furthermore, with the help of the cross-modal fused query, the AAUE manager achieves substantially better performance than other managers. This demonstrates the effectiveness of adaptive token-level aggregation with the cross-modal fused query compared to static sentence-level aggregation. Notably, the cross-modal fused query incorporates output representations of both visual and textual parts of the previous cross-modal layer, which can better help managers to correctly aggregate uni-modal semantic knowledge required by the current cross-modal layer.

#### 4.2.2 Number of Cross-Modal Layers

We compare ManagerTower to BridgeTower with different numbers of cross-modal layers in Table 2 to further evaluate the effectiveness of ManagerTower. Regardless of the number of cross-modal layers, ManagerTower consistently and signifi-

$L_C$	VQAv2 Test-Dev		Flickr30K $R_{\text{MEAN}}$	
	BT	Ours	BT	Ours
2	74.86	75.47 ( $\uparrow 0.61$ )	92.45	93.31 ( $\uparrow 0.86$ )
3	75.33	76.04 ( $\uparrow 0.71$ )	92.50	93.41 ( $\uparrow 0.91$ )
4	75.74	76.26 ( $\uparrow 0.52$ )	92.76	93.59 ( $\uparrow 0.83$ )
6	75.91	<b>76.65</b> ( $\uparrow 0.74$ )	93.33	<b>93.97</b> ( $\uparrow 0.64$ )
8	75.89	76.47 ( $\uparrow 0.58$ )	93.03	93.65 ( $\uparrow 0.62$ )

Table 2: Performance of BridgeTower (BT) and ManagerTower with different number of cross-modal layers.

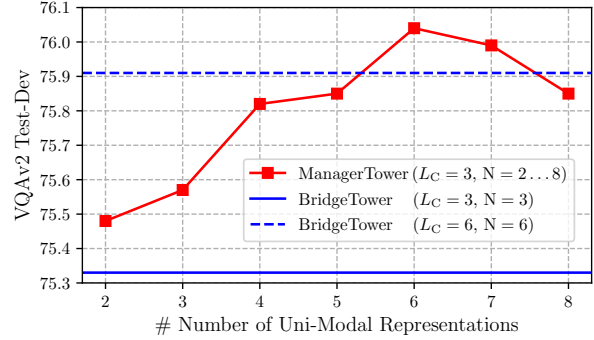


Figure 5: Effect of using different numbers of uni-modal representations in ManagerTower( $L_C=3, N=2 \dots 8$ ).

cantly outperforms BridgeTower on both datasets.

More interestingly, the performance of ManagerTower with  $L_C=3$  (76.04) is even better than that of BridgeTower with  $L_C=6$  (75.91). Unlike BridgeTower, the number of uni-modal layer representations used  $N$  in ManagerTower is not tied to the number of cross-modal layers  $L_C$  and can be flexibly adjusted. We fix  $N=6$  as the default setting. Therefore, ManagerTower actually uses the same number of uni-modal layer representations as BridgeTower, but achieves even better performance using half the number of cross-modal layers. This further demonstrates the flexibility and effectiveness of ManagerTower to adaptively aggregate uni-modal semantic knowledge, compared to layer-by-layer exploitation in BridgeTower.

#### 4.2.3 Number of Uni-Modal Experts.

We further investigate the effect of varying  $N$  in ManagerTower with  $L_C=3$ . As shown in Figure 5, there exist two interesting observations: (i) ManagerTower ( $L_C=3, N=3$ ) is still better than BridgeTower ( $L_C=3, N=3$ ). This indicates that when the same number of uni-modal layer representations are introduced, ManagerTower allows more effective aggregation of uni-modal semantic knowledge, thus facilitating cross-modal alignment and fusion in each cross-modal layer. (ii) the performance of ManagerTower first increases

Model	# Pre-train Images	VQAv2		SNLI-VE		NLVR <sup>2</sup>		Flickr30K	
		Test-Dev	Test-Std	Dev	Test	Dev	Test-P	IR@1	TR@1
<i>Base-size models pre-trained on 4M public data</i>									
ViLT <sub>BASE</sub> (Kim et al., 2021)	4M	71.26	-	-	-	75.70	76.13	64.4	83.5
UNITER <sub>BASE</sub> (Chen et al., 2020) *	4M	72.70	72.91	78.59	78.28	77.18	77.85	72.52	85.90
UNIMO <sub>BASE</sub> (Li et al., 2021b)	4M	73.79	74.02	80.00	79.10	-	-	74.66	89.70
ALBEF <sub>BASE</sub> (Li et al., 2021a) *	4M	74.54	74.70	80.14	80.30	80.24	80.50	82.8	94.3
METER-Swin <sub>BASE</sub> (Dou et al., 2022)	4M	76.43	76.42	80.61	80.45	82.23	82.47	79.02	92.40
VLMO <sub>BASE</sub> (Wang et al., 2021a)	4M	76.64	76.89	-	-	82.77	83.34	79.3	92.3
METER-CLIP <sub>BASE</sub> (Dou et al., 2022)	4M	77.68	77.64	80.86	81.19	82.33	83.05	82.22	94.30
BridgeTower <sub>BASE</sub> (Xu et al., 2022)	4M	78.66	78.73	81.11	81.19	81.85	83.09	85.83	94.73
ManagerTower <sub>BASE</sub> (Ours)	4M	<b>79.39</b>	<b>79.15</b>	<b>81.26</b>	<b>81.44</b>	<b>82.81</b>	<b>83.34</b>	<b>86.56</b>	<b>95.64</b>
<i>Models pre-trained on more data and/or with larger size</i>									
UNITER <sub>LARGE</sub> (Chen et al., 2020) *	4M	73.82	74.02	79.39	79.38	79.12	79.98	75.56	87.30
UNIMO <sub>LARGE</sub> (Li et al., 2021b)	4M	75.06	75.27	81.11	80.63	-	-	78.04	89.40
ALBEF <sub>BASE</sub> (Li et al., 2021a) *	14M	75.84	76.04	80.80	80.91	82.55	83.14	85.6	95.9
SimVLM <sub>BASE</sub> (Wang et al., 2021b)	1.8B	77.87	78.14	84.20	84.15	81.72	81.77	-	-
BLIP <sub>BASE</sub> (Li et al., 2022a) *	129M	78.24	78.17	-	-	82.48	83.08	87.3	97.3
SimVLM <sub>LARGE</sub> (Wang et al., 2021b)	1.8B	79.32	79.56	85.68	85.62	84.13	84.84	-	-

Table 3: Comparisons with previous models on downstream VL tasks. The best score is bolded. \* indicates that the model also uses VG-QA data to fine-tune on VQAv2.

gradually, but decreases after  $N > 6$ . We assume that lower-layer uni-modal representations may not help ManagerTower learn cross-modal fusion and also increases the computational cost, which is also consistent with the observation in Xu et al. (2022).

### 4.3 Comparison with Previous Arts

**Pre-train Settings.** We pre-train ManagerTower with two standard VLP objectives, masked language modeling (MLM) and image-text matching (ITM), on the commonly used 4M public data: Conceptual Captions (CC) (Sharma et al., 2018), SBU Captions (Ordonez et al., 2011), MSCOCO Captions (Chen et al., 2015), and Visual Genome (VG) (Krishna et al., 2017). The pre-train settings are the same as BridgeTower and METER for a fair comparison. ManagerTower is pre-trained for 100k steps with a batch size of 4096 and a learning rate of  $1e^{-5}$ . The image resolution for VLP is  $288 \times 288$  and only center-crop (Radford et al., 2021) is used without any data augmentation.

**Main Results.** Table 3 shows the performance of ManagerTower compared with other previous works on various downstream VL tasks. ManagerTower achieves superior performances on these datasets with only 4M VLP data. With the same pre-training and fine-tuning settings and uni-modal backbones as previous strong baselines METER and BridgeTower, ManagerTower significantly improves performances on various downstream VL tasks, especially 79.15% accuracy on VQAv2

Test-Std, 86.56% IR@1 and 95.64% TR@1 on Flickr30K. This further demonstrates that with all other factors fixed, compared to BridgeTower that introduces bridges to METER, ManagerTower allows more effective aggregation of multi-layer uni-modal representations via well-designed managers. Managers can adaptively aggregate more accurate uni-modal semantic knowledge to facilitate comprehensive cross-modal alignment and fusion in each cross-modal layer. Notably, ManagerTower not only outperforms many base-size models pre-trained on 4M data, but also surpasses some models pre-trained on more data and/or with larger size.

### 4.4 Visualization of Aggregation Weights

We delve into managers by visualizing the average aggregation weights they generate for each cross-modal layer over all samples in VQAv2 Valid in Figure 6. For each row, the first column shows the learned aggregation weights of SAUE managers. The other five columns show the aggregation weights generated by AAUE managers and share the Y-axis to provide easy horizontal comparison.

Interestingly, the aggregation weight distributions provided by managers are completely different from the one-hot distributions specified in BridgeTower, and there are two distinct trends: (i) For SAUE managers in the 1<sup>st</sup> cross-modal layer, vertically: textual manager exhibits increasing and then decreasing weights, most favoring  $T_{10}$ , unlike  $T_{12}$  and  $T_7$  used in METER and BridgeTower,

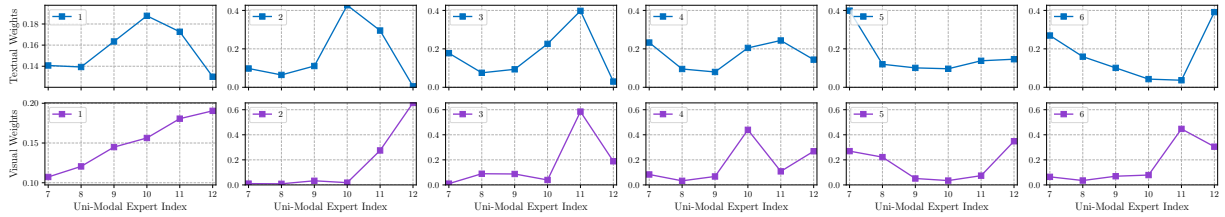


Figure 6: A visualization of aggregation weights of textual and visual AAUE managers in each cross-modal layer after VLP. The X-axis is the index of the uni-modal expert, and the legend shows the index of the cross-modal layer.

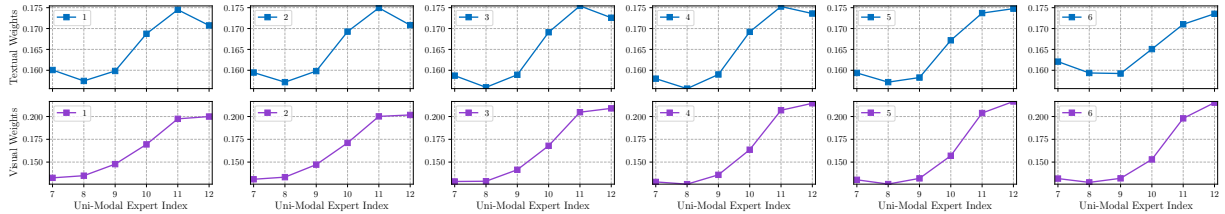


Figure 7: A visualization of aggregation weights of textual and visual SAUE managers in each cross-modal layer. The X-axis is the index of the uni-modal expert, and the legend shows the index of the cross-modal layer.

respectively; visual manager exhibits increasing weights, most favoring  $V_{12}$ , the same as METER and BridgeTower. (ii) For AAUE managers in the 2<sup>nd</sup> to 6<sup>th</sup> cross-modal layers, horizontally: whether textual or visual managers, they exhibit diverse aggregation weight distributions in different layers.

Overall, comparing the aggregation weight distributions horizontally and vertically, ManagerTower learns diverse distributions in different cross-modal layers. This provides strong evidence that the introduced managers can adaptively aggregate uni-modal semantic knowledge for comprehensively cross-modal representation learning.

#### 4.5 Intuitive Comparison Between BT&MT

We provide brief illustrations in Figure 8 to intuitively compare BridgeTower (BT) and ManagerTower (MT) with different type of managers.

**BT vs. MT with SAUE Managers.** In Table 2 & 5, we provide the performance comparison between BridgeTower and ManagerTower.<sup>3</sup> In fact, BridgeTower can be seen as an approximate special case of ManagerTower with SAUE managers if we replace the learned weights  $\mathbf{W}$  in each manager with layer-by-layer one-hot distributions<sup>4</sup> used in BridgeTower. However, as shown in Figure 7, the aggregation weight of textual and visual SAUE managers share a similar progressive trend

across cross-modal layers, which is completely different from the distributions in BridgeTower. This allows ManagerTower with SAUE managers to achieve significant performance gains (from 75.91 to 76.55) compared to BridgeTower. Besides, the similar trend of aggregation weights is consistent with the observations in Figure 3, that is, the cosine similarity of aggregated uni-modal representations between managers is always similar to 1.

**SAUE Manager vs. AAUE Manager.** When we compare Figure 6 & 7, their respective aggregation weight distributions are completely different. This further demonstrates that compared with SAUE managers, AAUE managers can adaptively **generates** different aggregation weights for different tokens in different samples. Interestingly, the first column of two figures both comes from the SAUE managers, but the distributions are still clearly different. We presume that high-layer AAUE managers may help low-layer SAUE managers **rectify** their management of experts.

We also provide the visualizations of aggregation weights of SAE and AAUE managers without VLP in Figure 9 & 10. Comparing the visualization of three types of managers without VLP, we can find that (i) the learned aggregation weights of SAE and SAUE managers are still a little close to the average initialization we used and they all share a similar progressive trend across cross-modal layers; (ii) for each AAUE manager, its generated aggregation weights vary significantly across 6 uni-modal experts; comparing different cross-modal layers,

<sup>3</sup>The re-implemented BridgeTower obtained higher experimental results than the original paper due to the better fine-tuning settings we used for all experiments in Section 4.2.

<sup>4</sup>It means that, for each cross-modal layer, only one uni-modal expert is activated at a time in the bottom-up direction.



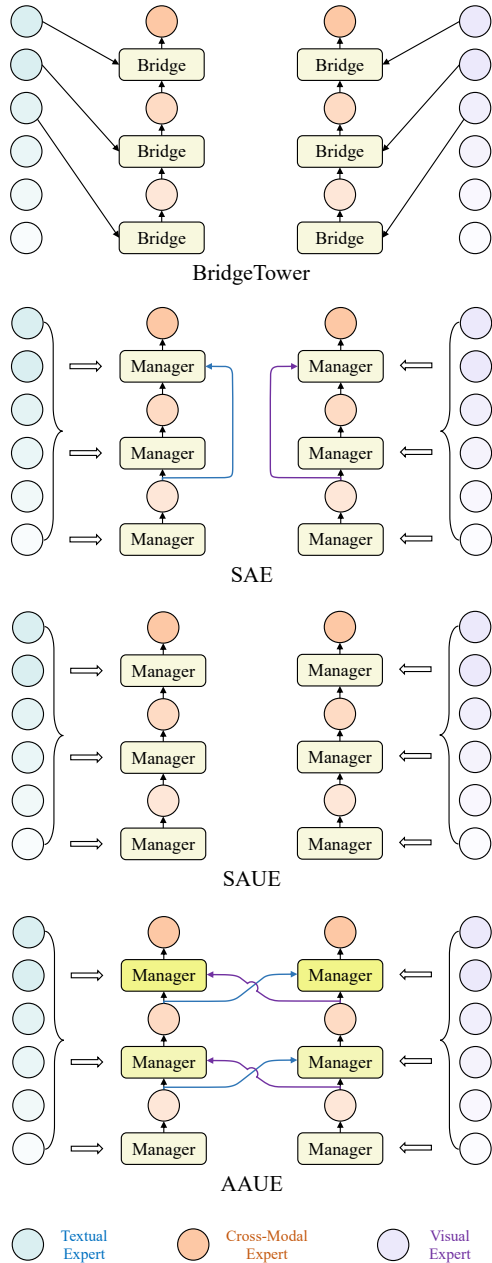


Figure 8: Brief illustrations of BridgeTower and our ManagerTower with SAE, SAUE and AAUE managers. Hollow arrows indicate the transmission of multi-layer uni-modal representations in ManagerTower instead of layer-by-layer transmission in BridgeTower. Each uni-modal or cross-modal layer is seen as a uni-modal or cross-modal expert. The arrow between the cross-modal expert of the previous layer and the manager of the current layer is to get the cross-modal fused query.

the distribution of aggregation weights generated by the AAUE manager is also very different.

## 5 Related Work

**Vision-Language Models.** Although VL models differ in model architecture, most of them use uni-

modal encoders to extract visual and textual representations, and then fuse them in a cross-modal encoder, which can be unified into the Two-Tower architecture (Lu et al., 2019; Su et al., 2020; Chen et al., 2020; Li et al., 2020a,b; Zhou et al., 2020; Kim et al., 2021; Radford et al., 2021; Jia et al., 2021; Li et al., 2021a,b, 2022a; Dou et al., 2022; Wang et al., 2021a,b, 2022a,b; Yu et al., 2022). As a representative model, METER (Dou et al., 2022) adopts pre-trained uni-modal encoders and feeds their last-layer representations into the cross-modal encoder. BridgeTower (Xu et al., 2022) proposes building layer-by-layer connections between the top uni-modal layers and each cross-modal layer to utilize different uni-modal layer representations. However, they still cannot provide adaptive and effective aggregation of multi-layer pre-trained uni-modal representations in each cross-modal layer.

**Multi-Layer Representation Aggregation.** The effectiveness of layer representation aggregation in learning comprehensive representations has been well demonstrated in vision (Lin et al., 2017; Huang et al., 2017; Yu et al., 2018; Xie et al., 2021) and language (Peters et al., 2018a; Wang et al., 2018, 2019; Wei et al., 2020). Recent VL models also explore utilization of multi-layer uni-modal representations for better cross-modal representation learning. METER feeds the weighted sum of uni-modal representations into the first cross-modal layer. BridgeTower introduces bridges into METER so that different uni-modal layer representations are fed layer by layer into each cross-modal layer. In this work, ManagerTower explores adaptive and effective aggregation of multi-layer uni-modal representations via well-designed managers.

## 6 Conclusion

We propose ManagerTower, a novel VL model architecture that gathers and combines the insights of pre-trained uni-modal experts at different levels via the introduced managers in each cross-modal layer. The feasibility of various designs of managers is well explored, and the effectiveness of ManagerTower on various downstream VL tasks is well demonstrated. More comprehensive cross-modal alignment and fusion in each cross-modal layer is achieved by adaptive aggregation of different levels of uni-modal semantic knowledge. We hope that our work can inspire more research on how to better exploit multi-layer pre-trained uni-modal representations for cross-modal representation learning.

## Limitations

In this work, we propose managers that allow adaptive aggregation of uni-modal layer representations in each cross-modal layer. Inevitably, AAUE managers significantly improve performance which slightly increasing the computational budget, as we detailed discussed in Appendix C. This needs to be further optimized in the future. Analysis and optimization are also needed for the other types of managers as shown in Appendix D. Moreover, as shown in Figure 5, the performance of ManagerTower first increases gradually with the number of uni-modal representations, but then stops increasing and even decreases when the number of uni-modal representations exceeds 6. How to obtain better ManagerTower performance using a lower computational budget while utilizing more insights of uni-modal experts, especially when scaling the model, *e.g.*, 24-layer CLIP-ViT L-224/16 and 24-layer RoBERTa<sub>LARGE</sub>, is a question worth further exploration. For example, designing reasonable sparse activation functions for managers in ManagerTower, instead of simple top-N or top-p sampling (which did not work well in our preliminary experiments).

## Acknowledgements

This work was supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 62236004 and 61976072.

## References

- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. [Layer normalization](#). *ArXiv preprint*, abs/1607.06450.
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. [Microsoft coco captions: Data collection and evaluation server](#). *ArXiv preprint*, abs/1504.00325.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. [Uniter: Universal image-text representation learning](#). In *Proc. of ECCV*.
- Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. 2020. [Randaugment: Practical automated data augmentation with a reduced search space](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *Proc. of ICLR*.
- Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, Zicheng Liu, and Michael Zeng. 2022. [An empirical study of training end-to-end vision-and-language transformers](#). *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. [Large-scale adversarial training for vision-and-language representation learning](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. [Making the V in VQA matter: Elevating the role of image understanding in visual question answering](#). In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6325–6334. IEEE Computer Society.
- Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. [Densely connected convolutional networks](#). In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society.
- Ganesh Jawahar, Benoît Sagot, and Djamel Seddah. 2019. [What does BERT learn about the structure of language?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. [Scaling up visual and vision-language representation learning with noisy text supervision](#). In *Proc. of ICML*.
- Andrej Karpathy and Fei-Fei Li. 2015. [Deep visual-semantic alignments for generating image descriptions](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA*,

- June 7-12, 2015, pages 3128–3137. IEEE Computer Society.
- Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proc. of ICML*.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*.
- Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020a. [Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 11336–11344. AAAI Press.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022a. [Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation](#). *ArXiv preprint*, abs/2201.12086.
- Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021a. Align before fuse: Vision and language representation learning with momentum distillation. *Proc. of NeurIPS*.
- Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2021b. [UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2592–2607, Online. Association for Computational Linguistics.
- Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2022b. [Unimo-2: End-to-end unified vision-language grounded learning](#). *ArXiv preprint*, abs/2203.09067.
- Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020b. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proc. of ECCV*.
- Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. 2017. [Feature pyramid networks for object detection](#). In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944. IEEE Computer Society.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. [Linguistic knowledge and transferability of contextual representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. [Roberta: A robustly optimized bert pretraining approach](#). *ArXiv preprint*, abs/1907.11692.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. [Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23.
- Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2021. Intriguing properties of vision transformers. *Proc. of NeurIPS*.
- Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. [Im2text: Describing images using 1 million captioned photographs](#). In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1143–1151.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. [Dissecting contextual word embeddings: Architecture and representation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proc. of ICML*.



- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.
- Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. 2021. Do vision transformers see like convolutional neural networks? *Proc. of NeurIPS*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. **Neural machine translation of rare words with subword units**. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. **Conceptual captions: A cleaned, hypervised, image alt-text dataset for automatic image captioning**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.
- Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. **How much can clip benefit vision-and-language tasks?** *ArXiv preprint*, abs/2107.06383.
- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. **VL-BERT: pre-training of generic visual-linguistic representations**. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. **A corpus for reasoning about natural language grounded in photographs**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.
- Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. 2018. **Tips and tricks for visual question answering: Learnings from the 2017 challenge**. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4223–4232. IEEE Computer Society.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. **Attention is all you need**. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022a. **Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework**. *ArXiv preprint*, abs/2202.03052.
- Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. **Learning deep transformer models for machine translation**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.
- Qiang Wang, Fuxue Li, Tong Xiao, Yanyang Li, Yinqiao Li, and Jingbo Zhu. 2018. **Multi-layer representation fusion for neural machine translation**. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3015–3026, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. 2022b. **Image as a foreign language: Beit pretraining for all vision and vision-language tasks**. *ArXiv preprint*, abs/2208.10442.
- Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. 2021a. **Vlmo: Unified vision-language pre-training with mixture-of-modality-experts**. *ArXiv preprint*, abs/2111.02358.
- Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021b. **Simvlm: Simple visual language model pretraining with weak supervision**. *ArXiv preprint*, abs/2108.10904.
- Xiangpeng Wei, Heng Yu, Yue Hu, Yue Zhang, Rongxiang Weng, and Weihua Luo. 2020. **Multiscale collaborative deep models for neural machine translation**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 414–426, Online. Association for Computational Linguistics.
- Enze Xie, Wenhui Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. 2021. **Segmenter: Simple and efficient design for semantic segmentation with transformers**. *Proc. of NeurIPS*.
- Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. **Visual entailment: A novel task for fine-grained image understanding**. *ArXiv preprint*, abs/1901.06706.
- Xiao Xu, Chenfei Wu, Shachar Rosenman, Vasudev Lal, and Nan Duan. 2022. **Bridge-tower: Building bridges between encoders in vision-language representation learning**. *ArXiv preprint*, abs/2206.08657.
- Fuzhao Xue, Ziji Shi, Futao Wei, Yuxuan Lou, Yong Liu, and Yang You. 2022. **Go wider instead of deeper**. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8779–8787.



Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. [From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions](#). *Transactions of the Association for Computational Linguistics*, 2:67–78.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. 2018. [Deep layer aggregation](#). In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2403–2412. IEEE Computer Society.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. [Coca: Contrastive captioners are image-text foundation models](#). *ArXiv preprint*, abs/2205.01917.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. [Vinvl: Revisiting visual representations in vision-language models](#). In *Proc. of CVPR*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. [Unified vision-language pre-training for image captioning and vqa](#). In *Proc. of AAAI*.

## A Implementation Details

### A.1 Vision-Language Pre-training

We use two commonly used VLP objectives.

**Masked Language Modeling.** For MLM, we follow the conditional masking approach used in UNITER (Chen et al., 2020) that randomly masks 15% of the tokens in the text token sequence while keeping the image patch sequence unchanged. The model is then trained to predict the original masked tokens given the incomplete text sequence and the complete image patch sequence. The masking strategy and MLM task head we use are the same as RoBERTa. The output top-layer representation of the textual part of the cross-modal encoder is used as input for the MLM task head.

**Image-Text Matching.** For ITM, both matched and mismatched image-text pairs are fed into the model with equal probability. The model is trained to predict whether a given image-text pair is a matched (positive) or a mismatched (negative) pair. The output top-layer representations of `[class]` and `[<s>]` tokens are activated by the non-linear function `Tanh`. Then the concatenation of the above output representations is fed into a linear classifier with cross-entropy loss for binary classification.

	COCO	VG	CC	SBU
# Images	113K	108K	2.9M	860K
# Captions	567K	4.8M	2.9M	860K

Table 4: Statistics of the pre-train datasets. We remove duplicate image-caption pairs in VG (Kim et al., 2021; Dou et al., 2022) and only 2.9M image-caption pairs can be downloaded in CC.

**Pre-training Settings.** Table 4 shows the statistics of the pre-train datasets. Following previous work (Kim et al., 2021; Chen et al., 2020; Li et al., 2021a; Dou et al., 2022), we adopt four public image-caption datasets for pre-training, including Conceptual Captions (CC) (Sharma et al., 2018), SBU Captions (SBU) (Ordonez et al., 2011), MSCOCO Captions (COCO) (Chen et al., 2015), and Visual Genome (VG) (Krishna et al., 2017). The total numbers of the unique images and image-caption pairs in the combined training data are 4M and 9M. Table 8 describes the hyperparameters for pre-training the ManagerTower. The learning rate of the cross-modal encoder is five times higher than that of uni-modal encoders (Dou et al., 2022).

### A.2 Fine-Tuning on Downstream Tasks

**Dataset Setting.** Standard settings and splits are used for all datasets. For Flickr30K dataset (Young et al., 2014), we follow the standard Karpathy Split (Karpathy and Li, 2015). For VQAv2 (Goyal et al., 2017) dataset, we follow the common practice (Goyal et al., 2017; Teney et al., 2018): convert VQAv2 to a classification task with 3, 129 answer classes; train the model with training data and validation data, and evaluate the model on the Test-Dev and Test-Std data.

**Image Augmentation.** We follow previous works (Li et al., 2021a, 2022a) to use RandomResizedCrop, RandomHorizontalFlip, and RandAugment (Cubuk et al., 2020) to augment the images.

**Fine-Tuning Strategy.** For visual question answering, visual entailment and visual reasoning, the fine-tuning strategy is similar to the strategy we used in ITM. For image-text retrieval, we follow the approach used in ALBEF (Li et al., 2021a) to optimize our model with both image-text contrastive (ITC) and ITM objectives. In the training phase, we first add two linear projections on top of the uni-modal encoders and calculate the contrastive similarity of uni-modal representations of image-text pairs by dot product to compute the

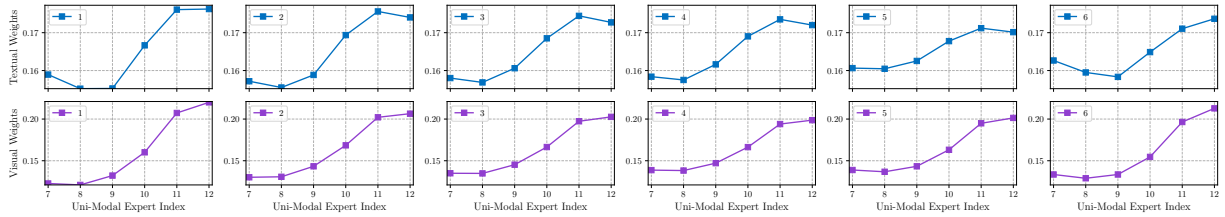


Figure 9: A visualization of aggregation weights of textual and visual SAE managers in each cross-modal layer. The X-axis is the index of the uni-modal expert, and the legend shows the index of the cross-modal layer.

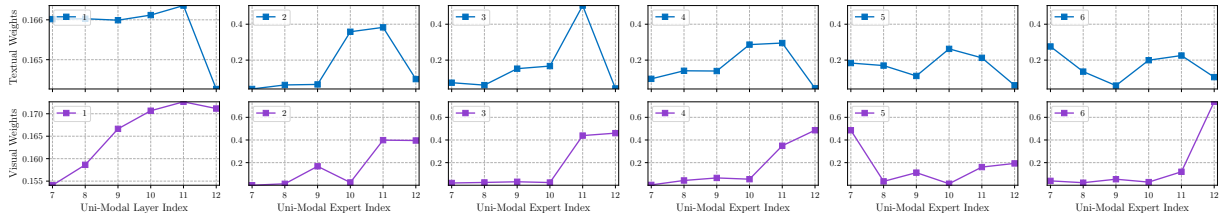


Figure 10: A visualization of aggregation weights of textual and visual AAUE managers in each cross-modal layer. The X-axis is the index of the uni-modal expert, and the legend shows the index of the cross-modal layer.

	Visual Backbone	Textual Backbone	VQAv2 Test-Dev		Flickr30K $R_{MEAN}$	
			BridgeTower	ManagerTower	BridgeTower	ManagerTower
	DeiT B-224/16	RoBERTa	71.22	72.20 ( $\uparrow$ 0.98)	87.63	88.72( $\uparrow$ 1.09)
	ViT B-224/16	RoBERTa	72.82	73.67 ( $\uparrow$ 0.85)	90.48	90.92( $\uparrow$ 0.44)
	ViT B-384/16	RoBERTa	72.94	73.80 ( $\uparrow$ 0.86)	90.51	90.96( $\uparrow$ 0.45)
	CLIP-ViT B-224/32	RoBERTa	73.73	74.79 ( $\uparrow$ 1.06)	91.33	91.76( $\uparrow$ 0.43)
	CLIP-ViT B-224/16	BERT	75.74	76.36 ( $\uparrow$ 0.62)	92.84	93.42( $\uparrow$ 0.58)
	CLIP-ViT B-224/16	RoBERTa	75.91	<b>76.65</b> ( $\uparrow$ 0.74)	93.33	<b>93.97</b> ( $\uparrow$ 0.64)

Table 5: Performance of BridgeTower and ManagerTower with different visual and textual backbones. B, N and M in “ViT B-N/M” denote the model size, image resolution and patch size, respectively.

ITC loss. Formerly, negative image-text pairs in ITM loss are sampled randomly. However, after computing the ITC loss, we can use contrastive similarity distribution to sample one hard in-batch negative text (image) for each image (text) in a mini-batch. In the inference phase, we first compute the contrastive similarity for all images and texts, and then select the top-k candidates based on their contrastive similarity. We then calculate their ITM scores for these candidates to determine the final ranking.

**Fine-Tuning Settings.** Similar to the image-text matching (ITM) pre-training objective, we pass the final representation of [class] token and [ <s> ] token to the non-linear layer activated by Tanh, and feed the concatenation of the output into a linear classifier (Flickr30K) or an MLP classifier (VQAv2, SNLI-VE and NLVR<sup>2</sup>). We apply cross-entropy loss for SNLI-VE, NLVR<sup>2</sup> and Flickr30K and binary cross-entropy loss for VQAv2 (Kim et al., 2021; Dou et al., 2022). Fine-

tuning hyperparameters for VQAv2, SNLI-VE, NLVR<sup>2</sup>, and Flickr30K are given in Table 9.

## B Switch Visual and Textual Backbones

We experiment with different pre-trained visual and textual backbones as uni-modal encoders to further investigate the impact on performance of the managers of ManagerTower compared to the bridges of BridgeTower. As shown in Table 5, regardless of the visual and textual backbones we apply, ManagerTower significantly and consistently outperforms BridgeTower on both datasets. This further proves the effectiveness and generalization of our proposed ManagerTower architecture and managers, which can provide adaptive and effective aggregation of multi-layer uni-modal representations for vision-language representation learning.

## C Computational Budget

Table 6 shows the computational budget and downstream task performance without VLP for

Model	Manager Type	Manager Visual Query	# Params (M)	# FLOPs (G)	Inference Time (ms)	VQAv2 Test-Dev	Flickr30K R <sub>MEAN</sub>
BridgeTower <sub>BASE</sub> *	-	-	326.58	101.25	39.43±1.55	75.91	93.33
ManagerTower <sub>BASE</sub>	SAUE	-	326.77	101.34	41.12±1.41	76.55 (↑0.64)	93.73 (↑0.40)
ManagerTower <sub>BASE</sub>	AAUE	$C_{\ell-1}^V$	326.77	101.35	41.80±1.05	76.52 (↑0.61)	93.84 (↑0.51)
ManagerTower <sub>BASE</sub>	AAUE	$C_{\ell-1}^V, C_{\ell-1}^T$	338.64	105.52	43.20±1.37	76.65 (↑0.74)	93.97 (↑0.64)

Table 6: Computational budget and downstream task performance without VLP for BridgeTower and ManagerTower. \* denotes our re-implementation.

BridgeTower and ManagerTower, including the number of parameters, the number of Floating-Point operations (FLOPs)<sup>5</sup>. We measure the average inference time of processing 1 VQA instance over 10K runs on 1 NVIDIA TITAN V GPU. The sequence length is 50, and the image resolution is  $384 \times 384$ . Compared with BridgeTower (1<sup>st</sup> row), ManagerTower (4<sup>th</sup> row) uses an acceptable additional computational budget (3.69% parameters, 4.22% FLOPs, and 3.77ms inference time) and achieves significant performance improvements of 0.74% and 3.1% on VQAv2 and Flickr30K, respectively. We further analyze other well-performed variants of ManagerTower in the 2<sup>nd</sup> and 3<sup>rd</sup> rows. It is worth noting that the two variants share a similar computational budget as BridgeTower, but achieve better performance. This not only demonstrates the efficiency and effectiveness of our ManagerTower architecture, but also reminds us that the cross-modal fused query via the cross-attention mechanism is the main reason for the additional computational budget of ManagerTower (4<sup>th</sup> row), as it is the only difference between the 3<sup>rd</sup> and 4<sup>th</sup> row models. This inspires us to explore a more efficient method to fuse  $C_{\ell-1}^V$  and  $C_{\ell-1}^T$  to get the cross-modal fused query in the future.

## D Details on Cross-Attention and Concat-Attention Managers

**Cross-Attention Managers.** We implement the standard cross-attention mechanism (Vaswani et al., 2017) and reduce the linear projection layer for value to save computational budget.<sup>6</sup> Take the visual manager for example, it takes  $C_{\ell-1}^V \in \mathbb{R}^{L \times D}$  as the query, and the first token of multi-layer unimodal representations, *i.e.*,  $\mathbf{V}[:, 0] \in \mathbb{R}^{N \times D}$ , as the key. Hence, the shape of generated aggregation weights is  $N \times L$ , which can be broadcast to

the aggregation weights  $\mathbf{W}_A \in \mathbb{R}^{N \times L \times D}$ . The following calculation is the same as AAUE managers in Figure 4. The results in Table 1 show a significant decrease compared to other managers on Flickr30K. We leave the detailed analysis of this phenomenon to the future work.

**Concat-Attention Managers.** Take the visual manager as an example, it broadcasts  $C_{\ell-1}^V \in \mathbb{R}^{L \times D}$  to  $\mathbb{R}^{N \times L \times D}$ , and concatenates it with  $\mathbf{V} \in \mathbb{R}^{N \times L \times D}$  along the last dimension as the concatenated query. It then directly projects the query to  $\mathbf{W}_A \in \mathbb{R}^{N \times L \times D}$ . The following calculation is the same as AAUE managers in Figure 4. In fact, this type of manager is different from all other managers from the perspectives of the generated aggregation weights. Although its aggregation weights delve into the feature dimension of  $C_{\ell-1}^V$  and  $\mathbf{V}$ , the substantially increased number of parameters and computational cost do not result in a significant performance gain, making it impractical and inefficient. More efficient variants of this type of manager should be investigated in the future.

## E Detailed Comparison with Previous Arts

Due to the space limitations, we omit some baselines and details in Table 3. Here we provide more details on the comparison with previous arts in Table 7.

<sup>5</sup>We use Facebook Research’s `fvcore` to calculate FLOPs.

<sup>6</sup>The calculation of cross-modal fused query also uses this simplified version of the cross-attention mechanism.

Model	# Pre-train Images	Visual Backbone	VQAv2		SNLI-VE		NLVR <sup>2</sup>		Flickr30K	
			Test-Dev	Test-Std	Dev	Test	Dev	Test-P	IR@1	TR@1
<i>Base-size models pre-trained on 4M public data</i>										
ViLT <sub>BASE</sub> (Kim et al., 2021)	4M	ViT B-384/32	71.26	-	-	-	75.70	76.13	64.4	83.5
UNITER <sub>BASE</sub> (Chen et al., 2020) *	4M	Faster R-CNN	72.70	72.91	78.59	78.28	77.18	77.85	72.52	85.90
VILLA <sub>BASE</sub> (Gan et al., 2020) *	4M	Faster R-CNN	73.59	73.67	79.47	79.03	78.39	79.30	74.74	86.60
UNIMO <sub>BASE</sub> (Li et al., 2021b)	4M	Faster R-CNN	73.79	74.02	80.00	79.10	-	-	74.66	89.70
ALBEF <sub>BASE</sub> (Li et al., 2021a) *	4M	DeiT B-224/16	74.54	74.70	80.14	80.30	80.24	80.50	82.8	94.3
VinVL <sub>BASE</sub> (Zhang et al., 2021)	5.7M	ResNeXt-152	75.95	76.12	-	-	82.05	83.08	-	-
METER-Swin <sub>BASE</sub> (Dou et al., 2022)	4M	Swin B-384/32	76.43	76.42	80.61	80.45	82.23	82.47	79.02	92.40
VLMO <sub>BASE</sub> (Wang et al., 2021a)	4M	BEiT B-224/16	76.64	76.89	-	-	82.77	83.34	79.3	92.3
METER-CLIP <sub>BASE</sub> (Dou et al., 2022)	4M	CLIP-ViT B-224/16	77.68	77.64	80.86	81.19	82.33	83.05	82.22	94.30
BridgeTower <sub>BASE</sub> (Xu et al., 2022)	4M	CLIP-ViT B-224/16	78.66	78.73	81.11	81.19	81.85	83.09	85.83	94.73
ManagerTower <sub>BASE</sub> (Ours)	4M	CLIP-ViT B-224/16	<b>79.39</b>	<b>79.15</b>	<b>81.26</b>	<b>81.44</b>	<b>82.81</b>	<b>83.34</b>	<b>86.56</b>	<b>95.64</b>
<i>Models pre-trained on more data and/or with larger size</i>										
UNITER <sub>LARGE</sub> (Chen et al., 2020) *	4M	Faster R-CNN	73.82	74.02	79.39	79.38	79.12	79.98	75.56	87.30
VILLA <sub>LARGE</sub> (Gan et al., 2020) *	4M	Faster R-CNN	74.69	74.87	80.18	80.02	79.76	81.47	76.26	87.90
UNIMO <sub>LARGE</sub> (Li et al., 2021b)	4M	Faster R-CNN	75.06	75.27	81.11	80.63	-	-	78.04	89.40
ALBEF <sub>BASE</sub> (Li et al., 2021a) *	14M	DeiT B-224/16	75.84	76.04	80.80	80.91	82.55	83.14	85.6	95.9
VinVL <sub>LARGE</sub> (Zhang et al., 2021)	5.7M	ResNeXt-152	76.52	76.63	-	-	82.67	83.98	-	-
BLIP <sub>BASE</sub> (Li et al., 2022a) *	14M	DeiT B-224/16	77.54	77.62	-	-	82.67	82.30	87.2	96.6
SimVLM <sub>BASE</sub> (Wang et al., 2021b) *	1.8B	ResNet-101	77.87	78.14	84.20	84.15	81.72	81.77	-	-
BLIP <sub>BASE</sub> (Li et al., 2022a) *	129M	DeiT B-224/16	78.24	78.17	-	-	82.48	83.08	87.3	97.3
SimVLM <sub>LARGE</sub> (Wang et al., 2021b) *	1.8B	ResNet-152	79.32	79.56	85.68	85.62	84.13	84.84	-	-
VLMO <sub>LARGE</sub> (Wang et al., 2021a)	4M	BEiT L-224/16	79.94	79.98	-	-	85.64	86.86	84.5	95.3
SimVLM <sub>HUGE</sub> (Wang et al., 2021b) *	1.8B	Larger ResNet-152	80.03	80.34	86.21	86.32	84.53	85.15	-	-

Table 7: Comparisons with previous models on various downstream VL tasks. The best score is bolded. B, N and M in “ViT B-N/M” denote the model size, image resolution and patch size, respectively. \* indicates that the model also uses VG-QA data to fine-tune on VQAv2. \* denotes the model is trained from scratch. “# Pre-train Images” denotes the number of unique images used in VLP.



Hyperparameters	ManagerTower
Number of Layers	6
Hidden size	768
FFN inner hidden size	3,072
Number of Attention heads	12
Dropout Ratio	0.1
Attention dropout	0.1
Total Steps	100k
Batch Size	4,096
Optimizer	AdamW
Learning Rate	$1e^{-5}$
Learning Rate Decay	Linear
Weight Decay	0.01
Warmup Steps	10k
Adam $\epsilon$	$1e^{-8}$
Adam $\beta_1$	0.9
Adam $\beta_2$	0.98
Center-Crop	✓
Random Resized Crop	✗
Random Augmentation	✗
Random Horizontal Flipping	✗
Textual Encoder	RoBERTa <sub>BASE</sub>
Visual Encoder	CLIP-ViT B-224/16
Patch Size	16
Image Resolution for VLP	288

Table 8: Hyperparameters for pre-training. The first block is the hyperparameters for the cross-modal encoder.

Hyperparameters	VQAv2	SNLI-VE	NLVR <sup>2</sup>	Flickr30K
Total Epochs	10	4	5	20
Batch Size	576	64	256	512
Optimizer	AdamW	AdamW	AdamW	AdamW
Learning Rate	$9e^{-6}$	$3e^{-6}$	$1.4e^{-5}$	$6e^{-6}$
Learning Rate Decay	Linear	Linear	Linear	Linear
Weight Decay	0.06	0.01	0.01	0.01
Warmup Ratio	0.06	0.06	0.1	0.1
Adam $\epsilon$	$1e^{-8}$	$1e^{-8}$	$1e^{-8}$	$1e^{-8}$
Adam $\beta_1$	0.9	0.9	0.9	0.9
Adam $\beta_2$	0.98	0.98	0.98	0.98
Center-Crop	✗	✗	✗	✗
Random Resized Crop	✓	✓	✓	✓
Random Augmentation	✓	✓	✓	✓
Random Horizontal Flipping	✗	✓	✓	✓
Textual Encoder	RoBERTa <sub>BASE</sub>	RoBERTa <sub>BASE</sub>	RoBERTa <sub>BASE</sub>	RoBERTa <sub>BASE</sub>
Visual Encoder	CLIP-ViT B-224/16	CLIP-ViT B-224/16	CLIP-ViT B-224/16	CLIP-ViT B-224/16
Patch Size	16	16	16	16
Image Resolution for FT	576	384	384	384
Loss Function	BCE	CE	CE	CE

Table 9: Hyperparameters for fine-tuning ManagerTower on various downstream VL tasks. FT denotes fine-tuning. CE and BCE are short for cross-entropy loss and binary cross-entropy loss, respectively.

## ACL 2023 Responsible NLP Checklist

---

### A For every submission:

- A1. Did you describe the limitations of your work?  
*Section Limitations.*
- A2. Did you discuss any potential risks of your work?  
*This paper proposes a novel Vision-Language model architecture, ManagerTower, which may not contain potential malicious or unintended harmful effects and uses.*
- A3. Do the abstract and introduction summarize the paper’s main claims?  
*Section Abstract, and Section 1. Introduction.*
- A4. Have you used AI writing assistants when working on this paper?  
*Left blank.*

### B Did you use or create scientific artifacts?

*Section 2. Preliminary, and Section 3. Manager Design.*

- B1. Did you cite the creators of artifacts you used?  
*Section 2. Preliminary, and Section 3. Manager Design.*
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?  
*All the pre-train data and pre-trained uni-modal backbones are publicly available, we just use them for pre-training and initialization, and we don’t repackaging or release them. Our code and pre-trained models will be released, and the name of the license for each asset will be stated in our code repo.*
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?  
*Our code and pre-trained models will be released, and the name of the license for each asset will be stated in our code repo.*
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?  
*All the data we used are publicly available and used safely by previous works.*
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?  
*We will provide the documentation of our code and pre-trained models in our code repo.*
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.  
*Section Appendix A. Implementation Details.*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

**C  Did you run computational experiments?**

*Section Appendix C. Computational Budget.*

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?

*Section 4. Experiments.*

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

*Section 4. Experiments.*

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

*Section 4. Experiments.*

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

*Not applicable. Left blank.*

**D  Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

*No response.*

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

*No response.*

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

*No response.*

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

*No response.*

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

*No response.*